

Міністерство освіти та науки України
Національна металургійна академія України

Теорія та методика
навчання математики,
фізики, інформатики

*Збірник наукових праць
Випуск VI*

Том 3

Кривий Ріг
Видавничий відділ НМетАУ
2006

Теорія та методика навчання математики, фізики, інформатики:
Збірник наукових праць. Випуск VI: В 3-х томах. – Кривий Ріг: Видавничий відділ НМетАУ, 2006. – Т. 3: Теорія та методика навчання інформатики. – 408 с.

Збірник містить статті з різних аспектів дидактики інформатики і проблем її викладання в вузі та школі. Значну увагу приділено проблемам розвитку методичних систем навчання інформатики і питанням захисту інформації та інформаційної безпеки у шкільній та вузівській практиці.

Для студентів вищих навчальних закладів, аспірантів, наукових та педагогічних працівників.

Редакційна колегія:

В.М. Соловійов, доктор фізико-математичних наук, професор

Є.Я. Глушко, доктор фізико-математичних наук, професор

О.І. Олейніков, доктор фізико-математичних наук, професор

М.І. Жалдак, доктор педагогічних наук, професор

П.С. Атаманчук, доктор педагогічних наук, професор

В.І. Клочко, доктор педагогічних наук, професор

Ю.О. Дорошенко, доктор технічних наук, професор

О.Д. Учитель, доктор технічних наук, професор

І.О. Теплицький, відповідальний редактор

С.О. Семеріков, відповідальний секретар

Рецензенти:

Г.Ю. Маклаков – д-р техн. наук, професор кафедри кібернетики та обчислювальної техніки Севастопольського національного технічного університету, науковий керівник лабораторії біокібернетики, дійсний член Міжнародної академії біоенерготехнологій

А.Ю. Ків – д-р фіз.-мат. наук, професор, завідувач кафедри теоретичної фізики Південноукраїнського державного педагогічного університету (м. Одеса)

*Друкується за ухвалою вченої ради
Національної металургійної академії України*

ISBN 966-8413-21-7

КОМП'ЮТЕРНО-ОРІЄНТОВАНІ МЕТОДИЧНІ СИСТЕМИ НАВЧАННЯ У ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДАХ

Ю.В. Триус

м. Черкаси, Черкаський національний університет
імені Богдана Хмельницького
trius@cdu.edu.ua

Сучасний етап розвитку вищої освіти в Україні, обумовлений процесами інформатизації, глобалізації й інтелектуалізації суспільства, вимагає, по-перше, якісних змін у системі професійної підготовки фахівців у сфері природничо-математичних наук, комп'ютерної техніки, економіки, по-друге, перегляду цілей, завдань, змісту, методів, засобів і форм організації навчально-пізнавальної діяльності студентів ВНЗ. Фундаментальна роль у формуванні професійних якостей, ключових і загально-галузевих компетентностей цих фахівців належить математичним дисциплінам.

Світовий досвід і наукові дослідження свідчать про те, що перебудова системи вищої математичної освіти насамперед повинна здійснюватися на основі широкого впровадження у навчальний процес сучасних інформаційно-комунікаційних технологій. Використання цих технологій створює не лише реальні умови для розширення й поглиблення змісту математичної освіти, а й сприяє інтенсифікації процесу навчання, його результативності, інтелектуальному розвитку студентів, формуванню конкурентноспроможних фахівців в галузі математики і системного аналізу, інформатики та інформаційних технологій, економіки і фінансів.

Аналіз досліджень закордонних і вітчизняних фахівців у галузі теорії і методики навчання природничо-математичних дисциплін показав, що одним з реальних шляхів підвищення якості професійної підготовки майбутніх фахівців на рівні ВНЗ, є розробка науково-обґрунтованих *методичних систем навчання* з фахових дисциплін, які б сприяли активізації навчально-пізнавальної, науково-дослідної діяльності студентів, розкриттю їх творчого потенціалу, збільшенню ролі самостійної та індивідуальної роботи і ґрунтувалися б на широкому впровадженні у навчальний процес новітніх педагогічних та інформаційних технологій. Такі методичні системи одержали назву *комп'ютерно-орієнтованих методичних систем навчання*.

В наших роботах [1; 2] при створенні комп'ютерно-орієнтованих методичних систем навчання математичних дисциплін використані як традиційні підходи до створення методичних систем навчання, так і нові підходи, які використовуються науковцями при розробці комп'ютерно-орієнтованих методичних систем навчання інформатики та математики для загальноосвітніх шкіл і для ВНЗ.

Одним з основних положень концепції комп'ютерно-орієнтованих методичних систем навчання математичних дисциплін, запропонованої нами,

є те, що досягнення якісно нового рівня у підготовці фахівців з вищою освітою неможливе без забезпечення розвитку вищої школи на основі нових прогресивних концепцій, запровадження сучасних педагогічних та інформаційних технологій, науково-методичних досягнень, відходу від засад авторитарної педагогіки і застарілих технологій навчання. Проведене дослідження показало, що серед педагогічних інновацій, використання яких може забезпечити підвищення якості вищої математичної освіти, сприяти пізнавальній активності студентів і набуттю ними комунікативних умінь і навичок, формуванню вмінь самостійно конструювати свої знання та орієнтуватися в інформаційному просторі, є технологія модульно-рейтингового навчання, навчання в співпраці, метод проектів, ситуаційне та продуктивне навчання.

Кожна із зазначених вище інноваційних педагогічних технологій, інтегруючись з інформаційно-комунікаційними технологіями, повинна зайняти своє місце в навчально-виховному процесі ВНЗ, поступово витісняючи пасивні методи й форми навчання. Це згодом надасть можливість організувати навчальний процес у вищих навчальних закладах на високому рівні, з урахуванням специфіки вітчизняної вищої школи й національного культурного середовища.

Увесь хід об'єктивного розвитку інформатики як науки свідчить про те, що математика була не тільки материнською наукою для інформатики, але й сама інформатика в міру свого становлення і відокремлення в своїх основах і методах неухильно математизується. Також з'являється все більше і більше свідчень того, що методи інформатики, інформаційні технології проникають у глибини математики, впливають на деякі риси стилю, техніки і змісту математичної роботи. Зокрема це стосується використання математичних моделей та інформаційних технологій як одного з найпотужніших засобів пізнання реального світу для дослідження об'єктів дійсності, розв'язування практичних задач, що виникають у різних сферах діяльності людини. Тому не випадково, що при підготовці у ВНЗ фахівців у галузі математики, фізики, природничих наук, техніці, економіці побудові математичних моделей та їх дослідженню за допомогою інформаційних технологій приділяється значна увага.

Останнім часом акцент використання математичного моделювання та інформаційних технологій як засобу наукових, і, зокрема, математичних досліджень, зміщується з традиційного підходу, який ґрунтувався на розробці дослідницьких програм за допомогою проблемно-орієнтованих мов програмування (Fortran, Pascal, C, C++), у бік використання інтегрованих візуальних середовищ, які базуються на алгоритмічних мовах високого рівня (Delphi, Visual Basic, Visual C++ тощо), крім того, зростає роль універсальних математичних пакетів (MathCAD, Matlab, Maple, Mathematica, Derive) та спеціалізованих програмних продуктів.

Є кілька вагомих причин, які вимагають знань основ роботи з кількома

системами комп'ютерної математики від професіоналів у галузі математики, науково-технічних досліджень. Це:

- необхідність оптимального вибору систем комп'ютерної математики з урахуванням специфіки задач, що розв'язуються;
- необхідність розв'язування складних задач за допомогою різних систем, щоб перевірити правильність результатів, не покладаючись на одну систему (збільшити ймовірність правильності одержаного результату);
- необхідність підготовки математичних документів (статей, звітів, книг, навчальних занять і т.д.) підвищеної якості.

Широкі аналітичні, обчислювальні і графічні можливості сучасних математичних пакетів роблять їх одними з основних інструментів у професійній діяльності математика-аналітика, інженера, економіста-кібернетика тощо. Тому їх використання у навчальному процесі ВНЗ при вивченні математичних дисциплін дозволить підвищити рівень професійної підготовки студентів, рівень їх математичної та інформаційної культури, зробити майбутніх фахівців конкурентноспроможними на міжнародному ринку праці.

Нажаль, результати проведеного анкетування свідчать про невикористано обмежене використання потужної комп'ютерної підтримки при вивченні математичних дисциплін та розв'язуванні складних математичних і прикладних задач. При цьому ці результати можна певною мірою екстраполювати й на інші ВНЗ України. Для виправлення ситуації, що склалася:

– Міністерству освіти і науки України та ВНЗ необхідно здійснити реальні кроки для того, щоб якнайшвидше на математичних кафедрах в достатній кількості з'явилися ліцензійні або вільно розповсюджені математичні пакети для забезпечення навчального процесу на математичних, природничих, комп'ютерних, технічних та економічних спеціальностях;

– викладачам математичних кафедр необхідно проводити цілеспрямовану роботу щодо створення комп'ютерно-орієнтованих методичних систем навчання математичних дисциплін з широким використанням систем комп'ютерної математики.

Комп'ютерно-орієнтовані методичні системи навчання у ВНЗ повинні створюватися на основі новітніх педагогічних та інформаційно-комунікаційних технологій, використання яких повинно забезпечити створення у ВНЗ єдиного освітньо-наукового інформаційного середовища, в якому навчальна діяльність буде своєрідною моделлю майбутньої професійної діяльності студентів в умовах інформаційного суспільства. Нами за єдиною концепцією створено навчально-методичні комплекси таких математичних дисциплін, як: “Математична логіка і теорія алгоритмів”, “Комп'ютерна математика”, “Математичні методи оптимізації” для студентів математичних і комп'ютерних спеціальностей, “Математичне програмування” і “Дослідження операцій” для студентів економічних спеціальностей, які є основою освітньо-наукового інформаційного середовища математичних дисциплін ВНЗ.

Запропонований підхід щодо створення комп'ютерно-орієнтованих методичних систем навчання виявився досить ефективним, про що говорять результати тривалого педагогічного експерименту, численні впровадження створених автором навчально-методичних комплексів у практику ВНЗ України.

Проведене дослідження, присвячене удосконаленню методики навчання математичних дисциплін у вищих навчальних закладах на основі широкого використання новітніх педагогічних технологій та ІКТ, відкриває нові перспективи в подальших наукових пошуках зазначеного спрямування. Передусім потребують подальшого вирішення такі проблеми: розробка комп'ютерно-орієнтованих методичних систем навчання всіх математичних дисциплін, які вивчаються у ВНЗ III-IV рівнів акредитації; створення освітньо-наукових інформаційних середовищ математичних дисциплін на основі порталних технологій; дослідження особливостей використання технологій дистанційного навчання на денній і заочній формах навчання, подання та ефективного використання інформаційних ресурсів дистанційних курсів математичних дисциплін; забезпечення персоналізованого доступу до освітньо-наукових інформаційних ресурсів ВНЗ студентів різних форм навчання; розробка гнучкої системи індивідуальних траєкторій організації навчально-пізнавальної діяльності студентів та ефективної системи зворотного зв'язку на основі результатів поточного контролю, зокрема тестування в "онлайн" і "офлайн" режимах; інтеграція в системи дистанційного навчання навчальних модулів, розроблених у різних середовищах; вивчення психолого-педагогічних особливостей і розробка методики навчання математичних дисциплін в умовах дистанційного навчання.

Література:

1. Триус Ю.В. Комп'ютерно-орієнтовані методичні системи навчання математики / Черкаський національний ун-т ім. Богдана Хмельницького. – Черкаси : Брама-Україна, 2005. – 400 с.
2. Триус Ю.В. Комп'ютерно-орієнтовані методичні системи навчання математичних дисциплін у вищих навчальних закладах: Автореф. дис... д-ра пед. наук: 13.00.02 / Національний педагогічний ун-т ім. М.П. Драгоманова. – К., 2005. – 48 с.

Розділ І

Дидактика інформатики вищої школи

МЕТОДИЧНІ ТА ОРГАНІЗАЦІЙНІ ПРОБЛЕМИ НАВЧАННЯ КОМП'ЮТЕРНОГО ПРОГРАМУВАННЯ У ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДАХ

О.П. Поліщук^а, С.О. Семеріков^б

м. Кривий Ріг, Криворізький державний педагогічний університет

^а apol@cabletv.dp.ua

^б cc@kpi.dp.ua

Поточний стан. Комп'ютерне програмування – не тільки найважча з усіх існуючих інтелектуальних робіт, а й найбільш прибуткова, найбільш рентабельна галузь сучасних високих технологій, яка не потребує матеріальної сировини для вироблення продукту, енергоресурсів, транспортних затрат для перевезень і за мінімальних початкових затрат могла б приносити в бюджет країни неабиякі кошти – при наявності “критичної маси” висококваліфікованих (і відповідно оплачуваних) системних та прикладних програмістів. Недарма найбагатшою людиною планети є не нафтогазовий або металургійний магнат, а володар фірми з виробництва “софту”.

Дефіцит фахівців в цій галузі спостерігається не тільки в Україні, а й в усіх розвинених країнах (для кваліфікованих програмістів вони тримають свої двері гостинно розчиненими), але причини його різні – не можна не помітити наше відставання в виробництві програмних продуктів і від таких країн, як Індія та Китай. Бюджет України зовсім не відчуває надходжень від галузі комп'ютерного програмування.

Чомусь підготовці спеціалістів якраз в цій перспективній галузі у нас приділяється найменше уваги. Формально дисципліни, пов'язані з комп'ютерним програмуванням, передбачені в навчальних планах майже всіх спеціальностей – щоправда з таким самим результатом, як і вивчення іноземних мов, що починається в середніх класах загальноосвітніх шкіл і безрезультатно продовжується в усіх університетах (крім спеціалізованих навчальних закладів).

Спроби проаналізувати причини і відшукати вихід з такого становища приводять до замкнутого кола, в якому важко знайти первинний елемент, за який можна було б “витягти” весь ланцюг причин і наслідків. І все таки, якщо в електронній промисловості можна змиритися з безнадійним відставанням “назавжди”, то в галузі комп'ютерного програмування це було б рівнозначно визнанню інтелектуальної неповноцінності нації, що не тільки неприйнятно, але й невірно – доказом цьому поодинокі успіхи деяких наших школярів на міжнародних олімпіадах.

Хто винен? Тоді залишається згадати відому аксіому про те, що риба починає смердіти з голови, і подивитися в нормативні документи, що якимось стосуються цього питання.

Почнемо з підготовки шкільних вчителів інформатики – і виявиться,

що такої спеціальності не передбачено зовсім. Педагогічні університети готують вчителів інформатики по спеціальностям “з додатком” під назвами “Фізика та основи інформатики”, “Хімія та основи інформатики”, “Математика основи інформатики ” і т.п. Ресурс навчальних годин (кредитів, або, конкретніше, фонду оплати праці викладачів) на такі додатки не передбачено жодним стандартом, стандарти є тільки для моноспеціальностей – фізика, хімія, математика. Уникнути перетягування “кредитної ковдри” між кафедрами в боротьбі за збереження (або збільшення) робочих місць не можна і результатом перманентного скорочення годин на “дodatки з інформатики” вже сьогодні є повна неможливість в межах виділеного часу масової підготовки кваліфікованих вчителів інформатики.

Відомо, що спеціаліст з програмування професійно деградує, якщо не займається програмуванням по кілька годин щоденно. А як “професійно розвивається” студент, який займається програмуванням 2 академічні години (80 хвилин) на тиждень в проміжку між іншими дисциплінами і з перервами в кілька семестрів при відсутності комп’ютерних класів для самостійної роботи і коштів на придбання власного комп’ютера і потрібних посібників?

Якщо ще згадати про те, що контингент абітурієнтів “важких в навчанні” фізико-математичних факультетів педагогічних університетів формується переважно по “залишковому принципу”, що університети не мають коштів для придбання необхідних багатьох екземплярів дорогих підручників (тут необхідні затрати значно більші, ніж на придбання комп’ютерних класів) і в університетських бібліотеках на абонементі в цій галузі порожньо, що немає комп’ютерних класів для самостійної роботи (передбаченої в навчальних планах) – результат передбачити не важко. Трохи більш віддалений результат – абітурієнт педагогічного університету, підготовлений не кваліфікованим вчителем інформатики.

Ми досі не маємо офіційних вимог до кваліфікації вчителя інформатики – що повинен знати і вміти студент, якому присвоюється відповідна кваліфікація, тематика та рівень випускних робіт – тобто того, з чого й витікав би перелік необхідних дисциплін і послідовність їх викладання. Тому не дивно, що формулювання кваліфікаційних вимог покладається на викладача-предметника, який загальної картини не бачить і укладає робочу навчальну програму на свій розсуд. Відсутність стандартів зі спеціальності “Інформатика” у педагогічних ВНЗ призводить до абсурдної ситуації, коли зміст одного й того ж предмету у різних вузах (і навіть на різних факультетах одного педуніверситету) може бути повністю різним.

Коли вже згадали про викладачів, тут є свої проблеми. Як казав відомий “вчитель-предметник всіх дисциплін” Й.В. Сталін, “кадри вирішують все”. Якими повинні бути кадри викладачів комп’ютерного програмування? Загальні побажання очевидні – це повинен бути діючий зараз або в недалекому минулому спеціаліст з достатнім досвідом роботи в цій галузі, співвав-

тор ряду програм, що користуються попитом в тій чи іншій галузі матеріального виробництва або наукових досліджень. Якщо такий спеціаліст і був би в наявності, то він, як правило, не мав би вчених звань і ступенів (професійні програмісти не часто намагаються їх отримати – по основній роботі їх оцінюють за іншими критеріями) і міг би бути прийнятий на посаду асистента або в крайньому випадку старшого викладача з досить низьким посадовим окладом – навряд чи це для нього було б привабливим.

Традиційний розподіл роботи між викладачами полягає в тому, що професор (доцент) читає лекції, а асистент веде практичні заняття. Специфіка програмування в тому, що для проведення практичних занять потрібна більш висока кваліфікація, ніж для читання лекцій – викладач повинен дуже швидко виявляти помилки студентів і допомагати їх виправляти, а ці помилки бувають досить фантастичними. Справа в тому, що практичне системне чи прикладне комп'ютерне програмування, яке може давати прибуток, – це не тільки наука, це ще й “ремесло”, тут треба не тільки знати, а й вміти моментально використовувати знання для практичного втілення.

Ще одна традиція, яка не вписується в специфіку комп'ютерного програмування – читання лекцій з крейдою в руках для відображення на дошці ілюстративних фрагментів програм. Крім того, що така робота потребує багато часу, вона ще й неефективна – на вищербленій поверхні дошки (часто дуже малого розміру), не завжди ідеальним почерком викладача, не дуже зрозумілий для початківця текст на ще незнайомій мові програмування після внесення помилок при переписуванні студентом своїм неідеальним почерком стає малоприслужним для аналізу і підготовки до практичних занять.

Практичні заняття проводяться в комп'ютерному класі з чисельністю студентів в кращому випадку рівному половині стандартної групи – 12–13 студентів. “Обслужити” ефективно таку кількість жоден викладач не може, максимально допустима кількість 6–7 студентів при наявності сучасного обладнання (локальної комп'ютерної мережі, комп'ютерного проектора і великого екрану). При більшій кількості і відсутності обладнання є тільки два варіанти – або викладач біля все тієї ж дошки з крейдою намагається домогтися синхронної роботи всієї групи, або працює індивідуально, обслуговуючи тих, кого встигне за відведені 80 хвилин, та залишаючи решту поза увагою.

Ми не ставимо задачу перерахувати абсолютно всі організаційно-методичні проблеми, пов'язані з ефективністю процесу навчання комп'ютерному програмуванню, але ще на одну треба звернути увагу. Це так званий норматив “кількість студентів на одного викладача”, який приводить до фактичної неможливості відрахування хронічно невстигаючих студентів, бо це пов'язано з необхідністю скорочення і штату викладачів. Завдяки цьому нормативу державні університети попадають в ту ж ситуацію, що і приватні комерційні, в яких плата за навчання є єдиним джерелом фінансування і відрахування студентів призводить до “недобору” необхідних коштів. Нема

потреби згадувати про те, що далеко не всі студенти мають потрібні здібності і мотиви для успішного навчання, але всі вони добре розуміють свою “провідну роль” в навчальному процесі.

Що робити? З чогось треба ж починати – по-перше, з організації хоча б одного загальноукраїнського “розплідника програмістів” у вигляді науково-навчально-виробничого закладу, який фінансувався б з бюджету, займався відбором талановитої молоді, пошуком “незайнятих зон” і перспективної тематики для розробок, розробкою і збутом програмних виробів власного виробництва, навчанням програмістів вищої кваліфікації за контрактами, які передбачали б обов’язкову роботу в цьому ж закладі певну кількість років як плату за отриману кваліфікацію. Цей заклад треба було б наділити правами атестації і регулярної переатестації спеціалістів, що мають право викладати дисципліни з комп’ютерного програмування з диференціацією по різних методологіям і технологіям. Отримані в цьому закладі атестати повинні прівнюватись до відповідних звань професора і доцента і для них необхідно розробити окрему тарифну сітку, бо існуюча “зрівнялівка” в оплаті праці викладачів “белетристичних” дисциплін і дисциплін, що потребують здібностей і великої інтелектуальної напруги для підготовки навчальних матеріалів, не є стимулюючою. Студенти-початківці (і слухачі, що підвищують кваліфікацію) цього закладу могли б після деякого терміну навчання приймати участь в розробці комерційних програмних проектів, набуваючи досвіду під наглядом професіоналів. Базою для організації такого закладу міг би стати наприклад Інститут кібернетики ім. В.М. Глушкова або один з його підрозділів.

По-друге, необхідне створення та затвердження стандарту зі спеціальності “Інформатика” для педагогічних ВНЗ. Це дозволить уніфікувати підготовку вчителів інформатики (фахівців з програмування, що мають додаткову методичну підготовку), які здатні забезпечити масову підготовку кваліфікованих абітурієнтів на кібернетичні спеціальності.

По-третє, необхідно створення нових підручників з інформатики для педагогічних ВНЗ з алгоритмізації, прикладного та системного програмування, методів обчислень тощо, інваріантні до використовуваної мови програмування та операційної системи. Це дасть можливість стабілізувати курси інформатики, знявши нав’язувану країні штучну залежність від фірм-виробників програмного та апаратного забезпечення.

СПЕЦКУРС З СУЧАСНИХ МЕТОДІВ І ТЕХНОЛОГІЙ РОЗРОБКИ ПРОГРАМНИХ ВИРОБІВ КОМПОНЕНТНОЇ АРХІТЕКТУРИ

О.П. Поліщук^а, С.О. Семеріков^б

м. Кривий Ріг, Криворізький державний педагогічний університет

^а apol@cabletv.dp.ua

^б cc@kpi.dp.ua

Введення освітніх стандартів у педагогічних ВНЗ відіграло позитивну роль при плануванні навчання за моноспеціальностями “Фізика” та “Математика”. Проте на подвійних спеціальностях “Фізика та основи інформатики”, “Математика та основи інформатики” чіткі вимоги стандартів до кількості кредитів, з одного боку, та обмеження тижневого навантаження, з іншого, практично не залишили місця для додаткової спеціальності – інформатики.

При складанні та перегляді навчальних планів підготовки вчителів інформатики за подвійними спеціальностями (адже окремий напрямок підготовки “Інформатика” в педагогічних ВНЗ просто відсутній) відбувається перманентне скорочення ресурсів, відведених для навчання комп’ютерному програмуванню. У стандарті моноспеціальностей “Математика” та “Фізика” обсяг годин на інформатику менший, ніж на фізичне виховання, у навчальних планах подвійних спеціальностей – ненабагато більший, до того ж при їх складанні іде заміна програмістських дисциплін курсами загальноознайомчого характеру типу “Інформаційні технології і системи”, “Інформаційні технології в наукових дослідженнях”, (або в навчанні, або взагалі – “Сучасні інформаційні технології”) і т.п. Як правило, тематика таких дисциплін включає 15–18 розділів з досить серйозними назвами (“Інтелектуальні і експертні системи”, “Системи комп’ютерної графіки”, “Комп’ютерні мережі”, “Системи управління базами даних” і т.ін.). На всі такі розділи разом відводиться один семестр по одному практичному заняттю і від 0 до 1 лекції на тиждень – по одному заняттю на кожний розділ. Можна зустріти тематичні плани, в яких в курсах алгоритмізації і програмування практичні заняття відсутні зовсім. Така підміна приводить до підготовки в кращому випадку просто комп’ютерного користувача “побутового” рівня замість професійної підготовки фахівця в галузі високих технологій.

Будь-який технологічний процес виконується “в прямому напрямку” – від сировини до готового продукту, розробляється ж технологія “в зворотному напрямку” – від вимог до кінцевого виробу до розробки операцій і вимог до сировини і комплектуючих виробів. Технологія навчання не є виключенням. Тому ми спробуємо побудувати ескіз навчального плану для спеціальності “Інформатика”, що передбачає присвоєння випускнику кваліфікації “інженер-програміст, вчитель інформатики”, сформулювавши вимоги до його кваліфікації просто на основі “здорового глузду”.

Сучасні комп'ютеризовані системи, як правило, гетерогенні, тобто складаються як по апаратному, так і по програмному забезпеченню з різних компонентів, виготовлених різними виробниками. Проблема об'єднання таких компонентів в єдине ціле до кінця не вирішена, але в цьому напрямку ведуться інтенсивні роботи і отримані результати широко використовуються на практиці з початку 90-х років. Будемо вважати, що інженер-програміст повинен в якійсь мірі бути знайомим з сучасними методологіями програмування і в 9-му семестрі має прослухати відповідний спецкурс. Ми підготували такий спецкурс для визначення мінімально необхідної підготовки студентів спеціальності "Інформатика" для його сприйняття.

Спецкурс під назвою "Методи розробки гнучких інтегрованих програмних виробів" присвячений вивченню компонентних технологій в розробці складних комп'ютерних програм, складові частини (компоненти) яких можуть бути реалізовані на різних мовах програмування, для різних платформ, модернізуватися незалежно одне від одного без необхідності внесення змін в систему, що використовує їхні сервіси.

Курс розрахований на 36 годин лекцій і 36 годин лабораторного практикуму з програмування. Курс лекцій включає 10 тем різної складності.

Вступна лекція присвячується огляду основних конкуруючих компонентних технологій розробки програмних виробів компонентної архітектури – (D)COM, CORBA, Java, висвітленню особливостей кожної з них, аналізу таких характеристик, як підтримка доступу клієнта до компонента через комп'ютерні мережі, мобільність, незалежність від мови програмування, платформи, трудомісткість освоєння програмістами. Обмежений об'єм відведених на предмет часових ресурсів не дозволяє детально розбиратися в теоретичних основах і особливостях використання кожної технології і змушує розглянути характерні риси компонентного підходу на одній з них. Такою вибрана технологія Microsoft під назвою (D)COM (Components object model – об'єктна модель компонентів), що дозволяє використовувати широкий спектр можливостей по розробці компонентів і їх взаємодії з клієнтом – від компонентів на рівні екземплярів класу до компонентів, розміщених в різних потоках процесу, або в різних процесах, що виконуються на одному або на різних комп'ютерах.

Курс побудовано по принципу поступового ускладнення матеріалу і в першому змістовному розділі розглядається важливість уніфікації інтерфейсів між компонентами і клієнтами і найпростіший метод реалізації інтерфейсів (клієнта і компонентів в одному процесі і в одному потоці) на основі абстрактних базових класів з множинним наслідуванням, без використання динамічної компоновки. Розглядаються позитивні риси технології інкапсуляції інтерфейсів (можливість вдосконалювати їх функціональність без порушення роботи системи) і недоліки (можливість взаємодії клієнта і компонента в обхід інтерфейсів).

Наступний розділ присвячений удосконаленню цієї взаємодії через спеціальний інтерфейс (доступ до якого отримується при створенні компонента), що в свою чергу використовується для запиту про підтримку потрібного клієнту інтерфейсу. Приділяється увага важливому питанню обліку посилань на інтерфейси компонентів і визначенню часу життєдіяльності компонентів. Тільки після цього здійснюється перехід до розміщення компонентів в бібліотеках динамічної компоновки і розглядається процес експорту функцій і взаємодії кількох клієнтів і компонентів в різних комбінаціях. Розміщення компонентів в динамічних бібліотеках дає можливість замінити їх під час виконання, забезпечуючи велику гнучкість інтегрованої програмної системи.

Після приділення необхідної уваги деяким деталям (наприклад методам генерації неповторюваних кодів інтерфейсів, розгляду структури системного реєстру, реєстрації компонентів-серверів в реєстрі) здійснюється перехід до наступного ускладнення – використанню фабрики класів (компонента, який служить для створення інших компонентів) і її стандартних інтерфейсів.

Наступний етап – вивчення методів включення і агрегування як інструментів ієрархічної інтеграції для повторного використання компонентів, які самі можуть бути клієнтами компонентів.

Після цього ідуть розділи, присвячені досить питанням розміщення компонентів-серверів в різних процесах, локальним і віддаленим викликам процедур, техніці створення замісників і заглушок, основам мови опису інтерфейсів IDL (Interface Definition Language) і використанню компілятора MIDL для автоматичної генерації динамічних бібліотек “замісника/зглушки”, перетворенню через реєстр локальних серверів у віддалені, створенню диспетчерських інтерфейсів і автоматизації.

Заключна тема курсу – багатопоточність і розміщення компонентів в різних потоках, реалізація розділених і вільних потоків, підрозділи (конгломерат з потоку і циклу вибірки повідомлень) та питання потокобезпеки.

Всі лекції ілюструються по можливості фрагментами програм, приклади повної програмної реалізації ілюстрацій до кожної теми розміщуються на комп'ютерах робочих місць в класі для практичних занять по програмуванню.

Всі приклади носять “шкільний характер”, інтерфейси забезпечують просту функціональність (прийом чисел і рядків з клавіатури, сортування).

Методично-теоретичний матеріал базується переважно на роботах Дейла Роджерсона та Дональда Бокса. Приклади програм, що пропонуються Д. Роджерсоном, доповнені більш змістовною функціональністю інтерфейсів. В зв'язку з тим, що курс програмування в графічному середовищі навчальним планом не передбачений, графічний інтерфейс в прикладах замінено текстовим консольним (хоч це й не відповідає сучасним стандартам програмування).

Очевидно, що для сприйняття курсу студенті повинні мати базові знання і навички по основам об'єктно-орієнтованого програмування (вміти реалізовувати класи, розробляти конструктори, операторні функції, володіти методами одиночного і множинного наслідування класів в їх ієрархії тощо).

Об'єктно-орієнтоване програмування, як засіб подолання складності сучасних програмних виробів, в свою чергу вимагає надбання навичок роботи зі складними багатофайловими проектами, в яких присутня та складність, яку треба "долати". Базисом для переходу на об'єктну методологію повинна бути обізнаність з методами структурного процедурного програмування, фундаментальними алгоритмами і структурами даних, не кажучи вже просто про володіння технікою програмування на сучасних мовах високого рівня, знання бодай стандартних бібліотек функцій і класів, володіння допоміжними мовами типу IDL для опису інтерфейсів, синтаксисом створення файлів опису модулів, make-файлів тощо.

Необхідність розробки розподілених компонентів програмних систем вимагає освоєння багатозадачних операційних систем, теорії комп'ютерних мереж, основ динамічної компоновки і побудови динамічних бібліотек, принципів побудови компіляторів.

Звідси й випливає перелік і послідовність вивчення дисциплін, що складають цільову основу програмістської кваліфікації в чисто технічному плані – ми спеціально не зупинялись на основах теоретико-математичної та методичної підготовки.

Поза межами цієї статті залишилось багато невирішених питань організаційного і методичного характеру, питань технічного забезпечення процесу навчання комп'ютерному програмуванню, без вирішення яких ефективна підготовка потрібних країні фахівців з високими технологій неможлива.

СУЧАСНІ МЕТОДИ І ЗАСОБИ НАВЧАННЯ, ЩО АКТИВІЗУЮТЬ ПІЗНАВАЛЬНУ ДІЯЛЬНІСТЬ СТУДЕНТА ПРИ ВИВЧЕННІ ДИСЦИПЛІН КОМП'ЮТЕРНОЇ НАПРАВЛЕНОСТІ

Л.О. Савчук

м. Хмельницький, Хмельницький економічний університет
hmbiz37@infocom.km.ua, lyudasavchuk@ukr.net

В останні роки у зв'язку з комп'ютеризацією сфери освіти набули поширення нові методи, які передбачають використання сучасних технічних та програмних засобів у навчанні, з'явилися дослідження, присвячені впровадженню інформаційної технології в навчальний процес.

Проникнення засобів обчислювальної техніки в повсякденну практичну діяльність людей стимулює появу компонентів нових інформаційних технологій у складі навчального матеріалу практично всіх дисциплін. Слід підкреслити, що нова інформаційна технологія навчання характеризує собою педагогічну технологію.

Інформаційна технологія включає в себе функціональні компоненти, пов'язані з процесом циркуляції і переробки інформації, змістовний компонент (базу знань, яка складається з моделі предметної області, бази даних і прикладного програмного забезпечення) і опорну інформаційну технологію, в основі якої знаходяться базові апаратні і програмні засоби зберігання, переробки й обміну даними. Конкретна інформаційна технологія створюється шляхом реалізації бази знань на опорну технологію. Питання створення, використання і розвитку інформаційних технологій досліджувались в працях Є.П. Веліхова, В.М. Глушкова, В.І. Гриценко, Б.М. Паньшина, В. Гузєєва, Б.П. Єршова, В. Кайміна, А.А. Муранова та ін. Водночас вони не вичерпують проблему, оскільки в працях названих авторів не враховується специфіка і можливості використання нових інформаційних технологій навчання, які вносять суттєві зміни в організацію навчального процесу. Серед визначень поняття інформаційної технології, наведемо трактування М.І. Жалдака: "Під інформаційними технологіями розуміються сукупність методів та технічних засобів, що поглиблюють знання людей і розвивають їх можливості по управлінню технічними й соціальними процесами" [2].

Визначимо нову інформаційну технологію навчання як методологію і технологію навчально-виховного процесу з використанням новітніх електронних засобів навчання та активних методів, при яких студент, будучи суб'єктом педагогічного процесу не тільки сприймає те, що подає йому викладач, але й сам організовує свою самостійну роботу.

Серед основних складових нових інформаційних технологій, що активізують пізнавальну діяльність студентів, можна виокремити нові методи, технічні та програмні засоби активізації пізнавальної діяльності (таблиця 1).

У даній статті пропонуються до розгляду ті методи активізації навча-

льно-пізнавальних дій та технічні засоби навчання, які, на нашу думку, найбільш ефективно викликають пізнавальну, емоційну і практичну активність студентів та формують у них нове цілісне світорозуміння, нову модель підготовки членів майбутнього інформаційного суспільства.

Таблиця 1

Нові інформаційні технології навчання (НІТН)		
Методи активізації пізнавальної діяльності	Технічні засоби навчання	
	<i>Апаратні</i>	<i>Програмні</i>
1. Лекційний метод		
2. Інтелектуальна розминка	1. Персональні електро-но-обчислювальні машини	1. Навчальні програмні засоби.
3. Сократичний метод	2. Комунікаційні засоби (комп'ютерні мережі, телебачення, радіо)	2. Інформаційно-пошукові та інформаційно-довідкові ПЗ.
4. “Мозковий штурм”	3. Засоби введення інформації (сканери, цифрові фотокамери, мікрофони)	3. Системи мультимедіа та гіпермедіа.
5. Навчально-тематична дискусія	4. Засоби виведення інформації (принтери, плотири)	4. ПЗ-тренажери.
6. Ігрові методи (ділові ігри)	5. Засоби візуального відображення інформації (монітори, мультимедійні проектори)	5. Імітаційні та моделюючі ПЗ
7. Практичні технології, тренінги	6. Засоби збереження великих об'ємів інформації (магнітні, оптичні, флеш носії)	6. Системи штучного інтелекту. Експертні системи.
8. Методи проєктів (дослідницький, пошуковий)	7. Інтерактивна дошка SMART – Interactive White Board (IWB).	7. Програмні засоби міжкомп'ютерного зв'язку, сервісні ПЗ.
9. Авторська технологія конструювання знань при вивченні дисциплін комп'ютерної направленості.		8. Програмні засоби, що реалізують CASE – технології.

Методи активізації навчально-пізнавальної діяльності – це сукупність прийомів і способів психолого-педагогічного впливу на студентів, що спрямовані на розвиток у них творчого самостійного мислення, активізацію пізнавальної діяльності, формування творчих навичок та вмінь нестандартного розв'язання певних професійних проблем. Відтак кожен з методів може набути ознак проблемного, інтерактивного. Це стане можливим в разі комбінованого застосування дискусії, діалогу, евристичної бесіди, елементів “мозкового штурму”, інтелектуальної розминки, організації практикумів, обговорення проблем, розробки проєктів для самостійного дослідження,

аналізу та узагальненню. Педагогічний та дидактичний потенціал методів активізації пізнавальної діяльності, є беззаперечний лише разом з навчально-методичним, нормативно-технічним, інструктивним матеріалом.

Під технічними засобами варто розуміти програмно-апаратні засоби й пристрої, що функціонують на базі мікропроцесорної техніки, сучасних засобів і систем телекомунікацій, інформаційного обміну, аудіо-, відеотехніки й т.п., що забезпечують операції по збору, продукуванню, нагромадженню, зберіганню, обробці, передачі інформації.

Прискорення науково-технічного прогресу, засноване на впровадженні у виробництво гнучких автоматизованих систем, мікропроцесорних засобів і пристроїв програмного управління, роботів й обробних центрів, поставило перед сучасною педагогічною наукою важливе завдання – виховати й підготувати молодь, здатну активно включитися в якісно новий етап розвитку сучасного суспільства, пов'язаний з інформатизацією. Рішення цього завдання – виконання соціального замовлення суспільства – докорінно залежить як від технічної оснащеності навчальних закладів електронно-обчислювальною технікою з відповідним периферійним устаткуванням, навчальним, демонстраційним устаткуванням, що функціонує на базі НІТ, так і від готовності студентів до сприйняття постійно зростаючого потоку інформації, у тому числі й навчальної.

Використання інформаційних ресурсів, що є продуктом інтелектуальної діяльності найбільш кваліфікованої частини працездатного населення суспільства, визначає необхідність підготовки творчо активного резерву. Із цієї причини стає актуальною розробка певних методичних підходів до використання НІТ. Зокрема, для розвитку творчого потенціалу особистості, формування уміння здійснювати прогнозування результатів своєї діяльності, розробляти стратегію пошуку шляхів і методів розв'язання завдань – як навчальних, так і практичних.

Не менш важливе завдання забезпечення психолого-педагогічними й методичними розробками, спрямованими на виявлення оптимальних умов використання технічних засобів із метою інтенсифікації навчального процесу, підвищення його ефективності і якості, активізації пізнавальних дій. Актуальність оптимального використання технічних ресурсів у навчально-виховному процесі визначається не тільки соціальним замовленням, але й потребами особистості до самовизначення й самовираження в умовах сучасного інформатизованого суспільства. Студенти виступають у ролі розробників, коли вони використовують сучасні технічні засоби у якості інструментів пізнання для аналізу світу, інтерпретації і організації своїх власних знань та подання цих знань іншим людям.

В умовах інформатизації освіти змінюється парадигма педагогічної науки, змінюється структура й зміст освіти. Йде процес використання програмних засобів і пакетів програмних засобів навчального призначення для підтримки всіх методів навчання. При цьому програмним засобам переда-

ються навчальні функції й, отже, кожна програма будується згідно дидактичних принципів навчання.

Для ефективного використання нових інформаційних технологій навчання необхідне створення адаптивної моделі дій педагога, що враховує цілі, методи, результати навчання і вирішує такі основні завдання, як завдання діагностики психологічного стану й рівня знань того, кого навчають, і завдання керування його пізнавальною діяльністю. Вивчення нових методів та засобів навчання не повинне бути зведене до освоєння конкретних засобів інформаційних і комунікаційних технологій, необхідно, насамперед, сформулювати наукові основи, базу для освоєння нових технологій.

Ключовими питаннями вивчення сучасних методів та засобів, що забезпечують єдність методичного підходу до їхнього впровадження, є питання єдності засобів і методів навчання, алгоритмічної основи реалізації технологій, функціональної повноти активізації пізнавальних дій, що потребують постійного подальшого вивчення та впровадження в практику викладацької діяльності.

Література:

1. Выбор методов обучения в средней школе. / Под ред. Ю.К. Бабанского. – М., 1981.
2. Вашенко Г. Загальні методи навчання. – К., 1997. – С. 187–188.
3. Жалдак М.И. Система подготовки учителя к использованию информационных технологий в учебном процессе. – М., 1989. – 48 с.
4. Сметанський М.І., Галузяк В.М. Педагогічна влада та її виховний потенціал // Педагогіка і психологія. – 1996. – № 4. – С. 32-39.
5. Хуторской А.В. Интернет в школе. Практикум по дистанционному обучению. – М.: ИОСО РАО, 2000.-304
6. Інформаційні системи і технології в економіці. / За ред. д.е.н. В.С. Пономаренка. – К.: Академія, 2002.

ЦІЛІ ВИВЧЕННЯ ДИСЦИПЛІН КОМП'ЮТЕРНОГО ЦИКЛУ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У ВИЩІЙ ШКОЛІ

В.Г. Логвіненко

м. Суми, Сумський національний аграрний університет
vs@sau.sumy.ua

Початкові передумови. Питання про цілі вивчення дисциплін у закладах освіти неодноразово підіймалося в педагогічній літературі. Існують різні дефініції цього поняття у педагогіці. Згідно [1, 17], під ціллю розуміється «деякий вираз, який описує очікувані результати викладання». У [2, 134] мова йде про те, що «ціль указує на майбутній стан об'єкту або системи, до якого воно прагне». За таксономією Б. Блума та Д. Кратволя [3], виділяють цілі у когнітивній, афективній і психомоторній області. Автори [4, 43-62] також аналізують дане поняття з погляду навчання. Цілі повинні бути діагностичними, щоб результат діяльності можна було представити у вигляді зовнішніх кількісних і якісних характеристик, які можна упізнати. Цей результат є деякий приріст досвіду особистості, який є сукупністю привласнених нею знань, умінь і навичок. Згідно В.П.Безпалько [5], використовуються наступні рівні результатів навчання: учнівський, репродуктивний, продуктивний та творчий.

Основною метою навчання є отримання знань, умінь та навичок, тому опис цих цілей є найбільше розповсюдженим у педагогічній практиці.

Дана стаття продовжує розгляд питання створення системи цілей щодо вивчення дисциплін комп'ютерного циклу та інформаційних технологій.

Постановка задачі. На підставі аналізу науково-педагогічних і навчально-педагогічних джерел з дидактичних основ організації навчально-пізнавальної діяльності *необхідно* виділити цілі вивчення дисциплін комп'ютерного циклу (ДКЦ) та інформаційних технологій (ІТ).

Результати. Виділення навчального змісту дисципліни, основних його компонентів необхідно пов'язувати з метою вивчення дисципліни. У кожній навчальній дисципліні можна виділити так звані тактичні цілі. У різних науково-педагогічних роботах називаються цілі вивчення ДКЦ, які назвемо тактичними. До них відносяться:

– перша група цілей – *формування наукового світогляду*. Вивчення ДКЦ у ВНЗ повинно формувати уяву про методи, що застосовуються і використовуються в інформатиці, а отже формувати високий рівень наукової підготовки студентів;

– друга група цілей – *забезпечення знань, умінь і навичок*. Вивчення ДКЦ у ВНЗ повинно гарантувати володіння науковим фундаментом вивчених комп'ютерних дисциплін, повне та глибоке розуміння його ідей, фактів, методів, структури, глобальних цілей і тонкості викладення окремих питань курсу. Вивчення ДКЦ у ВНЗ повинно забезпечувати студентам

знання, уміння та навички, що є необхідними для застосування комп'ютерних засобів у дослідженнях, пов'язаних із науковою та професійною діяльністю, можливості самостійної роботи з використанням комп'ютера як ефективного засобу пізнання інформатики;

– третя група цілей – *формування логічної і інформаційної культури*. Вивчення ДКЦ у ВНЗ повинно забезпечити розвиток логічного мислення студентів, що має наступні якості: повнота аргументації, домінування логічної схеми міркувань, чіткістю процесу міркувань. Студентів потрібно знайомити із проблемами використання комп'ютера як інструменту пізнання. Слід залучати студентів до думки, що інформатика вивчається не за ради неї самої, але й для того, щоб використовувати комп'ютер при вивченні інших дисциплін, що без відповідних умінь складно говорити про інформаційну культуру;

– четверта група цілей – *формування досвіду професійної діяльності*. Вивчення ДКЦ у ВНЗ повинно забезпечити студентам достатній досвід майбутньої діяльності, створити умови для формування поглядів майбутнього фахівця, формувати уміння перетворювати науковий матеріал навчальної дисципліни. Інформатика формує у студентів уміння та навички використання комп'ютерів для опрацювання інформації, вирішенню задач із різних областей;

– п'ята група цілей – *виховання інтересу до інформатики та ІКТ*. Вивчення ДКЦ у ВНЗ повинно забезпечувати стійкий інтерес до цих дисциплін.

Для організації навчального процесу представлені цілі необхідно конкретизувати. Для цього скористаємося допоміжним засобом – таксономією цілей (Б. Блум, Д. Кратволь). Цілі ДКЦ в термінології когнітивної і афективної областей представлені в табл. 1, 2.

Таблиця 1

Опис цілей вивчення ДКЦ та ІТ у когнітивній області через діяльність студента

<i>Категорія</i>	<i>Дії студента</i>
1. Знання	студент знає і відтворює основні поняття, терміни, методи, правила і принципи роботи комп'ютера; студент знає і відтворює основні поняття, терміни, використовувані методи в ІТ; студент знає етапи виконання (і їх суть) технологічних процедур в ІТ: представлення первинних даних; розв'язання задач обліку і аналізу техніко-економічних показників; розв'язання задач планування техніко-економічних показників; розв'язання задач моделювання і прогнозування; розв'язання задач проектування; організація зберігання даних на електронних носіях; організація

<i>Категорія</i>	<i>Дії студента</i>
	друкування інформації; передача даних по каналах зв'язку; створення і ведення баз даних; робота в локальних і глобальних мережах; студент знає технологію роботи і відтворює порядок роботи з вивченим ІТ.
2. Розуміння	студент розуміє основні принципи роботи комп'ютера; студент інтерпретує схеми, алгоритми; студент виділяє початкові дані, вводить необхідні дані і змінні; студент перетворює словесний матеріал на мову блок-схем і на мову програмування, і навпаки; студент розуміє суть застосування ІТ для певного класу задач; студент розуміє суть використовуваних методів; студент розуміє суть технологічних процедур в ІТ: представлення початкових даних; розв'язання задач обліку і аналізу техніко-економічних показників; розв'язання задач планування техніко-економічних показників; розв'язання задач моделювання і прогнозування; розв'язання задач проектування; організація зберігання даних на електронних носіях; організація друкування інформації; передача даних по каналах зв'язку; створення і ведення баз даних; робота в локальних і глобальних мережах.
3. Застосування	студент пояснює використання алгоритму; студент використовує поняття, терміни, правила, методи і принципи при вирішенні задач; студент демонструє правильне застосування методів і процедур, відомих алгоритмів дій; студент застосовує ІТ для розв'язання задач певного класу.
4. Аналіз	студент здійснює розчленовування навчального матеріалу на компоненти (блоки) для кращого з'ясування; студент робить висновки після виконання завдання; студент зіставляє відмінності після введення різних даних; знаходить (бачить) помилки в логіці розв'язання задачі; студент зіставляє застосування різних ІТ при вирішенні певного класу задач.
5. Синтез	студент складає новий план або алгоритм; студент виконує постановку задачі; студент має навички грамотної постановки задач для їх розв'язання за допомогою комп'ютера; студент має уміння формалізованого опису поставлених задач;

<i>Категорія</i>	<i>Дії студента</i>
	студент має уміння побудови алгоритмів розв'язання задач по їх математичних моделях; студент пропонує використання знань з інших областей знань.
6. Оцінка	студент оцінює логіку розв'язання задачі (алгоритму, програми); студент оцінює одержані результати виходячи з початкових даних, студент оцінює відповідні висновки; студент оцінює значущість свого продукту навчальної діяльності; студент оцінює застосування різних ІТ при вирішенні певного класу задач.

Таблиця 2

Опис цілей вивчення ДКЦ та ІТ в афективній області через діяльність студента

<i>Категорія</i>	<i>Дії студента</i>
1. Сприйняття	студент усвідомлює важливість навчання, важливість вивчення пропонованого курсу, усвідомлює свою реалізацію в навчанні як чинника професійного зростання; студент уважно слухає в аудиторії; студент уважно вивчає інструкції, по яких працює.
2. Реагування	студент виконує завдання, вказане викладачем; студент підкоряється внутрішньому розпорядку і правилам поведінки; студент добровільно викликається виконувати завдання; студент виявляє цікавість до предмету, до виконуваного завдання; студент задоволений процесом виконання завдання.
3. Засвоєння ціннісної орієнтації	студент проявляє стійке бажання оволодіти уміннями роботи з комп'ютером і уміннями вирішувати задачі із застосуванням ІТ; студент проявляє переконаність, відстоюючи застосування того або іншого способу дій, методу або застосування певного програмного засобу при вирішенні задач.
4. Організація ціннісної орієнтації	студент бере на себе відповідальність за свою поведінку, за виконання отриманих завдань; студент розуміє свої можливості при вивченні певної теми; студент розуміє значення своєї реалізації в навчанні, в майбутній професійній діяльності; студент проявляє готовність до виконання завдань;

<i>Категорія</i>	<i>Дії студента</i>
	студент проявляє прагнення до співпраці в групі; студент планує свою роботу над завданням відповідно до своїх здібностей, потреб.

Виділені вище цілі дають нам загальне уявлення про формування дій при вивченні певної комп'ютерної технології. Їх можна деталізувати, виходячи з конкретної ІТ, що вивчається, теми кожного лекційного і лабораторного заняття. Далі, виділяються оперативні цілі вивчення кожної теми ДКЦ та ІТ.

Висновки. Таким чином, приходимо до наступного:

1) представлені тактичні цілі вивчення ДКЦ та ІТ дозволяють виділити оперативні цілі вивчення кожної теми, що вивчається. Це є обов'язковим елементом методичної системи вивчення будь-якої ДКЦ та ІТ;

2) виділені цілі вивчення ДКЦ та ІТ дозволяють діагностувати стан студента у процесі його руху від незнання до знання конкретної інформаційної технології;

3) подана системи цілей ДКЦ та ІТ реалізує діяльний підхід вивчення ДКЦ та ІТ.

Література:

1. Мелецинек А. Инженерная педагогика. – М.: МАДИИ(ТУ), 1998. – 185 с.
2. Коваленко Е.Э. Методика профессионального обучения. Учебник для инженеров, педагогов, преподавателей спецдисциплин системы профессионально-технического и высшего образования. –Х.: ЧП «Штрих», 2003. – 480 с.
3. Кларин М.В. Педагогическая технология в учебном процессе. Анализ зарубежного опыта. – М.: Знание, 1988 (Новое в жизни, науке и технике. Сер. «Педагогика и психология», № 6). – 80 с.
4. Артюх С.Ф., Коваленко Е.Э., Белова Е.К., Изюмская Г.В., Беликова В.В. Педагогические аспекты преподавания инженерных дисциплин. Пособие для преподавателей. – Харьков: УИПА, 2001. – 210 с.
5. Беспалько В.П. Слагаемые педагогической технологии. – М.: Педагогика, 1989. – 191 с.

ОРГАНІЗАЦІЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ “ІНФОРМАТИКА” З ВИКОРИСТАННЯМ ІНФОРМАЦІЙНО-НАВЧАЮЧОГО СЕРЕДОВИЩА

О.І. Миронова

м. Луцьк, Волинський державний університет імені Лесі Українки
mirelena@ukr.net

Динамічний розвиток науки та техніки, формування інформаційного суспільства ставлять перед майбутнім спеціалістом нові вимоги. Особливо актуальним це є для спеціалістів у області ІТ-технологій: неможливо надати весь багаж знань, що постійно оновлюються, вдосконалюються, які будуть використані ним у професійній діяльності.

У зв'язку з цим все більш актуальною (особливо у контексті інтеграції вітчизняної освіти до європейського освітнього простору [7]) стає проблема підвищення ефективності самостійної роботи студентів як форми діяльності, що сприяє формуванню самостійності у набутті знань, яка “передбачає оволодіння складними вміннями і навичками бачити зміст та мету роботи, організувати власну самоосвіту, вміння по-новому підходити до вирішуваних питань, пізнавальну і розумову активність і самостійність, здатність до творчості” [1].

Згідно до “Положення про організацію навчального процесу у вищих навчальних закладах” самостійна робота студента є основним засобом оволодіння навчальним матеріалом у час, вільний від обов'язкових навчальних занять. Дана теза регламентує традиційну ціль самостійної роботи, а саме: засвоєння, розширення знань, вдосконалення вмінь та навичок, набуття досвіду творчої та науково-дослідної діяльності тощо. Ще однією метою самостійної роботи є “розвиток такої риси особистості, як самостійність” [4, 11], “розвиток внутрішньої та зовнішньої самоорганізації майбутнього спеціаліста, активно-перетворюючого відношення до інформації, що отримується, здатності вибудовувати власну траєкторію самоосвіти” [2, 105].

Самостійна робота дозволяє отримати глибокі та міцні знання, вміння застосовувати вже набуті; виробляє у студентів установку на свідоме систематичне поповнення своїх знань, вміння раціонально планувати та використовувати свій час; сприяє посиленню індивідуалізації навчання, активізації навчальної діяльності, підвищенню мотивації навчання, розвитку пізнавальних здібностей; формує загальні навички організації самостійної роботи, що дозволяє інтенсифікувати процес навчання.

Розглядаючи історію філософської та педагогічної думки, робимо висновки, що самостійній роботі людини у процесі навчання завжди надавалось особливе значення. Наприклад, в основу сократівського методу діалогу покладене твердження про те, що не можна передати людині готове знання, а потрібно допомогти їй самостійно відкрити істину для себе. Багатьма вче-

ними, педагогами робились спроби дати означення поняттю “самостійна робота”, розкрити її суть: Б.П. Єсипов, Н.Г. Дайрі, П.І. Підкасистий, В.К. Буряк, В.А. Козаков, В.Д. Мороз, Л.М. Журавська, Є.В. Заїка і т.д.

Незважаючи на тривалість досліджень у даній галузі, і на даний момент однією з найважливіших проблем залишається відсутність чіткого категоріального апарату стосовно даного питання, що спричиняє появу “різноманітних, часто суперечливих та не завжди вдалих підходів до організації самостійної роботи” [3, 29].

Організація самостійної роботи передбачає побудову деякої моделі, системи, яка включатиме: здійснення підготовки студентів до виконання самостійної роботи (мотивація, здійснення певного інструктажу); планування самостійної роботи викладачем (з зазначенням форм робіт, терміну (графіку) їх виконання, видів контролю); забезпечення необхідних умов для її реалізації (наявність потрібного технічного (комп’ютери, сканери, принтери, ксерокси і т.д.) та програмного забезпечення, підключення до мережі Інтернет); розробку, надання необхідного методичного забезпечення (навчальні розробки у паперовому, електронному вигляді), списки необхідної літератури, періодичних видань, електронних джерел); здійснення контролю за виконанням самостійної роботи, надання консультаційно-методичної допомоги студентам.

У [6] визначено місце, перспективи розвитку курсу “Програмне забезпечення ЕОМ”, на основі чого зроблені висновки щодо співвідношення самостійної роботи та аудиторних занять та запропоновано як варіант підвищення ефективності вивчення зазначеного курсу та досягнення якісного рівня самостійної роботи використання навчальних систем.

Продовжуючи дослідження у даній галузі, пропонуємо систему самостійної роботи, яка базується на використанні інформаційно-навчального середовища, що включатиме підсистеми, відповідні до означених вище компонент організації системи самостійної роботи.

Програмні підсистеми:

Організаційна – у даній підсистемі буде відображено план виконання самостійної роботи, зафіксовані завдання, надано доступ до методичних матеріалів, надано план проведення занять з зазначенням видів та тем.

Виконавча – навчаюча система [5], яка дозволить не лише подати теоретичний матеріал (як паперові посібники), а, що є дуже важливим моментом при вивченні програмних продуктів, продемонструвати роботу системи, відразу вийти у середовище, що розглядається, та здійснити контроль за взаємодією користувача з програмним продуктом, своєчасно скорегувавши його дії, що є необхідним компонентом ефективної самостійної роботи.

Наповнення такої системи здійснюватиме викладач згідно визначеного плану та розробленої системи завдань. Результати роботи студента у даній системі передаватимуться у наступну підсистему.

Результуюча – дана підсистема призначена для відображення результа-

тів роботи у виконавчій підсистемі, присутності на заняттях, результатів поточного контролю і роботи на занятті та загальний підсумок роботи, який буде здійснений викладачем.

Комунікаційна – дана підсистема є тим компонентом, який здійснюватиме зв'язок між викладачем та студентами та самими студентами.

Отже, після побудови плану виконання самостійної роботи, розробки системи завдань та відповідного методичного забезпечення, наповнення виконавчої підсистеми, викладач надає ці матеріали (та доступ до виконавчої підсистеми) у використання студентам. Залежно від поставлених завдань студент виконує їх за допомогою виконавчої підсистеми (результати роботи у якій автоматично потрапляють до результуючої підсистеми) або інших необхідних засобів (результати такої роботи попадають у результуючу підсистему після знайомства викладача з виконаною роботою). При потребі за допомогою комунікаційної підсистеми може бути здійснено зв'язок студента зі своїми колегами чи викладачем.

Викладач фіксує виконання плану робіт, рівень досягнень, проводить поточний, підсумковий контроль, робить висновки щодо рівня засвоєння матеріалу, набуття навичок самостійності та вносить необхідні корективи у процес.

Вважаємо, що така організація самостійної роботи дозволить:

а) підвищити мотивацію до навчання: постійне відображення результатів, аналіз їх викладачем та автоматично за допомогою побудови графіків динаміки роботи, навчальних досягнень і т.п. дозволить більш наочно здійснювати порівняння власних здобутків з результатами решти, що активніше сприяє “духу суперництва” (у позитивному розумінні цього терміну); використання інформаційних технологій у самостійній роботі для студентів-інформатиків набуває елементів професійності, що сприятиме зацікавленості;

б) підвищити рівень володіння інформаційними технологіями;

в) розвивати навички раціонально планувати свій час;

г) спростити роботу з відсутніми на аудиторних заняттях, відстаючими студентами, тими, які намагаються уникнути виконання завдань шляхами: “не чув”, “забув”, “не дали завдання” тощо;

д) підвищити рівень індивідуалізації (не лише за рахунок розробки індивідуальних варіантів завдань): викладач “бачить загальну картину” роботи студента, що дозволить зробити необхідні висновки та вчасно надати допомогу студентові (наприклад, фіксація часу роботи у системі надасть можливість викладачеві переконатись, що студент працював, але результат не отримав, отже, в нього виникли проблеми, на яких він сам може не наголосити), скорегувати процес.

Звичайно, така організація роботи буде ефективною лише при активній діяльності студента та ретельній підготовці викладача: розробки системи завдань, грамотного планування самостійної роботи з врахуванням розкла-

ду, можливостей, базової підготовки і т.д. студентів, матеріально-технічної бази університету, можливостей бібліотечних фондів тощо. Важливою також є мотивація студентів до роботи, управління, контроль, вчасне надання консультаційної допомоги, власне зацікавлення викладача.

Отже, вважаємо дослідження у даній галузі перспективним напрямком та пропонуємо представлену модель організації самостійної роботи до використання не лише при вивченні курсу “Програмне забезпечення ЕОМ”.

Література:

1. Алексюк А.М. Педагогіка вищої освіти України.– К. :Либідь, 1998. – 560 с.
2. Беляева А. Управление самостоятельной работой студентов // Высшее образование в России, 2003. – №6. – С. 105-109.
3. Дайри Н. О сущности самостоятельной работы // Народное образование, 1963. – №5. – С. 29-34.
4. Козаков В.А. Самостоятельная работа студентов и ее информационно-методическое обеспечение: Учеб. пособие. – К.: Выща шк., 1990. – 248 с.
5. Миронова О.І. Самостійна робота студентів в умовах системної інтеграції інформаційних технологій // Теорія та методика навчання математики, фізики, інформатики: Збірник наукових праць. Випуск 4: В 3-х томах. – Кривий Ріг: Видавничий відділ НМетАУ, 2004. – Т. 3: Теорія та методика навчання інформатики. – С. 193-199.
6. Миронова О.І. Самостійна робота студентів у процесі вивчення курсу “Програмне забезпечення ЕОМ” // Теорія та методика навчання математики, фізики, інформатики: Збірник наукових праць. Випуск V: В 3-х томах. – Кривий Ріг: Видавничий відділ НМетАУ, 2005. – Т. 3: Теорія та методика навчання інформатики. – С. 196-200.
7. Основні засади розвитку вищої освіти України в контексті Болонського процесу (документи і матеріали 2003-2004 рр.) // За редакцією В.Г. Кременя. – Київ-Тернопіль, 2004. – 147 с.

З ДОСВІДУ ВПРОВАДЖЕННЯ КРЕДИТНО-МОДУЛЬНОЇ СИСТЕМИ ВИКЛАДАННЯ ДИСЦИПЛІНИ “ІНФОРМАТИКА ТА КОМП’ЮТЕРНА ТЕХНІКА”

В.О. Воронов

м. Хмельницький, Хмельницький економічний університет

voron@master.km.ua

Сучасний етап розвитку людства характеризується глобалізацією та інтеграцією світових соціально-економічних процесів, формуванням єдиного інформаційно-освітнього простору, інтернаціоналізацією освітнього ринку. Поява і посилення цих ознак – результат переходу від постіндустріального до інформаційного суспільства.

Процеси європейської інтеграції охоплюють дедалі більше сфер життєдіяльності, в тому числі й вищу освіту. Україна, чітко визначивши орієнтир на входження в освітній і науковий простір Європи, здійснює модернізацію освітньої діяльності в контексті європейських вимог, дедалі наполегливіше працюючи над практичним приєднанням до Болонського процесу. [3]

Болонською декларацією, яку підписали європейські міністри освіти 1999 року, передбачено реалізацію багатьох ідей і перетворень. Передусім, це формування єдиного, відкритого європейського простору у сфері освіти, впровадження кредитних технологій на базі європейської системи трансферу кредитів, стимулювання мобільності і створення умов для вільного пересування студентів, викладачів, науковців у межах європейського регіону, прийняття системи освітньо-кваліфікаційних рівнів, яка складається з двох циклів, спрощення процедури визнання кваліфікацій, що сприятиме працевлаштуванню випускників і студентів на європейському ринку праці, розвитку європейської співпраці в сфері контролю за якістю вищої освіти та підсилення європейського виміру освіти. [4]

В процесі навчання інформатики на першому курсі традиційно постає проблема вирівнювання знань студентів. Завданням викладача є необхідність зацікавити кращих студентів, які фактично знайомі з основами інформатики, та познайомити з ними сільських вступників.

Важливу роль тут відіграє вміння викладача дати посильні завдання для студентів та прозора система оцінок, що надає студентам можливість реально оцінювати власні знання та при потребі працювати додатково, щоб наздогнати товаришів. Впровадження кредитно-модульної системи дозволяє:

- більш раціонально поділити навчальний матеріал на модулі;
- використати більш широку шкалу оцінок;
- стимулювати систематичну роботу студентів протягом навчального семестру;
- підвищити об’єктивність оцінювання знань студентів.

**Розподіл балів оцінювання знань студентів по модулях з дисципліни
“Інформатика та комп’ютерна техніка”**

I семестр

МОДУЛІ												Сума балів
Модуль № 1 Теоретичні основи інформатики та комп’ютерна техніка				Модуль № 2 Операційні системи				Модуль № 3 Сервісні програми				
БАЛИ												
Тема № 1	Тема № 2	Тема № 3	Тест по модулю	Тема № 4	Тема № 5	Тема № 6	Тест по модулю	Тема № 7	Тема № 8	Тема № 9	Тест по модулю	
5	5	5	20	5	5	5	20	5	5	5	15	100

II семестр

МОДУЛІ												Сума балів
Модуль № 4 Експертні й навчальні системи				Модуль № 5 Комп’ютерні мережі й телекомунікації				Модуль № 6 Глобальна співдружність комп’ютерних мереж Інтернет				
БАЛИ												
Тема № 10	Тема № 11	Тема № 12	Розрахункова робота	Тема № 13	Тема № 14	Тема № 15	Тест по модулю	Тема № 16	Тема № 17	Тема № 18	Творче завдання	
5	5	5	25	5	5	5	15	5	5	5	15	100

III семестр

МОДУЛІ												Сума балів			
Модуль № 7 Системи оброблення тексту				Модуль № 8 Системи табличного оброблення даних				Модуль № 9 Системи управління базами даних			Модуль № 10 Технологія використання комп’ютерної техніки в економіці і менеджменті				
БАЛИ															
Тема № 19	Тема № 20	Тема № 21	Творче завдання	Тема № 22	Тема № 23	Тема № 24	Розрахункова робота	Тема № 25	Тема № 26	Тема № 27	Контрольна робота	Тема №28	Тема №29	Творче завдання	
5	5	5	15	5	5	5	15	5	5	5	10	5	5	5	100/40

Предмет “Інформатика та комп’ютерна техніка” в Хмельницькому економічному університеті вивчається 3 семестри, які розбиті на 10 модулів: I семестр – 3 модулі, II семестр – 3 модулі та III семестр – 4 модулі. Кожний модуль завершається модульним контролем.

I та II семестр є заліковим, в кінці III семестру студенти складають іспит за весь курс.

Вивчення кожного модуля завершуються певним етапом контролю, таким як контролюючий тест, контрольна робота, творча робота, розрахункова робота.

При тестуванні викладач має змогу використовувати будь-яку доступну тестову платформу, яка дозволяє ввести базу запитань та задати довільну оцінюючу шкалу, або прив’язати до шкали тесту.

Контрольна, творча та розрахункова роботи складається з двох-п’яти завдань. Робота оцінюється за тривірневою шкалою за наступними критеріями.

<i>Бал</i>	<i>Критерій оцінки рівня засвоєння (діяльності)</i>
10 (5)	<p><i>Усна відповідь:</i> студент має початковий рівень знань, може механічно відтворити навчальний матеріал без логічних висновків, має елементарні нестійкі практичні навички роботи на комп’ютері, за допомогою викладача може виконати просте навчальне завдання репродуктивного характеру.</p> <p><i>Практичні навички:</i> відтворення навчального матеріалу механічне, без логічних висновків. Студент має елементарні навички роботи з операційною системою, дисками, прикладною системою, глобальною мережею Інтернет, елементарні поняття про алгоритми та їх призначення, виконує 45% від загальної кількості тестових завдань, самостійне опрацювання матеріалу викликає значні труднощі.</p>
15 (10)	<p><i>Усна відповідь:</i> студент може застосовувати вивчений матеріал у стандартних ситуаціях, пояснити суть стандартних процесів, що відбуваються при роботі інформаційної системи.</p> <p><i>Практичні навички:</i> студент орієнтується в основних можливостях операційної системи та її оболонок. Він може пояснювати суть стандартних процесів, що відбуваються при роботі операційної системи; уміє оперувати з файлами та каталогами; орієнтується у можливостях роботи з дисками; має навички використання програмного забезпечення, призначене для роботи з глобальною мережею Інтернет; може пояснювати суть стандартних процесів, що відбуваються при роботі в глобальній мережі Інтернет; має практичні навички роботи. Студент знає навчальний матеріал та вміє вирішувати задачі на рівні бібліотечних алгоритмів, може покроково пояснювати роботу таких алгоритмів і програм, виконує 75% від загальної кількості тестових завдань.</p>

Бал	<i>Критерій оцінки рівня засвоєння (діяльності)</i>
25 (15)	<p><i>Усна відповідь:</i> студент вільно висловлює власні судження, переконливо аргументує їх, визначає порядок особистої навчальної діяльності, відповідно до поставленої викладачем мети самостійно оцінює результати практичної роботи, без допомоги знаходить джерела інформації і використовує їх відповідно до мети і завдань власної пізнавальної діяльності; використовує набуті знання і вміння в нестандартних ситуаціях, уміє розв'язувати задачі та вправи обов'язкового і підвищеного рівня; вільно оперує науковою термінологією.</p> <p><i>Практичні навички:</i> студент може самостійно порівнювати навчальні знання з реальними життєвими явищами і фактами; уміє налагоджувати інформаційну систему на різні режими роботи; без допомоги викладача використовує одержані відомості відповідно до мети та завдань власної пізнавальної діяльності; оперує файлами та спеціальними програмами для ефективного використання інформаційних систем у нестандартних ситуаціях; знаходить можливість ефективного застосування прикладної системи в нестандартних ситуаціях; знаходить можливість ефективного застосування глобальної мережі Інтернет у нестандартних ситуаціях; може самостійно планувати складання алгоритмів і програм для вирішення нестандартних навчальних та життєвих задач; виконує 100% тестових завдань.</p>

Література:

1. www.mon.gov.ua
2. www.tspu.edu.ua
3. Вища освіта України і Болонський процес// Навчальна програма. – Київ–Тернопіль: Вид-во ТДПУ ім. В. Гнатюка, 2004. – 18 с.
4. Основні засади розвитку вищої освіти України в контексті Болонського процесу (документи і матеріали 2003–2004 рр.) / За редакцією В.Г. Кременя. Авторський колектив: М.Ф. Степко, Я.Я. Болубаш, В.Д. Шинкарук, В.В. Грубінко, І.І. Бабин. – Тернопіль: Вид-во ТДПУ ім. В. Гнатюка, 2004. – 147 с.

ОСОБЛИВОСТІ ВИКЛАДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ “ОСНОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ” ЗА ВИМОГАМИ КРЕДИТНО-МОДУЛЬНОЇ СИСТЕМИ НАВЧАННЯ

С.Р. Красильников, Г.В. Красильникова
м. Хмельницький, Хмельницький національний університет
krasil@ukr.net

Вивчення комп'ютерних технологій у вищій школі України почалося із середини 70-х років у рамках курсів “Основи обчислювальної техніки” та “Програмування”. Висока вартість обчислювальної техніки у той час та її недосконалість не давали змоги до масового поширення комп'ютерних знань.

Точкою відліку появи нових комп'ютерних технологій вважається урядова постанова “Про заходи щодо забезпечення комп'ютерної грамотності учнів середніх навчальних закладів і широкого впровадження електронно-обчислювальної техніки в навчальний процес”, прийнята в 1985 році [1]. Ця постанова передбачала введення в 9-10 класах загальноосвітніх шкіл нового предмета “Основи інформатики та обчислювальної техніки”. Почалося оснащення закладів середньої та вищої освіти комп'ютерними класами і з'явилися передумови до впровадження Концепції інформатизації освіти (1988 р.) із розвитком нових інформаційних технологій у всіх сферах суспільства.

Перспективи розвитку України в умовах інформатизації суспільства відображені у прийнятому Верховною Радою Законі України “Про Національну програму інформатизації” (1998 р.). Основні проблеми і напрямки інформатизації освіти відображено у багатьох державних документах: законах України “Про освіту” (1991 р.), “Про загальну середню освіту” (1999 р.), “Про професійно-технічну освіту” (1998 р.), “Про вищу освіту” (2002 р.), “Концепції програми інформатизації загальноосвітніх навчальних закладів, комп'ютеризації сільських шкіл” (2000 р.) [2]. Верховною Радою України у 2005 році прийнята національна програма інформатизації на 2006-2008 роки [3].

В рамках процесу інформатизації суспільства у Хмельницькому національному університеті десять років тому відбулася організація інформаційно-комп'ютерного центру із створенням локальної мережі університету, до якої підключено більше тисячі персональних комп'ютерів. Локальна мережа має чотири канали виходу в глобальну мережу Internet із сумарною пропускною здатністю 768 Кбіт/с. Такий шлях розвитку університету дозволив вийти на новий рівень використання інформаційних технологій в навчальному процесі і підняти рівень підготовки майбутніх фахівців з усіх спеціальностей.

Особливе значення комп'ютерна грамотність випускників відіграє при

вступі на спеціальності напряму 0915 “Комп’ютерна інженерія”. Від її рівня залежить якість підготовки майбутніх фахівців даного профілю. Тому в навчальний план підготовки фахівців спеціальності “Комп’ютерні системи та мережі” було введено пропедевтичний курсу “Основи інформаційних технологій”. Метою курсу є надання загальних відомостей про інформаційні технології та перспективи їх використання у майбутній професійній діяльності та формування умінь і навичок роботи в мережі Internet із використанням технології “клієнт-сервер”.

Навчальна дисципліна вивчається студентами I курсу: загальне навантаження складає 135 годин, тижневе навантаження – по 2 години відповідно лекційного та лабораторного курсу. Підсумкова формою контролю – іспит.

У зв’язку із впровадженням кредитно-модульної системи організації навчального процесу виникає необхідність перегляду вищезазначеного курсу відповідно до вимог сучасних технологій навчання. Так, зокрема, європейська система освіти велику увагу надає самостійній роботі студентів та її методичному забезпеченню. Доля аудиторної роботи в західноєвропейських країнах коливається від 1/3 до 1/4. У вітчизняних навчальних планах це співвідношення доведено до 1/2. За таких умов виникає потреба аналізу змісту дисципліни та його ретельного відбору для формування лекційних та лабораторних занять. Виконання структурування курсу дозволяє виділити два змістових модуля, наведених у таблиці 2.

Кількість годин, що відведена на самостійну роботу студента, використовується на такі види робіт: опрацювання лекційного матеріалу, підготовка до лабораторних робіт, розробка, створених студентом персонального статичного Web-сайту і динамічних HTML-додатків. Решта годин самостійної роботи потрібна студенту для підготовки до різних форм контролю з курсу: поточного у формі допуску до лабораторних робіт, модульного – у формі тестового контролю і захисту самостійної роботи. Підсумкова оцінка з курсу виставляється за рейтингом з урахуванням всіх видів робіт в семестрі. Розподіл балів, що присвоюються студенту наведено в таблиці 3.

Оцінювання результатів навчальних досягнень студентів проводиться за стобальною шкалою. Співвідношення вітчизняної 4-х бальної та 100-бальної шкал оцінювання наведено в таблиці 1.

Табл. 1. Співвідношення національної 4-х та 100-бальної шкал оцінювання

Оцінка ESTS	Бал	Національна чотирибальна шкала	
A	90-100	Відмінно	5
BC	75-89	Добре	4
DE	60-74	Задовільно	3
FX	35-59	Незадовільно з можливістю повторного складання іспиту	2
F	1-34	Незадовільно з обов’язковим повторним курсом	

Табл. 2. Структура залікових кредитів дисципліни

Теми	Лекції	Лабораторні роботи	Самостійна робота студента	Форми самостійної роботи студена
Змістовий модуль 1. Мова HTML. Основні етапи створення статичного WEB-сайту				
Тема 1. Вступ у дисципліну “Основи інформаційних технологій”	2	-	4	розробка персонального сайту
Тема 2. Структура мови HTML	8	8	16	
Тема 3. Основні етапи створення статичного WEB-сайту	6	8	15	
Всього за модуль	16	16	35	
Змістовий модуль 2. Мова JAVASCRIPT. Інтерактивна технологія “Клієнт-сервер”				
Тема 4. Підходи до створення WEB-додатків	2	-	4	розробка динамічних Web-додатків
Тема 5. Структура мови JAVASCRIPT	8	8	16	
Тема 6. Інтерактивна технологія “Клієнт-сервер”	4	8	12	
Тема 7. Технологія побудови DHTML-додатків	2	-	4	
Всього за модуль	16	16	36	
Разом	32	32	71	

Табл. 3. Розподіл балів, що присвоюються студенту

Модуль I Аудиторна робота, контрольні заходи			Модуль II Самостійна робота студента			Сума		
Змістовий модуль 1		Змістовий модуль 2	Завдання 1: розробка персонального сайту	Завдання 2: розробка динамічних Web-додатків				
40		40	10	10	100			
ЗЛР	УО	К				ЗЛР	УО	К
24	10	6				24	10	6

ЗЛР – захист лабораторних робіт;

УО – усне опитування;

К – колоквіум.

Отже, підсумкова оцінка за дисципліну виставляється в індивідуальний навчальний план студента у вигляді суми балів та відповідних їм оцінок за шкалою ESTS та 4-х бальної вітчизняної системи.

Зміна організації навчального процесу та запровадження кредитно-модульної технології навчання вимагає поряд з традиційними формами (лекціями, лабораторними та практичними роботами) використовувати сучасні інтерактивні технології навчання, удосконалювати методи і форми самостійної роботи студентів, розширювати доступ до різних інформаційних джерел та спрямовувати процес навчання на формування компетенції майбутніх фахівців адаптованих до вимог ринку праці.

Література:

1. Освітні технології: Навч.-метод. посіб. / О.М. Пехота, А.З. Кіктенко, О.М. Любарська та ін.; За ред. О.М. Пехоти. – К.: А.С.К., 2001. – 256 с.
2. Підготовка майбутнього вчителя до впровадження педагогічних технологій: Навч. посіб. / О.М. Пехота, В.Д. Булак, А.М. Старева, К.Ф. Нор та ін.; За ред. І.А. Зююна, О.М. Пехоти. – К.: А.С.К., 2003. – 240 с.
3. <http://www.rada.gov.ua>.

ВПРОВАДЖЕННЯ МОДУЛЬНИХ ТЕХНОЛОГІЙ НАВЧАННЯ

С.В. Говаленков

м. Харків, Академія цивільного захисту України
fd.apbu@list.ru

Згідно наказів МОН України від 23.01.04 р. №48 “Про проведення педагогічного експерименту з запровадження кредитно-модульної системи організації навчального процесу”, №49 “Про затвердження Програми дій щодо реалізації положень Болонської декларації в системі вищої освіти і науки України на 2004-2005 роки” та рішення методичної ради Академії цивільного захисту України від 17.06.04 р., професорсько-викладацьким складом АЦЗУ, фахівцями науково-методичного центру МНС України розроблені методичні рекомендації щодо впровадження кредитно-модульної системи у навчально-виховний процес вищих навчальних закладів МНС України.

Кафедрою інформатики та обчислювальної техніки АЦЗУ протягом навчального року впроваджується експеримент по застосуванню в навчальний процес кредитно-модульної системи в потоках курсантів і студентів з дисциплін кафедри. Алгоритм впровадження складається з наступних блоків: перший – планування, другий – розробка методичного забезпечення, третій – організаційний, четвертий – аналітичний.

Розглянемо результати впровадження модульних технологій навчання з дисципліни „Системний аналіз”.

При плануванні основна увага приділялась розробці робочої навчальної програми та тематичного плану, з урахуванням того, що на вивчення дисципліни відводиться 54 години, що відповідає одному кредиту у системі вищої освіти України. Цей час було розбито на 2 модулі (кожний по 0,5 кредиту). Вивчення навчального матеріалу завершується складанням іспиту, але у разі якісного виконання індивідуальних завдань студенту зараховується відповідна оцінка без складання іспиту.

Розробка методичного забезпечення. Для якісного вивчення дисципліни кожний студент отримує на паперовому носії або в електронному виді: курс лекцій, навчальний посібник “Системний аналіз в задачах і прикладах”, “Методичні вказівки до виконання розрахунково-графічної роботи”, розроблено дистанційний курс з дисципліни.

Організація вивчення дисципліни здійснюється на основі виконання студентом індивідуального навчального плану, який полягає у виконанні однієї розрахунково-графічної роботи (індивідуального завдання), яка містить 6 завдань по змісту двох модулів. Індивідуальний навчальний план складається студентом за допомогою викладача-куратора. Зарахування модулів, включених в індивідуальний навчальний план, здійснюється за результатами виконання завдань розрахунково-графічної роботи.

Аналіз результатів. Результати осіннього семестру дають змогу підвести попередні підсумки. Позитивним є те, що курсанти та студенти більш зацікавлено і активно вивчали дисципліну протягом семестру. За рахунок відмінного складання модулів отримали екзаменаційну оцінку без складання іспиту 44% курсантів і студентів, середній бал склав 4,23, відсоток якості – 81%, відсоток абсолютної успішності – 93%. Таким чином, аналіз показує, що курсанти і студенти досить швидко адаптуються до нової системи організації навчального процесу, і ті з них, хто дійсно хоче планомірно виконувати навчальний план має змогу робити це ефективно, при цьому оцінка знань стає більш об'єктивною.

Перехід від стандартної 4-бальної оцінки успішності студента до оцінок шкали ECTS не викликає труднощів. Співвідношення між оцінками за шкалою ECTS та шкалою вузу може мати наступний вигляд:

За шкалою ECTS	За національною шкалою	100-бальна шкала	12-бальна шкала	5-бальна шкала
A	відмінно	90-100	12	5
BC	добре	75-89	9-11	4
DE	задовільно	60-74	6-8	3
FX	незадовільно з можливістю повторного складання	35-59	3-5	2
F	незадовільно з обов'язковим повторним курсом	1-34	1-2	1

Якщо у вузі прийнято 100-бальну шкалу оцінювання (дозволяє враховувати нерівноцінність контрольних заходів), то необхідно кожній формі контрольних заходів, у залежності від її складності, призначити максимальний бал таким чином, щоб за підсумком виконання всіх контрольних заходів, що передбачаються під час вивчення модулю, максимальна кількість отриманих балів не перевищувала 100. Наприклад, для навчальної дисципліни “Системний аналіз”:

Модуль 1:

- за виконання 1 завдання розрахунково-графічної роботи – до 25 балів,
 - за виконання 2 завдання розрахунково-графічної роботи – до 35 балів,
 - за виконання 3 завдання розрахунково-графічної роботи – до 40 балів,
- Всього 100 балів.

Нехай при вивченні наведеного модуля студент отримав наступні бали: за виконання 1 завдання – 20 балів, 2 завдання – 25 балів, 3 завдання – 40 балів, всього 85 балів. Оцінка за модуль за шкалою ECTS – **В** (“дуже добре”).

Під час використання 12 або 5-бальної шкали оцінка за модуль може бути отримана наступним чином: за виконання 1 завдання – 4 бали, 2 завдання – 5 балів, 3 завдання – 5 балів. Таким чином, підсумкова оцінка за модуль становить 4,7, що відповідає **A** (“відмінно”) за шкалою ECTS.

Аналогічно можна отримати оцінку за модуль за 12-бальною шкалою.

Підсумкова оцінка за вивчення дисципліни повинна враховувати оцінки за кожен модуль і може бути отримана наступним чином:

1) якщо всі модулі рівноцінні, то підсумкова оцінка буде являти собою середню арифметичну оцінок за кожен модуль;

2) якщо кредити нараховуються модулям в залежності не тільки від часу, що потрібен на їх засвоєння, але й від складності модуля (тобто призначена кількість кредитів, перш за все, відбиває складність певного модуля), то підрахувати підсумкову оцінку (ПО) можна за формулою:

$$ПО = \frac{\sum_{i=1}^N O_i \cdot K_i}{\sum_{i=1}^N K_i},$$

де N – кількість модулів певної дисципліни; O_i – оцінка за певний модуль; K_i – кількість кредитів, призначених на вивчення певного модуля;

3) у іншому випадку викладач сам встановлює методіку отримання підсумкової оцінки.

Таким чином, кредитно-модульна система організації навчального процесу дає можливість студенту, у випадку отримання відповідної підсумкової оцінки, не складати підсумкового контролю з певної навчальної дисципліни. Якщо отримана підсумкова оцінка не задовольняє студента, він може її покращити під час складання підсумкового контролю з навчальної дисципліни. Якщо студент отримав незадовільну оцінку за виконання певного контрольного заходу (або зовсім його не виконував) під час вивчення відповідного модуля, то цей модуль йому не зараховується. Якщо не зараховано хоча б один модуль з певної навчальної дисципліни, то студент отримує незадовільну оцінку за вивчення всієї дисципліни

До недоліків системи на етапі її впровадження слід віднести велику кількість часу, який необхідно витратити викладачу на розробку “контрольних-тестів”, та їх перевірку по кожному модулю.

Висновки.

Модульність визначає підхід до організації оволодіння курсантом (студентом) змістовими модулями і проявляється через специфічну для модульного навчання організацію методів й прийомів навчально-виховних заходів, основним змістом яких є активна самостійно-творча пізнавальна діяльність. Тому найважливішим етапом системи є якісна самостійна робота, саме цей елемент навчально-виховного процесу необхідно посилити, зробити дійсно якісним.

Для викладачів вкрай необхідно зменшити навчальне навантаження, інакше система буде неефективною.

ТЕСТИРОВАНИЕ КАК СРЕДСТВО ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ЛЕКЦИОННОГО ПРОЦЕССА

Н.А. Кириенко

Беларусь, г. Минск, Объединенный институт проблем информатики
Национальной академии наук Беларуси
kir@newman.bas-net.by

Тесты являются эффективным средством проверки качества знаний, получаемых студентами, и оперативного контроля хода обучения.

Хороший эффект тестирования достигается при совместном использовании различных видов электронных учебных материалов [1]. Так, встраивание средств тестирования в состав электронных учебников и электронных учебных пособий позволяет оперативно оценивать ход изучения теоретического материала студентом.

Входное (до начала изучения курса) тестирование позволяет оценить начальный уровень знаний студента. Выходное тестирование позволяет либо выставить оценку по итогам изучения дисциплины, либо служит для допуска студента к сдаче экзамена (зачета) в традиционной форме.

Полезно встраивать средства тестирования в состав лабораторных практикумов. Входное тестирование в этом случае проверяет понимание студентами основ теоретического материала по теме лабораторной работы и служит для допуска к ее выполнению. Итоговое тестирование по результатам выполнения лабораторных практикумов позволяет проверить знания, приобретенные студентом в результате выполнения лабораторной работы или блока работ.

Существующие системы тестирования, можно разбить на две категории:

1. Тесты, выполняемые студентами в письменной форме (без использования компьютера) с дальнейшей проверкой вручную преподавателем.
2. Системы компьютерного тестирования с соответствующим наполнением тестовыми материалами.

Преимущества второй категории систем очевидны. Они позволяют освободить преподавателя от рутинной работы при проведении экзаменов и промежуточной оценке знаний в традиционном учебном процессе, а при обучении с использованием дистанционных технологий становятся основным средством контроля.

В некоторых случаях системы тестирования первой категории оказываются незаменимыми. В настоящей работе предлагается использование таких тестов для повышения эффективности лекционного процесса.

Процесс прослушивания лекции студентами – сложный психологический и эмоциональный процесс, проходящий различные фазы развития. Так, например, в начале лекции студенты обычно внимательны, сосредоточены,

легко усваивают материал. По мере накопления предоставляемой информации, а может быть и ее усложнения, внимание начинает рассеиваться. В такие моменты необходимы определенные приемы лектора направленные на снятие напряжения, переключение внимания, чтобы интерес к материалу не был потерян. В качестве таких приемов могут служить обобщение или повторение только что прочитанного материала, обращение к аудитории с какими-либо вопросами по теме, и т.д.

Хорошим приемом концентрации внимания и повышения интереса слушателей на лекции можно считать brief-тестирование слушателей по материалу последних прочитанных лекций. Под brief-тестированием понимается выполнение студентами кратких тестов (до 20 вопросов), ответы на которые состоят из одного-двух слов или да/нет. Вопросы могут быть написаны на доске, показаны на экране с помощью мультимедийных устройств, или заготовлены заранее на бланке. На тестирование отводится не более 15 минут. Ответы на листах отдаются лектору и позволяют провести текущую оценку усвоения материала. Опыт показал, что такое тестирование очень интересно студентам. Они охотно выполняют задание, а затем с нетерпением ждут результатов. В дальнейшей работе они становятся более активными и ответственными, восстанавливается интерес к излагаемому материалу.

Brief-тестирование позволяет определить разделы, которые представляют наибольшую сложность в изучении, и, возможно, корректировать процесс обучения в зависимости от результатов тестирования.

Состав тестовых вопросов должен отражать материал последних лекций, вопросы могут быть не слишком сложными.

При подготовке тестовых заданий необходимо определиться с выбором типов тестовых вопросов.

К наиболее распространенным относятся следующие:

- 1) закрытый тип вопроса с одним правильным ответом (альтернативный выбор);
- 2) закрытый тип вопроса с несколькими правильными ответами (многоальтернативный выбор);
- 3) вопрос на упорядочивание;
- 4) свободный ввод краткого числового или текстового ответа;
- 5) вопрос типа «заполнить пробелы» (вопрос на подстановку);
- 6) вопрос на соответствие нескольких частей ответов.

Brief-тестирование допускает любой из этих типов вопросов, хотя предпочтение следует отдать 1 и 4 вариантам.

При подготовке тестовых заданий необходимо уделять внимание предполагаемой форме представления вопроса. Желательно обеспечить их соответствие стандартам тестирования, наиболее распространенным среди которых является IMS QTI [2]. Данный стандарт определяет требования к тестовым материалам, сценариям тестирования, представлению результатов тестирования. Можно сказать, что спецификация IMS QTI при определении

типов тестовых заданий определяет как собственно тип, так и способ его представления пользователю, а также ограничения на типы используемых в вопросе данных. Так, например, классический закрытый вопрос с выбором одного правильного варианта в IMS QTI разбивается на:

1) Standard True/False (Text) – выбор из двух вариантов ответа: Правда/Ложь (текст);

2) Standard Multiple Choice (Text) – альтернативный выбор (текст);

3) Standard Multiple Choice (Images) – альтернативный выбор (изображения);

4) Multiple Choice with Image Hot Spot Rendering – альтернативный выбор (точки изображения);

5) Multiple Choice with Slider Rendering – альтернативный выбор (слайдер).

В условиях лекционного процесса приемлемыми вариантами являются первый и второй.

При оценке результатов тестирования имеет смысл учитывать не только правильные, но и частично правильные ответы. Например, если за правильный ответ засчитывать 2 балла, то за частично правильный – 1 балл.

Предлагаемый подход был опробован при чтении лекций по дисциплине «Визуальные средства разработки приложений» студентам 3 курса Белорусского государственного университета информатики и радиоэлектроники. Вышеназванная дисциплина входит в учебный план специальности «Информационные системы и технологии (в экономике)» [3], которая открыта на инженерно-экономическом факультете. Подготовка по ней осуществляется кафедрой экономической информатики.

В результате изучения дисциплины «Визуальные средства разработки приложений» студенты должны научиться самостоятельно разрабатывать Windows-приложения, проектировать информационные системы с использованием визуальных средств обработки данных. В качестве средства разработки используется язык C++, библиотека классов Microsoft Foundation Classes (MFC) и среда Microsoft Developer Studio.

Программа курса «Визуальные средства разработки приложений» (34 часа лекций, 34 часа лабораторных работ, 17 часов практических занятий) включает следующие основные разделы: технология проектирования приложений под Windows; обзор среды Microsoft Visual Studio; программирование Windows-приложений на языке C++ и библиотека MFC; архитектура Document-View, SDI- и MDI-приложения; сохранение-восстановление объектов и работа с файлами; диалоги, классы окон, элементы управления; доступ к базам данных (технологии ODBC и OLE DB), введение в технологии OLE и ActiveX.

В компьютерных классах студентам предлагается выполнить 8 лабораторных работ и 5 индивидуальных заданий. Индивидуальные задания выдаются по материалам лабораторных работ.

Основные трудности, которые возникают у студентов при изучении курса, связаны с освоением и восприятием довольно большого количества новых для них терминов и понятий. Поэтому при чтении лекций необходимо особое внимание уделять многократному изучению основных понятий стандартов программирования под Windows в различных контекстах.

Большое значение для эффективности усвоения материала имеет brief-тестирование во время лекций. За семестр удастся провести 2-3 тестирования. Ниже приведены примеры некоторых тестов.

Тесты по теме «Архитектура Document-View, SDI- и MDI-приложения»

	<i>Вопрос</i>	<i>Ответ</i>	<i>Оценка</i>
1	Какой класс называется главным классом приложения?		
2	Какой виртуальный метод класса CWinApp обязательно должен быть переопределен в производном от него классе?		
3	Для чего предназначена таблица сообщений класса?		
4	Для чего в заголовке таблицы сообщений класса указывается имя базового класса?		
5	Какая макрокоманда должна присутствовать в объявлении класса для того, чтобы этот класс мог обрабатывать сообщения?		
6	На базе каких классов MFC создаются документы и представления?		
7	Может ли один документ иметь несколько представлений?		
8	С помощью какого метода объект-представление может получить доступ к объекту-документу, чьи данные он отображает?		
9	Какой метод класса документа обеспечивает синхронизацию отображения данных для всех своих обликков?		
10	Перечислить классы, создаваемые для однодокументного приложения AppWizard.		

Тесты по теме «Диалоговые окна и элементы управления»

	<i>Вопрос</i>	<i>Ответ</i>	<i>Оценка</i>
1	Для чего предназначен механизм автоматического обмена данными (DDX) и механизм автоматической проверки данных (DDV)?		
2	Какой виртуальный метод класса CDialog должен быть переопределен для реализации механизмов DDX и DDV?		
3	Вызов какого метода (и с каким параметром) обеспечивает инициализацию диалоговой панели, при кото-		

	<i>Вопрос</i>	<i>Ответ</i>	<i>Оценка</i>
	рой информация из переменных класса отображается в элементах управления диалоговой панели?		
4	Создается ли Windows-окно диалога при создании объекта класса CDialog?		
5	Каким методом создается Windows-окно модального диалога и когда этот метод возвращает управление?		
6	Удаляется ли при этом: 1) Windows-окно диалога, 2) объект класса CDialog?		
7	В каком классе размещаются вызовы методов, обеспечивающих процесс обмена и проверки данных?		
8	Какой класс является родительским для всех классов диалоговых панелей?		

Проводимые brief-тестирования оживляют лекционный процесс, повышают интерес у студентов к предмету, позволяют преподавателю оценить работу студентов и определить материал, который плохо усвоен.

Литература:

1. Башмаков А.И., Башмаков И.А. Разработка компьютерных учебников и обучающих систем. – М.: Информационно-издательский дом «Филинь», 2003.
2. IMS Question & Test Interoperability: ASI Information Model Specification.
3. Комличенко В.Н., Поттосина С.А., Кириенко Н.А. О потенциале информационных технологий и математическом образовании специалиста по экономической информатике // Теорія та методика навчання математики, фізики, інформатики: Збірник наукових праць. Випуск V: В 3-х томах. – Кривий Ріг: Видавничий відділ НМетАУ, 2005. – Т. 3: – С. 108–113.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ДИАГНОСТИКИ СЛУШАТЕЛЕЙ ПРИ ПРЕПОДАВАНИИ ИНФОРМАТИКИ

А.В. Шматко, Е.А. Панина
г. Харьков, Академия гражданской защиты Украины
fd.apbu@list.ru

Адекватное управление учебным процессом со стороны преподавателя требует организации диагностической деятельности, которая включает:

- систематический сбор информации о текущих учебных достижениях каждого отдельного слушателя и учебной группы в целом (мониторинг текущей успеваемости); хранение этой информации в форме, удобной для дальнейшей обработки;
- обработка данных мониторинга; представление результатов обработки в форме, удобной для анализа преподавателем;
- анализ подготовленной информации, выработка корректирующих мер, соответствующих учебной ситуации.

Все последующие этапы основываются на информации, полученной в ходе мониторинга текущей успеваемости. Следовательно, адекватность управления определяется полнотой, достоверностью и своевременностью исходных данных. Обеспечение перечисленных требований в традиционном учебном процессе (без применения компьютерных информационных технологий) невозможно, поскольку текущий контроль носит эпизодический и выборочный характер.

Таким образом, преподаватель оказывается заинтересованным в технологиях, которые могли бы обеспечить его, в первую очередь, достоверными сведениями о текущей успеваемости, а также методами их обработки и интерпретации.

В настоящее время значительное внимание уделяется технологизации процедуры итоговой аттестации – все большее распространение получают компьютерные тесты и централизованное тестирование. Не умаляя значимости этих форм контроля, следует, вместе с тем, отметить, что, они фиксируют конечный результат обучения и не могут быть использованы для управления ходом обучения.

Поскольку диагностическая деятельность преподавателя связана с необходимостью формирования и обработкой значительных по объему информационных массивов, безусловно, она должна реализовываться посредством компьютерных технологий. В связи с этим выявляются и требуют решения следующие проблемы:

- разработка и внедрение в практику технологий текущего компьютерного контроля знаний и умений;
- построение информационных систем, обеспечивающих хранение и оперативный доступ к информации о ходе процесса обучения;

- разработка и внедрение статистических методов анализа данных о текущей успеваемости;
- изменение нормативной базы, определяющей порядок оценивания учебных действий учащегося, а также порядок хранения и обработки данных о текущей успеваемости в школе;
- подготовка преподавателей к использованию тестовых форм контроля (и, в частности, компьютерного тестирования) на всех этапах обучения, а также к применению методов статистической обработки и анализа первичных данных.

С точки зрения реального внедрения методов диагностики учебных достижений в практику высшей школы, безусловно, важнейшим оказывается фактор готовности преподавателей к осуществлению этого вида деятельности на основе информационных технологий, что требует целенаправленных усилий по их пропаганде.

К настоящему времени в Академии гражданской защиты Украины (АГЗУ) используется диагностический комплекс, который включает системы компьютерного контроля знаний (сетевую контролируемую систему SunRav Test Office Pro), компьютерные реализации методов накопления и анализа данных текущей успеваемости (накапливаемой отметки, поэлементного анализа, прогнозирования), а также указания и рекомендации по их применению.

Программа SunRav Test Office Pro позволяют проводить тестирование, создавать тесты, осуществлять администрирование. Функционально комплекс состоит из трех основных модулей:

1-й модуль: Программа создания тестов **tMaker** (позволяет создавать собственно вопросы, варианты ответов, критерии оценки, время тестирования, количество вопросов для проведения тестов);

2-й модуль: Программа тестирования **tTester**;

3-й модуль: Программа обработки и анализа результатов тестирования **tAdmin** (просмотр и анализ).

Программа Tester осуществляет:

- выбор различных типов вопросов;
- выбор одного варианта ответа из нескольких вариантов ответа;
- выбор нескольких ответов из нескольких вариантов ответа;
- непосредственный ввод с клавиатуры;
- использование рисунков и любых OLE-объектов непосредственно в вопросах и ответах;
- полноценное форматирование описание теста, вопросов и ответов (каждый символ может быть написан своим шрифтом, цветом и т.д.);
- возможность ограничения времени тестирования.

В зависимости от типа теста (обучение, самоконтроль, экзамен), пользователь может получать комментарии к своим ответам.

Комментарии могут быть четырех видов:

«Ответ правильный» – при правильном ответе.

«Ответ неправильный» – при неправильном ответе.

«Ответ неправильный» + комментарии – при неправильном ответе. Используется, обычно, в тестах, направленных на самоконтроль и обучение.

«Ответ правильный» + комментарии – при неправильном ответе. Используется обычно, в тестах, направленных на самоконтроль.

Допускается сопровождение вопросов мультимедийными файлами.

В экзаменационном режиме пользователю задается заданное количество вопросов из теста в случайном порядке. Это позволяет производить одновременное тестирование всей учебной группы студентов (из расчета 1 студент – 1 компьютер) в аудитории. Не повторяемость тестов и ограничение времени на ответы исключает возможность использования чужих знаний (подсказок).

Пользователю по ходу тестирования может быть предоставлена статистика:

- количество вопросов в тесте;
- текущий вопрос теста;
- количество правильных и неправильных ответов;
- время, оставшееся для тестирования.

Можно проводить и тестирование по различным темам одновременно до 99 тем. В одном тесте может быть несколько тем (разделов). Результат выдается по каждой теме в отдельности, и это выгодно отличает его от тестовой программы TEST2000. Система позволяет сформировать произвольное количество ответов на каждый вопрос. Количество правильных ответов может быть несколько. Последовательность ответов на вопрос меняется при каждом следующем тестировании.

Критерием оценки служат баллы (в процентном выражении), набранные во время тестирования и соответствующие принятой системе дифференцирования знаний – «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

В процессе формирования БД вводится информация, содержащая нормативное время, отведенное на тестирование по конкретной теме и конкретному вопросу. При превышении данного норматива тестирование прекращается. Создание БД сопровождается подсказкой, которую при необходимости можно отключить.

Удобный и простой графический интерфейс системы TEST2000 и SunRav Test Office Pro позволяет без специальной подготовки использовать данные системы при создании баз для различных дисциплин.

Система SunRav Test Office Pro дает возможность созданные ранее базы по отдельным темам объединить в один модульный тест. Эта возможность отсутствует в системе TEST2000, что делает её менее универсальной. В перспективе результаты тестирования могут использоваться при аттестации слушателей, а также при формировании зачетных или экзаменационных

оценок.

Осуществлена надежная защита результатов тестирования от несанкционированного изменения (подделки), благодаря использованию средств криптографии. Применен простой и интуитивно понятный интерфейс.

Программа Test Maker осуществляет: предварительный просмотр теста; отладочный запуск теста в Tester'e; редактирование установок теста по умолчанию; экспорт теста в формат RTF.

В настройках программы можно произвольно установить шрифт вопроса и вариантов ответа, сами варианты ответов, количество баллов за ответ, правильный/неправильный вариант.

Предусмотрено копирование шрифта вопроса и вариантов ответов из предыдущего вопроса во все остальные вопросы.

Программа Test Admin позволяет:

- добавлять и редактировать пользователей;
- автоматическое и ручное создание резервных копий файлов результатов;
- просмотр, печать и редактирование результатов тестирования;
- подробный отчет о прохождении теста.

Пользователю предоставляются статистические данные:

- дата тестирования;
- время начала тестирования;
- время окончания тестирования;
- количество вопросов в тесте;
- количество вопросов, на какие ответил пользователь;
- количество правильных ответов.

Использование перечисленных систем и методов не требует от преподавателя специальной подготовки в вопросах математической статистики и компьютерных технологий – достаточным является уровень пользователя персонального компьютера. Комплекс прошел апробацию на кафедре информатики и вычислительной техники Академии гражданской защиты Украины. Компоненты комплекса успешно осваиваются курсантами и студентами различных специальностей вуза.

Преподавателями кафедры были разработаны тесты по темам: «Основы эксплуатации операционных систем», «Аппаратное и программное обеспечение персональных компьютеров (ПК)», «Создания и обработка простых текстовых документов», «Создания комплексных текстовых документов», «Информационные сети», «Средства подготовки и представления презентаций». Эти тесты установлены во всех компьютерных классах Академии гражданской защиты Украины, что дало возможность не только проводить контроль текущей успеваемости слушателей преподавателями, но и использовать тесты слушателям для самоподготовки и самоконтроля.

На кафедре информатики и компьютерной техники с 2004 года проводится эксперимент по внедрению кредитно-модульной системы. Преподава-

телями кафедры были разработаны и внедрены тесты для сдачи модулей: Модуль 1. «Основы информационных технологий. Операционные системы (ОС)», Модуль 2. «Системы обработки текстовых документов», Модуль 3 «Современные компьютерные технологии обработки данных». Система SunRay Test Office Pro позволила выполнить эту работу за короткие сроки, так как дает возможность использовать ранее разработанные тесты. Аналогично был разработан тест проверки теоретических знаний за весь семестр. В этот тест вошло более 200 вопросов по 3 модулям (6 темам). Тестируемому предлагается ответить на 40 вопросов из теста в случайном порядке, которые охватывают все 6 тем. По результатам тестирования можно определить не только общий уровень знаний, но и по каждой теме в отдельности. Критерием оценки служат баллы (в процентном выражении), набранные во время тестирования и соответствующие принятой системе дифференцирования знаний – «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Этот тест был использован при проведении триместрового экзамена.

Тестовая сдача теоретической части экзамена дает объективную оценку теоретических знаний и исключает влияние личностных взаимоотношений преподаватель-студент.

Применение новых технологий тестирования, при котором весь процесс сдачи теста проходит дистанционно, либо с использованием компьютерных программ, является шагом вперед по сравнению с традиционными методами тестирования, и существенно сокращает как время обработки результатов, так и вероятность ошибок.

ИСПОЛЬЗОВАНИЕ XML ДЛЯ ОРГАНИЗАЦИИ И ХРАНЕНИЯ БАЗЫ ТЕСТОВЫХ ЗАДАНИЙ

В.В. Кравченко

г. Кривой Рог, Криворожский государственный педагогический университет

Современные приложения нуждаются не только в языке представления данных на экране клиента, но и в механизме, позволяющем определять структуру документа, описывать содержащиеся в нем элементы.

Выбор способа организации и хранения данных также является актуальным вопросом и при построении баз тестовых заданий для компьютерных систем тестирования. Чаще всего этот выбор определяется возможностями и средствами разработчика. При этом немаловажную роль играют необходимость защиты информации, возможность гибкого структурирования данных, максимальная оптимизация методов доступа к данным, их интерпретации и хранения.

Легкость в прочтении и открытость структуры данных дает возможность легкого конвертирования в нужный формат, а также редактирования доступными средствами (например, обычным текстовым редактором). Данная проблема особенно актуальна для желающих пополнить либо изменить существующую базу тестовых заданий используемой ими компьютерной системы тестирования, но не имеющих представления о ее структуре или не имеющих необходимых программных средств ее изменения и не владеющих данным инструментом. Конечно, изменение базы тестовых заданий может привести к некорректной работе программы из-за неправильной интерпретации используемых данных. Но разработчикам компьютерных систем тестирования стоило бы указывать в справочной системе программы структуру хранения данных и инструкцию по выполнению возможных ее изменений.

Укажем основные преимущества и недостатки некоторых возможных способов хранения информации определяющей содержание тестовых заданий для компьютерных систем тестирования.

1. Текстовый формат

Отсутствие каких-либо признаков структуры тестового задания, делает необходимость четкого определения количества места и места расположения отведенного на содержание того или иного структурного элемента тестового задания. И хотя данный способ является идеальным для хранения базы простых тестовых заданий, с точки зрения кажущегося отсутствия избыточности информации, следует заметить, что для создания тестовых заданий со сложной динамической структурой он вообще непригоден.

2. Базы данных

Хранения базы тестовых заданий в виде базы данных накладывает целый ряд ограничений на структуру данных (практически аналогичные про-

блемы возникают при хранении информации в типизированных файлах):

- невозможность универсализации структуры данных в случае значительной разнотипности тестовых заданий;

- сложность в описании структур хранящих содержание заданий, вызванная, прежде всего, различным количеством информации, имеющей отношение к ответам (имеется в виду количество вариантов ответов);

- ограничения на вариации с объемом и типом информации, как самих заданий, так и ответов (существенно ограничивается текстовое содержимое заданий, в большинстве случаев отсутствует возможность форматирования внешнего представления текста заданий; во многих системах тестирования отсутствует возможность вставки в задание даже одного статического изображения, аудио- или видеозаписи, анализ которых является необходимым и обязательным условием выполнения задания);

- введение каких-либо дополнительных элементов в структуру тестового задания практически невозможно, в противном случае приводит к трудоемкой и зачастую делающей необходимым полное перестроение базы данных работе, к чему могут оказаться неготовой система тестирования;

Кроме того, работа с базой данных предусматривает наличие соответствующих средств управления – СУБД, что вызывает необходимость установки дополнительного программного обеспечения, отсутствие которого делает базу тестовых заданий закрытой для какого-либо использования, в том числе и редактирования.

К преимуществам можно отнести удобство редактирования данных, однозначность и простота их интерпретирования, легкость конвертирования в другие форматы, и прочие возможности предоставляемые СУБД.

3. INI-файлы

Легкость в организации доступа к отдельным элементам тестового задания, возможность редактирования содержания любым текстовым редактором, минимальное влияние на работу системы тестирования нарушения структуры хранимых данных – несомненные преимущества данного метода.

Но он также имеет и большое количество недостатков. Среди них: отсутствие возможности хранения значений полей в несколько строк (как выход, возможно использование какого-либо символа в качестве разделителя, но это вызывает ряд дополнительных сложностей в процессе программирования, а о поиске чего-то или редактировании такой информации, как о нормальном процессе вообще говорить трудно); отсутствие возможности чтения и записи данных в одном разделе в различных полях с одинаковыми именами, как следствие необходимость договоренности об используемых именах разделов и их полей; хранение в одном файле большого числа вопросов существенно усложняет проблема именования разделов (каждый раздел должен иметь собственное оригинальное имя).

При выборе способа организации хранения базы тестовых заданий следует также помнить, что построение такой базы для компьютерных систем

тестирования требует расширенного анализа структуры хранимых данных, которая не всегда может быть изоморфна структуре тестового задания в условиях реализации организации базы тестовых заданий конкретными средствами. Хотя упомянутая проблема может быть решена за счет введения дополнительных спецификаций или усложнения структуры данных, которые, в свою очередь, потребуют более сложных алгоритмов анализа и управления данными, а также детального описания и изложения (для авторов программных продуктов) причин внесенных изменений. Тем более, что такие изменения не допускаются большинством программ и могут потребовать доработки программных продуктов.

Рассмотрение существующих средств хранения структурированной информации и полемика о выборе одного из них как универсального, неизменно ведут к необходимости выбора или даже разработки и использования определенной разметки, применение которой к информации, содержащей тестовое задание, позволило бы структурировать соответствующие данные для дальнейшего их использования.

Логично было бы использовать в качестве разметки язык, имеющий широкое распространение, открытый формат и возможность к расширению – дополнению собственными структурными элементами.

Таким языком является расширяемый язык разметки (Extensible Markup Language, аббревиатура – XML), который описывает класс объектов XML document, а также частично описывает работу компьютерных программ, обрабатывающих объекты с данными, реализующими этот класс. XML – это прикладной уровень или усеченная форма SGML, Стандартного Обобщенного языка разметки [ISO 8879]. По своему построению XML-документ является полноценным SGML документом [1].

Язык XML был разработан группой XML Working Group (первоначально называемой SGML Editorial Review Board), сформированной в 1996 году под патронажем World Wide Web Consortium (W3C).

При разработке языка XML ставились следующие задачи: 1) XML должен быть пригоден для непосредственного использования в Интернет; 2) иметь широкий круг применения; 3) быть совместим с SGML; 4) обработчики документов XML должны быть просты в написании; 5) количество факультативных свойств в XML должно быть сведено к абсолютному минимуму, в идеале число их вообще должно быть нулевым; 6) XML документы должны быть удобны для чтения и достаточно понятны; 7) подготовка XML документа должна осуществляться быстро; 8) процедура построения XML документа должна быть формальной и точной; 9) процедура создания XML документов должна быть проста; 10) краткость при разметке XML документа имеет минимальное значение.

Данная спецификация в сочетании с остальными связанными с нею стандартами (Unicode и ISO/IEC 10646 для символов, Internet RFC 1766 для тэгов идентификации языка, ISO 639 для кодов с названием языка и ISO

3166 для кодов с названием страны) дает всю необходимую информацию для понимания языка XML (версия 1.0) и создания компьютерных программ для его обработки [1].

Обработкой XML-документа может также заниматься любой анализатор XML-документов, который конвертирует их в необходимый формат представления (например, обычную HTML страницу) или извлекает из нее информацию для других приложений.

Любой XML-процессор, являясь, по сути, транслятором языка разметки, может быть разбит на несколько модулей, отвечающих за лексический, синтаксический и семантический анализ содержимого документа. Синтаксически правильно составленный XML-документ может быть разобран любым универсальным XML-анализатором, и XML-обработчику остается лишь использовать полученные на его выходе «чистые» данные (прошедшие синтаксический анализ) – интерпретировать содержимое документа, в соответствии с его DTD-описанием или схемами данных [3].

Если на компьютере установлен браузер Internet Explorer 4 (или более поздняя версия), то для анализа можно использовать встроенный в этот браузер XML-анализатор msxml в своих сценариях, написанных на Java Script или VBScript. В настоящий момент существуют две его реализации: одна предназначена для использования в виде написанного на C++ ActiveX-объекта (реализация на базе COM-технологии); другая, написанная на Java, не зависит от платформы. Оба анализатора не сложны, имеют сравнительно небольшой размер – msxml на C++ занимает около 100 k, версия на Java – 127 k. Анализатор, написанный на C++, в текущей реализации не поддерживает DTD-правил, более компактный и быстрый, чем его Java-версия. Оба они имеют поддержку иностранных языков, т.е. в составе Internet Explorer C++-анализатор работает со всеми языками, «понимаемыми» браузерами, а анализатор на Java – с теми языками, с которыми может работать виртуальная Java-машина [3]. При разработке приложений с помощью Delphi данный анализатор можно импортировать как активный элемент и работать, как с обычным объектом [4].

Приведем пример варианта использования XML в качестве разметки структуры тестового задания.

```
<?xml version=«1.0» encoding=«windows-1251»?>
<HEAD>
  <TITLE>Пример организации базы тестовых заданий средствами
XML
  </TITLE>
</HEAD>
<BODY>
  <TASK subject=«МАТЕМАТИКА»
        theme=«ТАБЛИЦА СЛОЖЕНИЯ»
        type=«radiobtn»
        level=«low»>
  <T>Сколько будет 2+2?</T>
```

```

    <A>2</A>
    <A result=«true» mark=«100%»>4</A>
  </TASK>
<TASK subject=«ЭТИКА СЕМЕЙНОЙ ЖИЗНИ»
      theme=«СЕМЕЙНОЕ ПОЛОЖЕНИЕ»
      type=«checkboxbtn»
      level=«medium»>
  <T>Так кто в семье главный?</T>
  <A>Я</A>
  <A result=«true» mark=«90%»>Папа</A>
  <A result=«false» mark=«10%»>Мама</A>
  <A>Все</A>
</TASK>
</BODY>

```

Данный пример показывает использование разметки для описания тестового задания с одиночным выбором и множественным. Задание предмета и темы, к которой относится данное задание, указываются в качестве параметров. При такой организации структуры очень легко без XML-обработчика отследить соответствующие параметры. Приведем другой пример использования XML в качестве разметки структуры тестового задания:

```

<?xml version=«1.0» encoding=«windows-1251»?>
<HEAD>
  <TITLE>Пример организации базы тестовых заданий средствами
XML
  </TITLE>
</HEAD>
<BODY>
  <SUBJECT>МАТЕМАТИКА
  <THEME>ТАБЛИЦА СЛОЖЕНИЯ
  <TASK>
    <type/>radiobtn
    <level/>low
    <T>Сколько будет 2+2?</T>
    <A>2</A>
    <A>4
    <result/>true
    <mark/>100%
  </A>
  </TASK>
  </THEME>
</SUBJECT>
  <SUBJECT>ЭТИКА СЕМЕЙНОЙ ЖИЗНИ
  <THEME>СЕМЕЙНОЕ ПОЛОЖЕНИЕ
  <TASK>
    <type/>checkboxbtn
    <level/>medium
    <T>Так кто в семье главный?</T>

```

```
<A>Я</A>
<A>Папа
  <result/>true
  <mark/>90%
</A>
<A>Мама
  <result/>false
  <mark/>10%
</A>
<A>Все</A>
</TASK>
</THEME>
</SUBJECT>
</BODY>
```

Данный пример значительно ускорит процесс поиска необходимых данных, кроме того, объединение вопросов в темы и предметы является более логичным и уменьшает избыточность информации, хранящейся в базе заданий. Но вид такого документа в обычном текстовом редакторе менее читабельный.

Таким образом, для организации хранения базы заданий в компьютерных системах тестирования средствами XML достаточно ввести необходимые теги и с их помощью описать структуру тестовых заданий, входящих в базу.

Литература:

1. Расширяемый язык разметки (XML) 1.0 (вторая редакция). Рекомендация W3C от 6 октября 2000 года:
<http://www.citforum.ru/internet/xmlspec/xmlspec.shtml>
2. Дакота М., Саганич А. XML и Java 2. – С-Пб.: Питер, 2001. – 377 с.
3. Чего мы ждем от XML Шелли Пауэрс (Мир ПК 3/1998 г.) –
<http://www.osp.ru/pcworld/1998/03/180.htm>
4. Использование XML в среде Delphi Александр Календарев –
<http://www.codenet.ru/progr/alg/Использование XML в среде Delphi.htm>

МЕТОДИЧНІ ОСОБЛИВОСТІ ВИКОРИСТАННЯ СИСТЕМИ “CONTROL” НА ЗАНЯТТЯХ З ІНФОРМАТИКИ У ВИЩІЙ ШКОЛІ

В.Ю. Баранов^а, Л.Л. Леоненко^б

м. Одеса, Одеська національна академія зв'язку імені О.С. Попова

^а bwr@ukr.net

^б leonid.leonenko@gmail.com

Комп'ютерна система тестування знань «Control» дозволяє студентам вводити відповіді на тестові запитання у “вільній” формі і здатна адекватно оцінювати відповіді з несуттєвими помилками, скороченнями та іншими видозмінами (див. [1; 2]).

Найважливішим завданням викладача щодо підготовки студентів до взаємодії з «Control» є роз'яснення того, що їх правильні відповіді на тестові питання можуть мати різну форму, і, отже, вони можуть не задумуватись над цією формою, основну увагу приділяючи змісту відповіді. З іншого боку, викладач має окреслити для студентів ті типи відповідей, де помилки недопустимі (як от у операторах мов програмування). Видається доцільним проведення практичних занять перед перевіркою знань з кожної нової навчальної теми. У новій темі нові типи задач потребують обговорення «з погляду» інформатики; і водночас до них можуть застосовуватись різні правила ідентифікації [2] системи «Control», – отже, може знадобитись ознайомлення студентів з особливостями подання відповідей на ці задачі. Наведемо тут деякі типи задач, які можна обговорювати зі студентами під час підготовки до тестування з тем «Система програмування DELPHI», «Розгалужені програми» та «Цикли».

Задача 1. У наступному переліку пропущено один з етапів роботи над DELPHI-проектом. Який етап ПРОПУЩЕНО? Опишіть його.

- вхід у систему DELPHI;
- побудова форми та її елементів;
- написання Pascal-коду, що визначає поведінку елементів форми;
- трансляція проекту та виправлення помилок;
- виконання проекту та отримання результатів;
- вихід з DELPHI.

Передбачається відповідь «Пропущено етап збереження проекту». Це той випадок, коли можливими є дуже багато різноманітних варіантів правильних відповідей. Студентам слід пояснити, що будь-яка їх відповідь, що є правильною за змістом, буде адекватно оцінена системою «Control». Можна, наприклад, розглянути такі варіанти правильних відповідей:

Слід зберегти проект.

Треба виконати запис файлів проекту та модуля на диск.

Пропустили етап Save Project

Save Unit, Save Project

і т.ін. З розгляду цих варіантів студенти повинні зрозуміти, що не всі слова у відповіді є однаково важливими – необхідно, щоб у ній було згадано 1) операцію збереження і 2) що саме зберігається. Якщо не буде присутня частина 1, відповідь взагалі неправильна; якщо ж пропущена частина 2, відповідь неповна (і оцінкою за неї може бути «3»). З іншого боку, студент бачить, що форма подання частин 1 та 2 є довільною.

Задача 2. *Напишіть ЛОГІЧНИЙ ВИРАЗ мови Pascal, який перевірятиме, що число X належить напівсегменту [A, B).*

Розбираючи цю задачу, слід пояснити студентам, що правильними відповідями на неї можуть бути різні за формою, але еквівалентні логічні вирази, як от $(x \geq A) \text{ and } (x < B)$, $(B > x) \text{ and } (A \leq x)$, або $\text{not}((x < A) \text{ or } (x >= B))$.

Задача 3. *Як можна охарактеризувати значення виразу not A or A мови Pascal?*

Тут слід пояснити студентам, що правильними відповідями будуть як слово true (константа мови Pascal), так і тексти натуральної мови «істина», «завжди є істинним»; а також відповіді на зразок «не може бути false», тощо.

Задача 4. *Відомо, що наступні два цикли є ЕКВІВАЛЕНТНИМИ (тобто результати їх роботи тотожні):*

- 1) while $x < (B + 0.1 * h)$ do begin ... <тіло циклу> ... end;
- 2) repeat ... <тіло циклу> ... until <умова>;

Яка умова повинна стояти після слова until у другому циклі? Запишіть цю умову.

Тут правильною відповіддю буде довільний логічний вираз, еквівалентний до $x \geq (B + 0.1 * h)$, наприклад $(x - 0.1 * h) \geq B$, тощо.

Систему «Control» можна вживати для перевірок готовності студентів до поточних лабораторних та практичних робіт з інформатики. Це є особливо важливим в умовах кредитно-модульної системи, яка, зокрема, передбачає перевірку якості підготовки студента до кожного практичного або лабораторного заняття. В умовах, коли у навчальній групі кожен студент забезпечений окремим комп'ютером, така перевірка може полягати у розв'язку 4-5 простих задач. Якщо ж комп'ютерів обмаль, можна обмежитись 2-3 задачами (складність яких, порівняно з попереднім випадком, можна зробити трохи вищою). В будь-якому випадку час, необхідний для тестування, не повинен перевищувати 7-10 хвилин (і тому параметр часу такої теми у банку даних «Control» слід встановити рівним 5-6 хвилинам, враховуючи «непродуктивні» витрати).

Будемо надалі називати таку «скорочену» за обсягом задач та часом перевірку знань «тестом допуску». Варіанти задач, включених до тесту допуску, повинні перевіряти розуміння студентами основних понять навчальної теми. Слід заборонити користування підказками-Help'ами (встановити параметр файлу задач тесту допуску, що відключить Help'и). Відключеним має бути також режим тренування з тесту допуску.

Студента, що не зміг пройти тесту допуску, тобто набрати мінімально необхідну суму балів, буде оцінено як невідготовленого до заняття, і він до нього не допускається. Таким студентам можна дати завдання прочитати відповідний розділ навчально-методичного посібника, після чого знов пройти тестування – але вже не через тест допуску, а з використанням повноцінного тесту до навчальної теми (називатимемо його далі «основним тестом»). Основний тест містить 10-12 (або й більше) задач, для нього можуть бути відкритими Help'и і дозволений режим тренування. Коли студент, не допущений до заняття, зможе отримувати «сталі» позитивні оцінки з основного тесту, можна дозволити йому розпочати лабораторну чи практичну роботу.

Звичайно, викладач може обрати і інші форми роботи з недопущеними студентами. Наприклад, можна перенести його роботу з «Control» на час додаткових занять (самостійної роботи). В цей час студент зможе тренуватись з проходження теми, а потім здати її і відпрацювати невиконану лабораторну роботу.

Розглянута форма поточного контролю сприяє більш відповідальному відношенню студентів до занять, активізації та закріпленню знань.

Наведемо тут приклади задач, які можуть бути включені до тесту допуску з навчальної теми «Розгалужені програми».

***Задача 5.** Запишіть, розділяючи комою, ВСІ КЛЮЧОВІ СЛОВА мови Pascal, які є необхідними (тобто без яких не можна обійтись) для запису РОЗГОРНУТОЇ форми оператора IF.*

***Задача 6.** Заповніть таблицю значень операції OR.*

Якщо «тест допуску» складається з 4-5 задач, подібних до щойно наведених, викладач може налагодити параметри оцінки за тест, наприклад, так, щоб вона була позитивною у випадку розв'язку не менше трьох задач. Він може на свій розсуд налагоджувати також параметри окремих задач (наприклад: чи поставить система хоч би задовільну оцінку на задачу 5, якщо у відповіді пропущено else, тощо).

Систему «Control» можна вживати для проведення тематичного контролю знань у кінці вивчення чергової навчальної теми. Для цього використовуються «основні тести» до теми, згадані вище. Час виконання такого тесту встановлюється рівним 15-20 хвилинам; сам тест містить близько десятка задач з усіх розділів теми. Щодо підказок-Help'ів, то викладач може, на свій розсуд, відключити або залишити їх (з урахуванням того, що звертання до Help'а знижує підсумкову оцінку за тест).

У залежності від вибраного методу проведення підсумкового заняття з навчальної теми викладач визначає «місце» системи «Control» у структурі цього заняття. Наприклад, якщо на занятті передбачається виконати контрольну роботу з побудовою і налагодженням Delphi-проєкту на комп'ютері, то основний тест «Control» може передувати побудові проєкту, або ж виконуватись у кінці заняття. З методичних міркувань, а також досвіду використання «Control», можна рекомендувати обрати саме другий варіант. Оскільки

ки метою викладання навчальної теми є навчити студентів вживати вивчений матеріал для розв'язування практичних задач на комп'ютері, уміння складати працездатні програми є головним предметом перевірки знань студентів. Отже, недоцільно відволікати їх від основного завдання на початку заняття. У процесі побудови та налагодження проекту студент може потребувати допомоги викладача, що, звичайно, впливає на його підсумкову оцінку. Якщо проект все ж було доведено до кінця, проходження студентом тесту «Control» допоможе викладачу скоригувати цю оцінку (у бік підвищення, якщо допомога була незначною а тест пройдено добре; і навпаки, якщо тест покаже, що потреба студента у втручанні викладача була невідповідною). Якщо ж студент так і не справився з Delphi-проектом, навіть з підказками викладача, проходження тесту «Control» видається вже недоцільним.

Наведені вище задачі 2 і 3 є прикладами варіантів задач основного тесту з теми «Розгалужені програми». Наведемо ще один приклад:

Задача 7. *Напишіть ОДИН розгорнутий умовний оператор, дія якого була б еквівалентною до дій двох наступних скорочених операторів:*

```
if x<=Pi/2 then y:=sin(x);  
if x>Pi/2 then y:=1-cos(x);
```

А от приклади задач з теми «Масиви»:

Задача 8. *Запишіть оголошення одновимірного масиву з іменем MAS, що складається з 10 дійсних чисел. Нумерація елементів масиву починається з нуля.*

Задача 9. *Яку задачу розв'язує наведений фрагмент програми ?*

```
...  
var A: array[1..10] of real;  
    D: real; k: integer;  
begin  
...  
D:=1; for k:=1 to 10 do if A[k]<>0 then D:=D*A[k];  
...  
...
```

Якщо файл задач «основного тесту» складається з 10-12 завдань, подібних до вищенаведених, можна рекомендувати налагодити його так, щоб для отримання оцінки «задовільно» студенту необхідно було розв'язати правильно не менше шести-семи задач. Можна задати для цих однакові або різні рівні складності. Якщо якісь задачі допускають одну з двох можливих відповідей, то число спроб на їх вирішення має дорівнювати 1. Якщо ж задача потребує обчислень, можна надати студенту більше спроб введення відповіді на неї, враховуючи, що студент міг обрати правильну схему розв'язку, але помилитись в обчисленнях.

На заліку, іспиті, або (в умовах кредитно-модульної системи) під час перевірки якості засвоєння матеріалу навчального модуля методика використання системи «Control» може бути подібною до методики, що вживалась під час тематичного контролю знань. Водночас тут можна вживати так звані

«залікові тести», що відрізняються від «основних» тестів за темами курсу наявністю задач, що відносяться до різних тем. Якщо дозволяє час, до «залікового тесту» можна включати більше задач (до 20) з відповідним подовженням часу тестування до 25-30 хвилин. Для отримання оцінки «задовільно» студенту необхідно розв'язати правильно не менше 10-12 задач залікового тесту. Як правило, ці задачі різні за своєю відносною складністю, що враховується системою «Control». Для деяких задач слід надати більше однієї спроби їх вирішення. Викладач повинен вирішити, якого типу підказки Help'и є допустимими в умовах заліку.

Система «Control» може бути корисною і в умовах іспиту з інформатики. Якщо, наприклад, у екзаменаційному білеті зазначено тему «Цикли», викладач може запропонувати студенту пройти «основний» тест з цієї теми перед тим, як студент розпочне свою усну відповідь. У випадку хорошої або відмінної оцінки можна запропонувати студенту для усної відповіді більш складні питання. Навпаки, якщо оцінка задовільна, можна спочатку перевірити знання студента з основних понять теми, і т.ін. Якщо у студента були труднощі при складанні та налагодженні проекту на комп'ютері, тест «Control» може стати додатковим завданням, що дасть змогу скоректувати його оцінку в той чи інший бік. При цьому у кожному конкретному випадку викладач зможе сам вибрати той варіант тесту, який допоможе оцінити знання конкретного студента («основний» тест з деякої теми чи «заліковий» тест).

Під час підготовки студента до контрольних робіт, заліку чи іспиту він може користуватись системою «Control» (у режимах «Тренування» та «Іспит») як засобом самоконтролю, тренажером для перевірки, закріплення та поглиблення отриманих знань.

Література:

1. Баранов В.Ю. Комп'ютерне тестування з інформатики: підсумки педагогічного експерименту в Одеській національній академії зв'язку. // Теорія та методика навчання математики, фізики, інформатики: Зб. наук. праць. Вип. 4: В 3-х т. — Кривий Ріг: НМетАУ, 2004. — Т. 3. — С. 6–12.
2. Леоненко Л.Л. Оцінювання відповідей “відкритого” типу за умов комп'ютерного тестування знань. // Теорія та методика навчання математики, фізики, інформатики: Зб. наук. праць. Вип. 4: В 3-х т. — Кривий Ріг: НМетАУ, 2004. — Т. 3. — С. 138–143.

ТЕСТИРОВАНИЕ ЗНАНИЙ СТУДЕНТОВ ПРИ ПОМОЩИ ПРОГРАММНОГО КОМПЛЕКСА «MATEST-2.0»

А.Ю. Мельников

г. Краматорск, Донбасская государственная машиностроительная академия
al_mel@mail.ru

Тестовый контроль знаний является неотъемлемой частью учебного процесса. В настоящее время разработано множество как универсальных, так и специализированных программ для проведения тестирования, каждая из которых имеет свои преимущества и недостатки. На ряде кафедр Донбасской государственной машиностроительной академии используется специальный программный комплекс MaTest, последняя версия (2.0) которого позволяет:

- создавать новые тесты в любом редакторе (используется специальный язык обработки данных) и добавлять их в библиотеку тестов;
- группировать вопросы по категориям (от 1 до 5);
- сохранять тесты в библиотеке в максимально сжатом виде;
- вести несколько библиотек тестов, определяя их использование переименованием файлов `my_test.trv` (дерево тестов) и `my_test.lib` (собственно библиотека);
- структурировать библиотеку тестов, объединяя тесты по предметам или модулям;
- определять число задаваемых вопросов (по умолчанию – 5) и общее время для ответов (по умолчанию – 100 секунд);
- случайным образом изменять размещение вариантов ответов;
- выбирать ответ как нажатием соответствующих клавиш («1», «2» и т.д.), так и щелчком мыши;
- отображать ход времени и число пройденных вопросов;
- выдавать результат тестирования как в виде числа отвеченных вопросов, так и в виде заранее заготовленного текста для определенного диапазона (например, результатом 18–20 правильных ответов из 20 вопросов может быть фраза «Ваша оценка – А»).

Пакет пока еще не позволяет создавать тесты в режиме конструктора, вставлять в тест графические материалы, учитывать одновременный выбор нескольких вариантов, проводить авторизацию пользователей, сохранять результаты в базе данных и работать в клиент-серверном режиме.

Типовой процесс тестирования выглядит следующим образом: из библиотеки выбирается тест; тестируемому поочередно задается определенное число вопросов, на каждый из которых ему предлагается к выбору от 2 до 10 вариантов ответа; нажатие соответствующей кнопки (или щелчок мышью) приводит к зачислению определенного числа баллов (в случае правильного ответа) и переходу к следующему вопросу. Параметры рабочей

среды (в том числе – процесса тестирования) при необходимости настраиваются и сохраняются на диске в файле config.tst.

Главное окно тестирующей системы приведено на рис. 1. Как видно, оно максимально функционально и не содержит ничего лишнего – только главное меню, область тестирования (вопросов и результатов), область выбора варианта ответа и полоски отображения хода времени и числа пройденных вопросов.

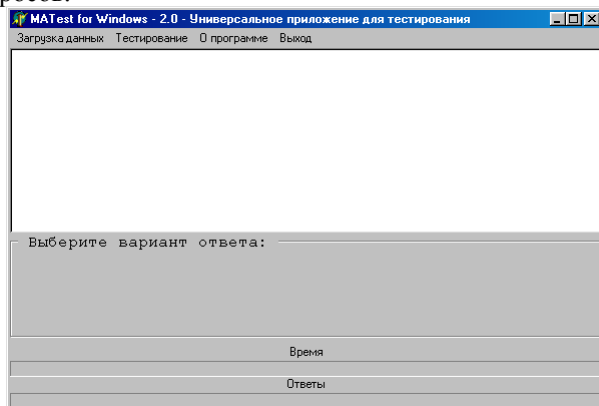


Рис. 1. Главное окно системы

Главное меню состоит из двух основных пунктов и двух вспомогательных («О программе» и «Выход»). «Загрузка данных» (или комбинация клавиш Ctrl+L) позволяет начать работу с библиотекой тестов, которая изображается в виде дерева (рис. 2). Для загрузки готового теста необходимо выбрать и нажать «Enter» или дважды щелкнуть мышью, после этого название теста будет отображаться в заголовке окна приложения. Следует помнить, что названия первого уровня – это предметы, а не тесты, и попытка их загрузки приведет к выдаче соответствующего предупреждения.

Если же необходимо осуществить какую-то деятельность с самой библиотекой (отличную от простой загрузки), то в этом вам поможет контекстное меню, вызываемое правой кнопкой мыши:

- «Загрузить список из файла» позволит выбрать библиотеку, отличную от «my_test»;
- «Добавить новый предмет» и «Добавить новый тест» позволят добавить в соответствующие ветви дерева новые пункты; следует помнить, что добавление нового теста означает фактически только ввод нового названия, но никак не самого теста;
- «Удалить предмет или тест» удаляет пункт из дерева;
- «Сохранить список на диске» позволит сохранить библиотеку с именем, отличным от «my_test»;
- «Создать список из *.bib» конвертирует старую библиотеку (форматы

- от dos-версий до MatTestW-1.0), не имеющую аналога в виде дерева; «Добавить новый тест» считывает из указываемого файла тест на языке создания тестов и помещает его в выделенное название (в случае отсутствия ошибок).

Структурирование библиотеки осуществляется простым «перетягиванием» выделенных пунктов (механизм «Drag&Drop»). Выход из режима библиотеки без загрузки тестов – пункт «Убрать библиотеку тестов» или клавиша «ESC».

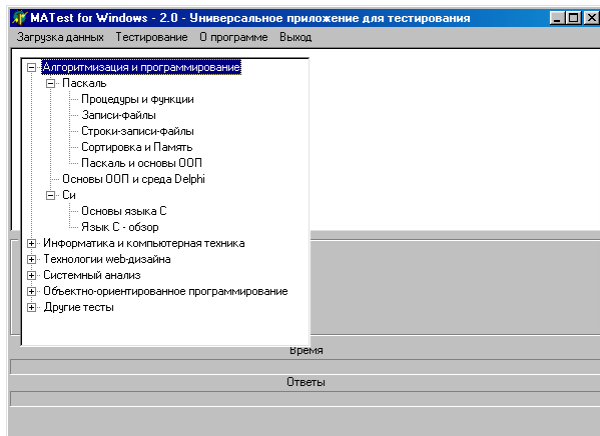


Рис. 2. Работа с библиотекой

Пункт «Тестирование» содержит разделы «Начать» (или комбинация клавиш Ctrl+T) и параметры среды и процесса тестирования (рис. 3). Значения большинства параметров понятны без комментариев, мелкие нюансы становятся понятны в процессе работы.

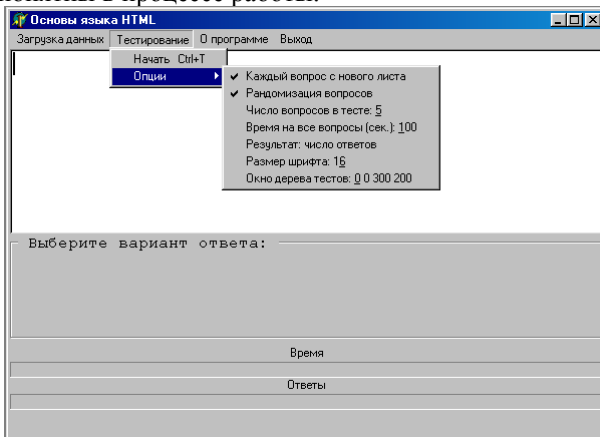


Рис. 3. Меню параметров

После выбора «Начать» начинается сам процесс тестирования, в ходе которого необходимо отвечать на предлагаемые вопросы (рис. 4) путем выбора соответствующего ответа щелчком мыши или нажатием на клавиатуре нужных («1», «2» и т.п. или первых букв ответа). Для прерывания тестирования можно нажать «ESC», после этого набранные ответы аннулируются.

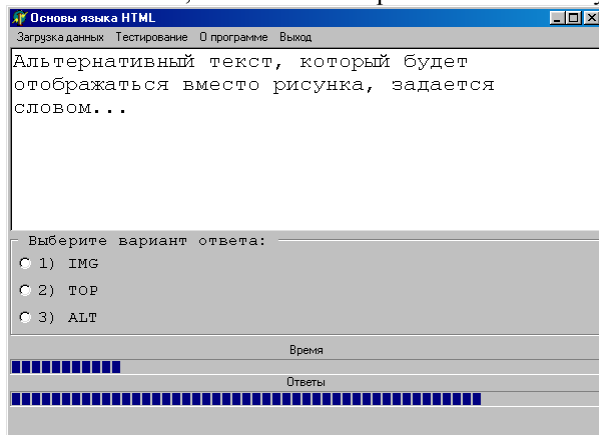


Рис. 4. Тестирование

После ответов на все вопросы или в случае окончания заданного времени тестирования результаты будут отображены в том же окне, где выво-дился вопрос (рис. 5). Если результаты представлены в виде не просто перечня правильных ответов, а как определенный текст, то его можно выделить и скопировать в любой текстовый редактор с целью последующего сохранения или вывода на печать.

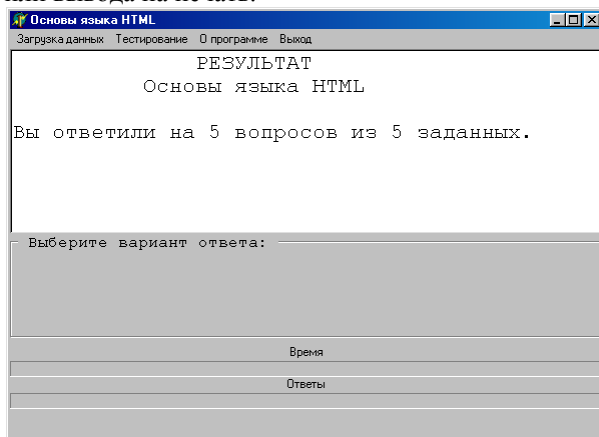


Рис. 5. Результаты

Для ввода тестов в пакет MaTest был разработан специальный язык [1] обработки данных (в дальнейшем – ЯОД), который предъявляет к вводимым тестам следующие требования:

а) каждый вопрос предполагает заранее определенные варианты ответа (возможное количество: от одного до десяти);

б) каждый вариант ответа предполагает начисление определенного числа очков, причем возможно не сплошное суммирование, а по частям;

в) результаты тестирования могут определяться по каждому диапазону набранных баллов.

Подробное описание данного языка, как и схема работы транслятора, выходит за границы данной статьи. Примеры записи тестов ничем не отличаются от тестов для прежних версий пакета, и с ними можно ознакомиться в [2–3].

Литература:

1. Рейурд-Смит В.Дж. Теория формальных языков. Вводный курс: Пер. с англ. – М.: Радио и связь, 1988. – 128 с.
2. Мельников А.Ю. Тестирующий программный комплекс «Matest» // Теорія та методика навчання математики, фізики, інформатики: Збірник наукових праць. Випуск 5: В 3-х томах. – Кривий Ріг: Видавничий відділ НМетАУ, 2005. – Т.3: Теорія та методика навчання інформатики. – С. 192-195.
3. Мельников А.Ю. Тестирование при помощи программного комплекса «Matest» // Нові інформаційні технології в навчальному процесі: П'ята міжнародна науково-практична конференція (29 червня – 2 липня 2005 р.). – Одеса: Астропринт, 2005. – С. 10-13.

ОПЫТ РАЗРАБОТКИ И ВНЕДРЕНИЯ ПРОГРАММНОГО КОМПЛЕКСА LAZYTEST ДЛЯ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ УРОВНЯ ЗНАНИЙ СТУДЕНТОВ

М.И. Румянцев

г. Павлоград, Западнодонбасский приватный институт экономики
и управления
renixa-1959@mail.ru

Цели и задачи разработки. Общемировые тенденции применения инновационных технологий в высшей школе, практика ведущих вузов СНГ и Украины свидетельствуют о том, что использование автоматизированных тестирующих систем на базе персональных компьютеров является одной из перспективных форм контроля знаний студентов. Не останавливаясь подробно на общеизвестных преимуществах подобного подхода, отметим лишь тот факт, что автоматизированное тестирование – мощное средство для повышения качества образовательных услуг и усиления конкурентных позиций конкретного вуза на этом рынке.

Однако далеко не все учебные заведения Украины имеют возможность самостоятельной разработки либо приобретения развитых систем автоматизированного тестирования вследствие недостатка средств и квалифицированных специалистов данного профиля. В связи с этим возникает необходимость в разработке более простых по возможностям программных средств, обладающих приемлемой функциональностью и служащих отправной точкой для дальнейших исследований в данной области.

Исходя из вышесказанного, на кафедре прикладной математики и информатики ЗПИЭУ в январе-феврале 2005 г. была разработана версия 1.0 программного комплекса LazyTest для автоматизированного контроля уровня знаний студентов. В марте того же года комплекс прошел опытное внедрение, по результатам которого был осуществлен ряд доработок (на момент написания данной статьи штатно эксплуатируется версия 1.4). При реализации проекта были учтены рекомендации по проведению психометрического тестирования в вузах США и развитых странах Запада, а также методики проведения сертификационных испытаний корпорации Майкрософт (MCSE).

По замыслу разработчика, созданный комплекс призван способствовать достижению следующих основных целей: *тактических* – повышению объективности оценивания знаний студентов; более полному охвату контролируемого учебного материала; сокращению непроизводительных затрат времени преподавательского состава; упрочению «фундамента» кредитно-рейтинговой системы; *стратегических* – повышению конкурентоспособности вуза на региональном рынке образовательных услуг за счет внедрения передовых технологий и методик оценивания знаний обучаемых; облегче-

нию перехода вуза на международные стандарты работы учреждений высшей школы.

В ходе реализации проекта были поставлены и успешно решены следующие этапные задачи: создание программной оболочки для проведения непосредственно тестирования; создание программы для анализа результатов тестирования и оценки качества материала теста; обеспечение возможности подготовки материала теста привычными для пользователей средствами; обеспечение защиты материалов теста от несанкционированного раскрытия студентами; создание технологии проведения тестирования как при наличии локальной сети, так и без таковой; экспериментальное внедрение комплекса; переход к промышленной эксплуатации.

Краткое описание возможностей LazyTest. Программный комплекс LazyTest состоит из следующих компонентов, реализованных в виде отдельных программ: программная оболочка *LazyTest* (рис. 1); модуль анализа результатов тестирования *LazyStat*; модуль шифрования исходных материалов теста *LazyCrypt*.

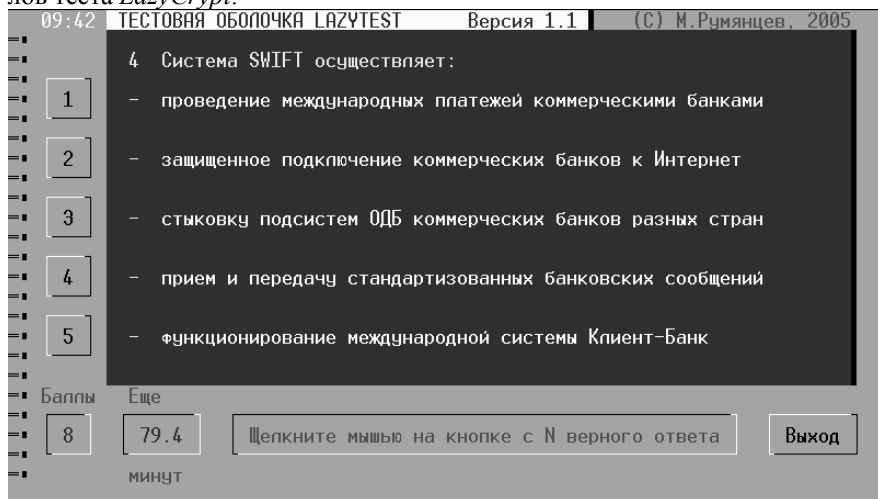


Рис. 1. Общий вид тестовой оболочки LazyTest (версия 1.1)

Тексты исходных тестовых материалов (вопросов и ответов с баллами) хранятся в кодировке 866 и могут быть подготовлены, к примеру, с помощью свободно распространяемого текстового редактора Vred2. С его помощью также удобно распечатывать протоколы работы тестовой оболочки LazyTest и модуля анализа LazyStat.

Оболочка *LazyTest* является оболочкой закрытого типа и поддерживает тестовые наборы размерностью до 100 вопросов (с количеством вариантов ответа на каждый вопрос – до 5). Максимальная длина вопроса – 60 знаков, длина каждого из вариантов ответа – до 120 знаков. Возможна работа со

сменными тестовыми наборами (по разным дисциплинам, темам, уровням сложности и т.п.). При тестировании осуществляется контроль времени тестирования (исходя из норматива 80 минут на 50 вопросов, т.е. для ответа на 1 вопрос отводится до 96 секунд). Вопросы задаются в случайном порядке – что уменьшает риск «коллективного» ответа на вопросы студентами, сидящими за соседними компьютерами. Правильные ответы оцениваются баллами 1, 2, 3, 4 и 5; за неправильные ответы начисляется 0 баллов; неправильные ответы, свидетельствующие о грубейших ошибках и непонимании учебного материала, оцениваются штрафом в –5 баллов. Таким образом, рациональное сочетание ответов различной степени правильности для каждого вопроса позволяет дифференцированно оценить глубину знаний студентов и избежать «эффекта лотереи» при тестировании.

В начале процедуры тестирования студент вводит с клавиатуры свою фамилию. Дальнейшие действия по выбору нужного варианта ответа на очередной вопрос выполняются с помощью мыши. По ходу тестирования на экран периодически выдается набранное количество баллов (нарастающим итогом) и оставшееся до конца тестирования время (в минутах). По завершении ответа на все вопросы (либо по истечении контрольного времени) оболочка оценивает и сообщает уровень знаний, продемонстрированный студентом в процентах от максимально возможного количества баллов за данный тест (рис. 2).

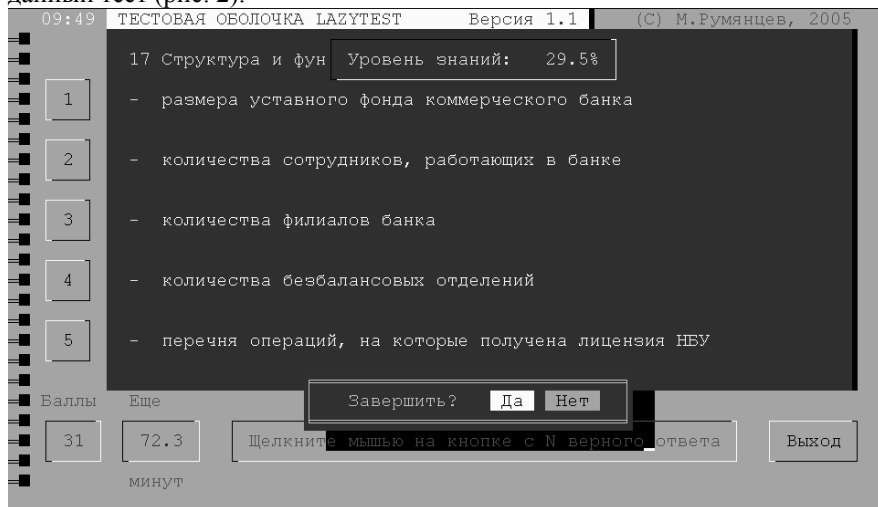


Рис. 2. Завершение работы с тестовой оболочкой LazyTest

Модуль LazyTest сохраняет результаты тестирования каждого студента в файле-протоколе lazytest.pnt, что позволяет преподавателю впоследствии объединить подобные файлы всей академической группы для дальнейшего статистического анализа. В протоколе содержится следующая информация

о ходе тестирования: дата тестирования; время окончания тестирования студентом; фамилия студента; время, затраченное студентом на тестирование; количество набранных баллов из максимально возможных; уровень знаний в процентах от максимально возможного для данного тестового набора; количество правильных ответов на вопросы; «профиль» ответов на вопросы (1 – ответил, 0 – не ответил для версий 1.0–1.3; в версии 1.4 фиксируются конкретные баллы, полученные за каждый ответ).

Модуль *LazyStat* позволяет сформировать и сохранить в файле *lazystat.dat* результаты статистического анализа итогов тестирования (на основе сводного протокола группы). Принятый в комплексе формат протокольного файла дополнительно позволяет при необходимости импортировать информацию в среду электронных таблиц для анализа в том виде, в котором удобно пользователю. Базовый протокол работы модуля *анализа* содержит следующую информацию, сгруппированную по разделам: наименование группы; количество вопросов в тесте и максимально возможное количество баллов; количество протестированных студентов; время прохождения теста (лучшее, среднее, худшее); количество верных ответов (лучшее, среднее, худшее); количество набранных баллов (лучшее, среднее, худшее); уровень знаний в процентах от максимума (лучший, средний, худший); гистограмму распределения студентов в зависимости от продемонстрированного уровня знаний; статистические характеристики уровня знаний (матожидание, дисперсия, доверительный интервал и т.д.); анализ сложности вопросов теста (количество ответивших менее чем на 50% вопросов; количество ответивших на более 50% вопросов; количество показавших менее 50% знаний; количество показавших более 50% знаний; уровень сложности – по каждому вопросу теста).

Модуль *LazyCrypt* предназначен для шифрования подготовленных исходных материалов теста (вопросов, вариантов ответов с баллами). Как и *LazyStat*, он выполняется в пакетном (недиалоговом) режиме и не требует никакой особой квалификации для применения.

Все 3 модуля комплекса разработаны с использованием компилятора PowerBASIC 2.10 и полностью работоспособны в операционных средах MS Windows 95/98/Me/XP, а также MS-DOS (DR-DOS). Функционирование модуля *LazyTest* текущей версии (1.4) под управлением MS Windows 2000 не гарантируется. Применение ориентированного на DOS компилятора позволило создать *exe*-файлы, входящие в комплекс, «микроскопического» по нынешним меркам размера (так, длина *lazystest.exe* версии 1.4 составляет всего 80284 байта).

Анализ результатов внедрения. Базой для опытного внедрения комплекса *LazyTest*, осуществленного в марте 2005 г., послужили 3 академические группы 4-го курса, сдававшие согласно графика учебных занятий зачет по дисциплине «Информационные системы и технологии предприятий» (студенты-экономисты) и экзамен по дисциплине «Информационные систе-

мы и технологии финансово-кредитных учреждений» (студенты-финансисты). Выбор 4-го курса был обусловлен тем, что четверокурсники, будучи выпускниками бакалаврата, являются более психологически устойчивыми по отношению к новациям, нежели 1-3 курсы. Помимо этого, за плечами студентов 4-го курса не только 7 сданных сессий, но и определенный профессиональный и жизненный опыт – прежде всего это касается студентов заочной формы обучения. Более того, привлечение к тестированию как студентов дневной, так и заочной формы обучения позволило всесторонне оценить качество подготовленных тестов (как в плане теоретических знаний, так и практических навыков) и объективно сопоставить уровень освоения учебных дисциплин обеими категориями студентов.

Таким образом, в категорию тестируемых вошли (из числа допущенных к зачету/экзамену) 26 студентов стационара, обучающихся по специальностям «Финансы» и «Экономика предприятия», а также 26 студентов-заочников специальности «Финансы». Соответственно, были подготовлены 2 набора тестов: один – для сдачи зачета по ИСТП, другой – для сдачи экзамена по ИСТФКУ. Оба тестовых набора состояли из 50 вопросов и охватывали практически весь учебный материал указанных дисциплин. Большинство вопросов требовали от студентов не бездумного зазубривания определений и формулировок, но предполагали *понимание* ключевых положений дисциплины, а также достаточный общеобразовательный уровень развития студента (в т.ч. знание «пограничных» учебных предметов). Со всеми группами предварительно на практических занятиях и консультациях был проведен инструктаж по предстоящему тестированию – во избежание возникновения стрессовых ситуаций в связи с фактором неожиданности.

Опытное внедрение комплекса LazyTest показало следующее.

С организационной и технологической сторон процесс тестирования не вызвал особых трудностей у студентов – более того, был проявлен определенный интерес к этому мероприятию. Дополнительных пояснений по интерфейсу программной оболочки не понадобилось. Времени на тестирование оказалось более чем достаточно – в «цейтнот» попали не более 3-4 человек из 52 протестированных, чему причиной были недостаточные и непрочные знания. Практически все тестируемые даже после ряда неверных ответов в первые минуты тестирования сумели взять себя в руки и сосредоточиться (исключение составил студент Б., бурно проявлявший эмоции по поводу своих неудач первые 10-15 минут тестирования и потерявший как минимум 15 баллов вследствие этого).

Статистическое распределение продемонстрированного студентами уровня знаний оказалось достаточно близким к нормальному, что было подтверждено как соответствующими гистограммами, так и рассчитанными статистическими характеристиками. Пофамильный анализ протоколов тестирования показал хорошую корреляцию с оценкой знаний студентов, полученной «традиционными» способами. Однако в силу большей объектив-

ности компьютерного тестирования выявилось смещение общей оценки уровня знаний студентов приблизительно на 1 балл в меньшую сторону: вместо «отлично» – «хорошо», вместо «хорошо» – «удовлетворительно» и т.д. Тем не менее по-настоящему успевающие студенты стационара все равно набрали максимум баллов (относительно своих коллег) и подтвердили обоснованность своих притязаний на высшую оценку. Выявленная таким образом объективная картина уровня знаний студентов по ИСТП и ИСТФКУ оказалась достаточно удручающей (хотя и не особо отличающейся от оценок по другим дисциплинам): средний уровень знаний студентов стационара по ИСТП – 39,2% (лучший результат в группе – 53,8%); средний уровень знаний студентов стационара по ИСТФКУ – 41,1% (лучший результат в группе – 65,7%); средний уровень знаний студентов-заочников по ИСТФКУ – 26,2% (лучший результат в группе – 46,7%).

Анализ сложности тестов, проведенный программой LazyStat, позволил выделить слишком легкие либо слишком трудные для данной категории студентов вопросы. Так, например, для студентов группы ЭД-21 «не по зубам» оказались вопросы 25 и 44 теста по ИСТП, зато на вопросы 17 и 42 ответили все студенты этой группы.

Сравнивая результаты, показанные студентами стационара и заочниками на экзамене по ИСТФКУ, необходимо отметить, что заочники в целом продемонстрировали лучшие знания относительно характерных особенностей финансово-кредитных учреждений, специфики автоматизации безбалансовых отделений банков, автоматизации обработки кассовых документов, трудностей при внедрении ИТ в учреждениях банков, выгод и преимуществ внедрения системы «Клиент-Банк» и т.п. Приблизительно равные стационару (по проценту ответивших) результаты заочники показали и в вопросах выбора критериев эффективности и оптимальности информационных систем, автоматизации операций кредитования, сущности и преимуществ платежей с применением пластиковых карт и т.п. Уступая в целом студентам стационара в объеме усвоенных знаний (прежде всего теоретических), поверхностно ознакомившись с учебной литературой и конспектом лекций, заочники все-таки показали себя с лучшей стороны в вопросах, требующих профессиональных навыков и знаний. Как и следовало ожидать, более-менее приличные результаты среди студентов группы Ф3-31 показали работники банков (2 – «Укрсиббанк», 1 – «Надра»), практики (экономисты и бухгалтеры предприятий и фирм) и т.д.; не работающие либо работающие не по специальности заочники продемонстрировали крайне низкий уровень знаний.

Как курьезное исключение из общей тенденции можно отметить тот факт, что студентка-заочница С. (очень неплохо подготовленная, с большим практическим опытом) показала при тестировании уровень знаний 0%. Наблюдение за ней во время тестирования (и ее дальнейшие объяснения) показали, что студентка слишком тщательно и придирчиво выбирала нужные

ответы на вопросы теста. Этому способствовал специфический профессиональный опыт, не позволявший ей непредвзято подойти к ответам на вопросы.

Успешное опытное внедрение комплекса LazyTest позволило в 1-м семестре 2005-2006 учебного года перейти к промышленной эксплуатации и охватить компьютеризованным тестированием не только студентов 4-го, но и 1-3 курсов. При этом на автоматизированные «рельсы» было переведено уже не только проведение зачетов и экзаменов, но и промежуточное тестирование по таким дисциплинам, как «Информатика и компьютерная техника» (1 и 2 курс), «Основы информационных систем» (3 курс), «Информационные системы и технологии предприятий» и «Информационные системы и технологии финансово-кредитных учреждений» (4 курс). Таким образом, тестированием с применением комплекса LazyTest было охвачено 70 первокурсников (в 3-х тестах), 73 студента 2 курса (в 3-х тестах), 67 студентов 4 курса (в 3-х тестах) и т.д. В частности, студенты 1-2 курсов продемонстрировали устойчивое смещение получаемых в результате оценок в лучшую сторону (по сравнению с традиционными экзаменами); студенты 4 курса, как и в прошлом учебном году, по-прежнему показали на тест-экзамене худшие оценки, чем в «классическом» варианте.

Дальнейшие направления развития системы. Исходя из вышеизложенного, можно считать первоначальный этап по разработке и внедрению комплекса для автоматизированного тестирования успешно завершенным – в рамках поставленных целей и задач.

С учетом результатов эксплуатации версий 1.0-1.4 комплекса просматриваются следующие направления развития LazyTest: *организационные* – упрощение технологии работы с комплексом с целью расширения сферы его применения на дисциплины социально-гуманитарного и экономического профиля; массированное внедрение тестирования для текущего и итогового контроля по всем дисциплинам компьютерного цикла; *технические* – обеспечение работоспособности комплекса в среде MS Windows 2000; совершенствование интерфейса оболочки с учетом требований эргономики; автоматизация формирования сводного протокола тестирования группы при работе в локальной сети; реализация более детального анализа степени освоения студентами учебного материала (как в разрезе контролируемых тем, так и отдельных вопросов).

К ВОПРОСУ О ФИНАНСОВОЙ ЭФФЕКТИВНОСТИ ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

М.С. Учитель

г. Днепропетровск, Национальная металлургическая академия Украины
alanor@list.ru

Не существует единого определения открытого и дистанционного обучения (далее – ДО). Скорее, существует много подходов к определению данного термина. Во многих определениях, однако, выделяются следующие характеристики [1]:

– разделение преподавателя и учащегося во времени или пространстве, или же и во времени, и в пространстве одновременно;

– признание учебным заведением: процесс обучения признается или сертифицируется каким-либо образовательным учреждением или организацией. Такой тип обучения отличается от самообучения – собственными силами, без официального признания со стороны учебного заведения;

– использование учебно-методических материалов, разработанных с применением различных медиа-средств, включая печатные материалы, радио и телевизионные передачи, видео- и аудиокассеты, компьютерных программ и телекоммуникаций. Учебные материалы курса обычно перед использованием проходят предварительное тестирование и апробацию;

– двусторонняя коммуникация, которая делает возможным взаимодействие учащихся и педагогов и отличается от пассивного восприятия транслируемой через сети вещания информации. Коммуникация может быть синхронной или асинхронной;

– возможность очных встреч для проведения семинаров, взаимодействия учащихся друг с другом, занятий в библиотеках, проведения лабораторных или практических сессий.

В последние годы можно было наблюдать возрастание интереса к ДО учебных заведений на постсоветском пространстве. Появление правовой базы, отражающей реалии, задачи и перспективы ДО [2; 3], обозначило переход к их практическому внедрению в Украине.

ДО – достаточно молодая технология, находящаяся в стадии развития, по-прежнему являющаяся предметом дискуссий и исследований даже в развитых странах, где оно имеет 20-30-летнюю историю. Привлекательность концепции ДО для педагогов и учащихся зачастую обусловлена инновационным, современным характером этой технологии, своеобразным «обаянием 21-го века», характерными приоритетами информационных технологий и глобализации. Эти же черты ДО отталкивают его противников. Организаторов образовательного процесса привлекает экономическая эффективность ДО. Широко распространено мнение, что ДО оказывается дешевле и экономически эффективнее, чем любая другая форма образования. Среди основ-

ных доводов в пользу этой оценки упоминаются такие факторы, как сравнительно более активное применение преподавателями мультимедийных технологий и средств (что, в свою очередь, существенно увеличивает численность учащихся), отсутствие необходимости в классах и других специально оборудованных учебных помещениях, а также относительно низкая стоимость образовательных процессов, обусловленная прежде всего структурой капитализации дистанционного обучения, в которой использование трудовых ресурсов педагогического состава в значительной мере заменено эксплуатацией технологических ресурсов, обеспечивающих распространение учебного материала.

Однако сопоставление средней стоимости обучения одного учащегося в разных учреждениях открытого и дистанционного образования и в столь же разных традиционных учебных заведениях при одинаковых юридических условиях функционирования тех и других [4], не позволяет сделать однозначный вывод о более высокой экономичности первых.

В таблице 1 представлен коэффициент экономической эффективности обучения студентов (исчисляемый путём деления средней стоимости обучения одного студента в дистанционном вузе на среднюю стоимость обучения одного студента в традиционном вузе) для ряда учебных заведений. Коэффициент, меньший 1, свидетельствует о том, что стоимость дистанционного обучения ниже стоимости традиционного.

Таблица 1.

Сравнение стоимости ДО и традиционного обучения в учебных заведениях

Учреждение ДО	Сравниваемое учебное заведение	Год	К-т
Institute de Radiodifusao Educativa da Bahia (IRDEB), Бразилия	Традиционная школа	1977	4,36
Telesecundaria, Мексика	Традиционная дневная средняя школа штата	1975	0,76
		1981	1,09
		1988	1,32
Centres APEC de Ensecanza a Distancia, Доминиканская республика	Дневная средняя школа второй ступени	1988	0,46
Universitas Terbuka, Индонезия	Традиционный курс педагогического образования	1988	0,60
Allama Iqbal Open University, Пакистан	Традиционные университеты	1988	0,22
University of the Air, Япония	Национальные университеты	1986	0,47
	Государственные университеты	1986	0,55
	Дневная программа частных университетов	1986	0,82

	Система заочного обучения с использованием рассылки материалов по почте	1986	7,64
Abbey National, Великобритания	Курс традиционного обучения Abbey National	1990	0,42
Delco Electronics, Великобритания	Курс традиционного обучения Delco Electronics	1990	1,08

Из данных таблицы 1 очевидно, что дистанционные методы обучения не всегда обеспечивают более эффективный с экономической точки зрения подход к образованию, а потому нельзя утверждать, что выбор в пользу этих методов является единственно верным в любых условиях. Целесообразно говорить о том, что дистанционное обучение можно рассматривать в качестве преимущественного способа решения тех или иных образовательных задач в тех случаях, когда главным критерием не является низкая стоимость образования. Пока же ситуация такова, что экономичность открытого и дистанционного образования не обязательно выступает одним из его преимуществ по отношению к традиционному, так что выбор той или иной формы обучения должен стать предметом тщательного анализа и определяться в каждом случае конкретной образовательной политикой.

Тем не менее, очевидно, что дистанционное образование может быть более рентабельным, чем традиционное образование. Основы финансового анализа, прогнозирования и оценки рентабельности ДО изложены в работах [5–8] и др.

В соответствии с принятым в настоящее время подходом, функция стоимости ДО имеет вид:

$$T = Sp + C_{\mu} + F,$$

где T – общая стоимость, S – число студентов, C – число курсов, или объем материалов, p – стоимость курса на студента, μ – стоимость раздела курсов или материалов, и F – фиксированные затраты.

Планируя систему ДО в надежде, что она будет приносить экономию, обусловленную ростом масштабов, необходимо обеспечить, чтобы:

1. Переменные затраты на студента были меньше, чем установлены в обычных системах, работающих на таком же образовательном уровне.

2. Число студентов S было бы достаточно большим, чтобы понизить величину среднего расхода средств на обучение одного студента (рассчитываемую по формуле $AC = (C_{\mu} + F) / (S + p)$) ниже уровня среднего расхода, установленного в обычных образовательных системах.

3. Процент отчисленных сохранился на разумно низком уровне.

4. Число курсов (или количество материалов) C было не настолько большим, чтобы увеличить стоимость $(C_{\mu} + F)$ до уровня, когда становится трудно обеспечить вероятное число студентов (S), дающее среднюю стоимость студента (AC) ниже уровня, «конкурентоспособного» со средней стоимостью на студента в обычных образовательных системах.

Эти условия имеют существенное значение для:

а) выбора медиа-средств. Теоретически, преподаватели дистанционного обучения имеют широкий выбор пригодных для них медиа. Практически же, этот выбор часто ограничивается: не столько абсолютными затратами на определенные медиа, сколько последствиями их применения, которые могут заметно отразиться на средних затратах на студента;

б) маркетинговых исследований, нацеленных на привлечение достаточного количества студентов к обучению по определенным программам вуза, с целью получения экономии, обеспеченной ростом численности студентов;

в) ресурсов, направляемых в сферу услуг для студентов. Так как эти затраты – переменная стоимость, степень инвестиций для поддержки студенческой сферы услуг должна соответствовать средним расходам на обучение одного студента (АС) и проценту отчисленных;

г) для определения ресурсов, направляемых в центральную инфраструктуру (постоянные затраты учреждения), которые должны быть сопоставимы с количеством студентов.

Таким образом, ДО может быть более дешевым, чем обычные методы, но это зависит от конкретных условий. Ясно, что обеспечение большого набора студентов имеет большое влияние на то, действительно ли средние затраты на студента снижены до уровня, на котором ДО становится экономически привлекательным. Также ясно, что дистанционные системы имеют высокую абсолютную стоимость, и что когда количество студентов ограничено, дешевле использовать обычные методы. В какой момент система ДО становится более дешевой в расчете на студента, чем обычная, – это вопрос, который предстоит решить в Украине с учетом специфики финансирования образования в нашей стране, практики действующих вузов, динамики общественного спроса на специалистов той или иной специальности.

Важным моментом является также высокая начальная стоимость дистанционных проектов. Поэтому даже если ДО может обеспечить экономию, обусловленную ростом масштаба, и уровень потенциального спроса высок, зачастую стоимость проекта настолько велика, что вуз не может обеспечить стартовые затраты.

Достижение более низких средних затрат на студента желательно с финансовой и экономической точки зрения, но это второстепенная цель. Тот факт, что некоторые студенты (например, взрослые, полностью занятые и живущие в отдаленных областях) могут получить образование только дистанционными средствами, сам по себе имеет определяющее значение при принятии решения о развитии программ дистанционного обучения, и сравнительная стоимость этого обучения может быть оценена как несущественный фактор при обсуждении таких программ.

Література:

1. Introduction to open and distance learning. The Commonwealth of Learning, 2000.
2. Концепція розвитку дистанційної освіти в Україні. (затверджено Постановою Міністерства освіти і науки України 20.12.2000 р.)
3. Положення про дистанційне навчання. Наказ Міністерства освіти і науки України №40 (затверджений 21.01.2004 р.)
4. Open and distant learning. Trends, Policy and Strategy Considerations. Division of Higher Education, UNESCO, 2002.
5. Rumble, Greville. 1988. «The Costs and Costing of Distance/Open Education.» In J. Jenkins, ed., Commonwealth Co-operation in Open Learning: Background Papers. London: Commonwealth Secretariat, pp. 255-58, 264-66.
6. Curran, Chris. 1989. «Resource Factors: Recurrent Costs.» In UNESCO and International Council for Distance Education, Developments in Distance Education in Asia: An Analysis of Five Case Studies, Paris/Oslo, pp. 23-26.
7. Hezel, Richard T. 1992. «Cost-effectiveness for Interactive Distance Education and Telecommunicated Learning.» In University of Wisconsin-Madison, From Vision to Reality: Providing Cost-effective, Quality Distance Education. Papers from the Eighth Annual Conference on Distance Teaching and Learning, Madison, Wisconsin, August 5-7, 1992, pp. 75-78.
8. Taylor, J., and V. White. 1994. «The Evaluation of the Cost-effectiveness of Multi-MediaMixed-Mode Teaching and Learning.» In SAIDE, Opportunities for Innovation in Higher Education. Proceedings of a Workshop and Conference of Vice-Chancellors and Rectors in South African Higher Education, November 10-12, 1993, pp. 109-17.

АВТОМАТИЗОВАНА НАВЧАЛЬНА СИСТЕМА ПО РОЗРОБЦІ WEB-СТОРОІНОК

В.В. Коломенська, Р.С. Пастухов
м. Донецьк, Донецький національний університет
kolom@dongu.donetsk.ua

В останні роки відбувається стрімке впровадження комп'ютерних інформаційних технологій у всі сфери життя. Перед навчальними закладами стоїть проблема підготовки досвідченого користувача, здатного оперативно вирішувати прагматичні задачі і самостійно засвоювати нові інформаційні технології.

Це знайшло відображення у структурі курсу інформатики – програма курсу швидко змінюється, але при цьому не вистачає часу для оперативної підготовки та видання відповідних навчальних посібників та методичних розробок. Вказане стосується як шкільного курсу “Основи інформатики і обчислювальної техніки”, так і курсів “Інформатика” та “Методика викладання інформатики” у вищій школі.

Аналогічні проблеми виникають і в інших навчальних курсах, що зазнають динамічних змін.

У зв'язку із цим в освіті велика увага приділяється застосуванню прогресивних методик навчання, що не потребують використання друкованих посібників. Зокрема це методики, що використовують обчислювальну техніку й можливості комп'ютерних мереж.

Комп'ютерні програми, призначені для передачі користувачам знань і розвитку у них практичних навичок з різних навчальних дисциплін, одержали назву “Автоматизовані навчальні системи” (АНС). В цих системах органічно поєднаний теоретичний матеріал, лекційні демонстрації, система практичних завдань, а також, що дуже важливо, можливість оперативної перевірки знань і умінь.

Викладання таких тем в курсі інформатики, як “Глобальна мережа Internet”, що розглядається в базовому курсі, ускладнюється ще й тим, що в деяких навчальних закладах взагалі немає прямого підключення до мережі Internet. Це, звичайно, робить неможливим вивчення практичного курсу.

Ситуація спрощується, якщо використовувати АНС. Для АНС по розробці WEB-сторінок, що представлена в даній роботі, не треба прямого підключення до мережі Internet, її програмний код вже має все необхідне: вбудований локальний сервер, браузер та необхідні графічні програми. Вказана АНС може бути використана при навчанні Internet-технологіям в курсі інформатики.

Розвиток навчальних систем у наш час йде у напрямку надання їм власливості адаптації до цілей та умов навчання.

Впродовж майже ста років психологи значну частину своїх наукових

зусиль витрачали на те, щоб зрозуміти процес навчання. При цьому досліджувалися, головним чином, фактори засвоєння, що впливають на швидкість отримання знань та їх втрати. Внаслідок цих зусиль був встановлений ряд надійних принципів, які можуть бути використані для побудови ефективних схем навчання [1]. Найголовніші з них перелічені нижче.

1. Навчання йде швидше, й матеріал засвоюється глибше, якщо учень виявляє активну зацікавленість до досліджуваного предмета.
2. Навчання є більш ефективним, якщо форми набування знань і навичок такі, що без зусиль можуть бути перенесені в умови “реального життя”, для чого вони й призначені.
3. Навчання йде швидше, якщо програма з предмету побудована за принципом послідовного ускладнення матеріалу. Заняття варто починати з найпростіших завдань, для виконання яких учні вже володіють необхідними навичками й знаннями. Поступово рівень складності матеріалу підвищується. Це триває доти, поки не буде досягнутий бажаний рівень досвідченості й уміння.
4. Знання учнем оцінки результатів власної роботи стимулює виконання ним чергового завдання, що надається за новим запропонованим матеріалом. Труднощі, які учневі необхідно переборювати, повинні виникати перед ним послідовно, а успішне їх подолання розвиває високий рівень активності.
5. Оскільки навчання саме по собі індивідуальне, процес навчання варто організувати так, щоб кожний учень міг проходити програму відповідно до своїх індивідуальних особливостей.

На основі перелічених принципів були розроблені вимоги до середовища АНС по розробці WEB-сторінок:

- реалізація безпосереднього діалогу між системою й користувачем за собою скриптів;
- нескладна внутрішня організація;
- послідовність навчального матеріалу та завдань практичної частини;
- можливість зміни навігаційного механізму навчальної системи, тобто інтерактивність її інтерфейсу.

Існує досить багато навчальних систем і засобів для їхньої розробки. Можна виділити три основні структури, на які діляться всі навчальні програми за принципами їх організації.

1. Системи на основі лінійного тексту. В цих системах матеріал подається у вигляді звичайного тексту, що може містити деякі малюнки. Ознайомлення користувача з текстом йде в заздалегідь певній послідовності, яку він не може змінити. У найкращому разі подібна система пропонує повернутися на крок назад або почати навчання із самого початку. Системи з подібною організацією даних звичайно не припускають яких-небудь тестових програм, а якщо такі є, то все, на що вони здатні, це повернути користувача до попередньої теми або виставити йому оцінку за прочитаний матеріал.

2. Мультимедійні навчальні системи. Такі системи дозволяють гармонійно об'єднати лекцію з демонстраціями, практикум у вигляді комп'ютерного імітатора, тестуючу систему й усі додаткові матеріали в єдиному інтерактивному комп'ютерному підручнику. Але при цьому мультимедійні системи навчання вимагають відповідної апаратної підтримки, займають великі обсяги пам'яті, що обмежує область їхнього застосування.

3. Системи на основі гіпертексту. Гіпертекст як підхід до керування інформацією відрізняється від інших підходів тим, що основний вид діяльності користувача при роботі з ним складається не стільки в пошуку потрібної інформації, скільки в ознайомленні з певним предметом за допомогою перегляду ряду інформаційних фрагментів, що зв'язані між собою за змістом. Ознайомлення здійснюється в певній послідовності, яка обумовлена цілями користувача.

Можливість варіювання послідовності ознайомлення зі змістом гіпертексту, на відміну від лінійного тексту, здійснюється за рахунок розбивки інформації на фрагменти і встановлення між ними логічних зв'язків. Це дозволяє користувачеві перейти від досліджуваної в теперішній момент теми до однієї з декількох пов'язаних з нею тем.

В представленій навчальній системі по розробці WEB-сторінок інтегровані усі три принципи конструювання навчальних систем.

Розроблена система складається із двох окремих модулів. Перший модуль є послідовністю HTML-файлів, які логічно зв'язані між собою гіпертекстом. В цих файлах розміщується структурований навчальний матеріал у вигляді лінійного тексту, що містить вставки мультимедійних елементів. Їх використання покращує сприйняття, перш за все, графічного матеріалу.

Другий модуль обробляє електронний підручник, формує його структуру залежно від рівня підготовки учня, виконує перевірку даних, що вводяться, тобто забезпечує інтерактивність сторінки. Для забезпечення інтерактивності інтерфейсу використовується інструментарій динамічного HTML (DHTML) [2]. Рівень підготовки учня визначається окремими програмами, в яких міститься результат обробки тестів і інших видів контролю.

Модуль обробки й відображення структури підручника реалізований мовою JavaScript як послідовність скриптів. JavaScript є машинно-незалежною мовою, тому даний модуль здатний працювати на будь-якій апаратній платформі. Розроблена в представленій навчальній системі по розробці WEB-сторінок послідовність скриптів реалізує механізм індивідуального навчання.

Принципова реалізація механізму індивідуального підходу полягає в наступному:

- користувач після ознайомлення з матеріалом проходить ряд послідовних тестів;
- спеціальні програми аналізують результати й видають діалогові вікна з рекомендаціями й оцінкою;

– увесь масив навчальних даних розділений на певні сегменти (структурні блоки), які містять специфічну інформацію, що підбирається залежно від результатів тестування користувача й інших видів перевірок;

– після аналізу результатів і визначення оцінки програма сама підбирає структурні елементи матеріалу й збирає їх у єдиний сумарний розділ, що враховує рівень підготовки й індивідуальні особливості користувача.

Для реалізації зазначених вище принципів індивідуального підходу до навчання в АНС була виконана чітка структуризація навчального матеріалу за темою “Розробка Web-сторінок на основі мови гіпертекстової розмітки HTML” шляхом поділу матеріалу на логічні елементи й структури: розділи, сторінки, параграфи, а також реалізований наступний сценарій навчання.

На першому етапі проводиться ознайомлення користувача з теоретичним матеріалом, далі йому пропонується виконати практичні завдання та пройти ряд послідовних тестів (контроль знань виконують також різні практичні частини). Результати тестів заносяться в змінні (змінна date) та приймаються скриптами, які обробляють результати. Далі ці програми видають діалогові вікна з результатами тестів а також фреймові вікна з аналізом подальшого ходу навчання. Змінна date з оцінкою далі обробляється скриптами-аналізаторами, які формують динамічний масив даних.

Якщо date=2, тобто користувач отримав “незадовільно”, запускається модуль з інструкціями по роботі з програмою, і учень знов повертається до вивчення теоретичної частини.

Якщо date=3 (знання користувача відповідають оцінці “задовільно”), то в цьому випадку для нього скриптами формується новий розділ, який містить той теоретичний матеріал, що не був засвоєний користувачем.

Якщо date=4 (знання користувача відповідають оцінці “добре”), то теми ж скриптами формується інший розділ за тим же принципом, що і для оцінки “задовільно”.

Якщо date=5 (користувач отримав оцінку “відмінно”), розділи не змінюються.

Програмно це реалізується таким чином. На мові MySQL розроблена віртуальна база даних, в якій містяться різні сегменти теоретичного матеріалу. В залежності від значення змінної date, спеціальні експертні програми формують розділ з окремих сегментів. Результуючий розділ формується з урахуванням рівня сприйняття матеріалу окремим користувачем та загальним рівнем його підготовки.

Таким чином, запропонована АНС по розробці WEB-сторінок є інтелектуальною навчальною програмою, яка враховує, перш за все, принципи ефективної організації навчального процесу, реалізує індивідуальний підхід при навчанні, а також вносить творчий аспект у набування нових знань та практичних навичок.

Навчальна система не потребує прямого підключення до мережі Internet і може бути використана при навчанні Internet-технологіям в курсі

інформатики.

Література:

1. Угринович Н.Д. Информатика и информационные технологии. – М.: Лаборатория базовых знаний, 2001. – 464 с.
2. Стауфер Т. Создание WEB-страниц. Самоучитель. – СПб.: Питер, 2003. – 448 с.

ИСПОЛЬЗОВАНИЕ СИСТЕМЫ ДИСТАНЦИОННОГО ИНТЕРНЕТ-ОБУЧЕНИЯ «МООДУС»

Л.В. Гусева, А.В. Шматко
г. Харьков, Академия гражданской защиты Украины
fd.apbu@list.ru

Академия гражданской защиты Украины (АГЗУ) является ведущим вузом Украины по подготовке специалистов для подразделений МЧС Украины и стран ближнего зарубежья. Специфика академического образования требует широкого и оперативного обмена научной, учебно-методической информацией, поэтому одним из направлений развития в АГЗУ является внедрение обучения на базе Интернет-технологий. Нами ставится цель – максимально полно вовлечь обучаемого в процесс обучения, наглядно изложить учебный материал и позволить взаимодействовать с объектом изучения (или его моделью). Использование ИКТ в процессе обучения позволяет сделать сам процесс учебы проще и доступнее.

Основной формой дистанционного образования в настоящее время становится образование с использованием информационных сетей и компьютерных технологий. Преимущества данной формы очевидны: 1) возможность чрезвычайно оперативной передачи на любые расстояния информации любого объема, любого вида (визуальной и звуковой, статичной и динамичной, текстовой и графической); 2) возможность интерактивности с помощью специально создаваемой для этих целей мультимедийной информации и оперативной обратной связи; 3) возможность организации электронных конференций, в том числе в режиме реального времени; 4) возможность работать с учебными материалами так и тогда, когда и как это наиболее удобно пользователю.

Обозначенную задачу, а именно: формирование в АЦЗУ системы информационно насыщенного общедоступного дистанционного образования, помогающего качественно выстроить слушателям самых отдаленных регионов индивидуальные учебные траектории, решает коллектив кафедры «Информатика и компьютерные технологии».

Результатом этой работы должна стать создание и апробация дистанционных элективных курсов с использованием ИКТ.

Основополагающими принципами дистанционного обучения, на наш взгляд, являются: 1) индивидуально-ориентированное обучение, обеспечивающее получение образования через самостоятельно выстраиваемые учащимся образовательные маршруты; 2) организация удаленного взаимодействия обучаемого с педагогом с помощью ИКТ; 3) модульность образовательных программ дистанционных элективных курсов; 4) преобладающее значение самообучения, самостоятельной работы обучаемых в рамках образовательной программы.

Организация учебного процесса при удаленном взаимодействии преподавателя и учащегося тоже имеет ряд особенностей.

Учащийся должен быть полностью информирован, что от него требуется и какова структура курса. Задачи, формы обучения система контроля должны быть четко и ясно описаны.

Стратегия подачи материала определяется в зависимости от решаемых учебных задач. Необходимо использовать известные принципы удобочитаемости.

Обратная связь с учащимся организуется при помощи электронной почты, on-line чатов и форумов.

При реализации интерактивного обучения основной дидактической единицей является курсы, которые состоят из набора модулей, содержащих теоретический и справочный материал, заданий для самостоятельной работы, а также контрольных вопросов (часто в виде тестов) для самопроверки и получения оценки [1]. Курс дистанционного обучения по дисциплине «Компьютерные информационные технологии» разработан на модульной основе: каждый модуль – это стандартный учебный продукт (пособие), включающий четко обозначенный объем знаний и умений, предназначенный для изучения в течение определенного времени, а также это зачетная единица, качество работы с которой фиксируется контрольными заданиями, а также тестовыми средствами. Зачетные единицы набираются учащимся после каждого модуля и выполнения итоговой работы.

Для внедрения дистанционного обучения в АГЗУ была взята система МООДУС [2], предназначенная для работы в среде Интернет. Система распространяется бесплатно, как Open Source-проект, по лицензии GNU GPL.

Система может работать на любом компьютере, на котором работает PHP, и поддерживает практически любую базу данных (MySQL, PostgreSQL, MS SQL, Oracle, Interbase, Foxpro, Access, ADO, Sybase, DB2 и ODBC). В системе сделан большой акцент на защите информации (шифрованные сессии, SSL, аутентификация с LDAP-, IMAP-, POP3-, NNTP-серверов).

Средства этой системы позволяют разрабатывать адаптивные учебные курсы и развивать их, осуществлять как индивидуальный контроль обучения, так и мониторинг учебного процесса, решать задачи управления образовательной деятельностью.

Поддерживаются различные структуры курсов: «календарный», «форум», «дерево»:

- богатый набор модулей-составляющих для курсов: форум, тетрадь, тест, ресурс, опрос, анкета, домашнее задание;

- изменения, произошедшие в курсе со времени последнего входа пользователя в систему, могут отображаться на первой странице курса;

- почти все набираемые тексты (ресурсы, сообщения в форум, записи в тетради, ...) могут редактироваться встроенным WYSIWYG RichText-редактором.

– все оценки (из форумов, рабочих тетрадей, тестов и заданий) могут быть собраны на одной странице (либо в виде файла);

– доступен полный отчет по входу пользователя в систему и работе, с графиками и деталями работы над различными модулями (последний вход, количество прочтений, сообщения, записи в тетрадях);

– e-mail-рассылки новостей, форумов, оценок и комментариев учителей.

Основной проблемой при внедрении ДО является создание электронных курсов. Проектирование учебного электронного издания (курса) представляет собой сложный и трудоемкий процесс, требующий высокой педагогической квалификации. Этот этап во многом совпадает с разработкой учебно-методического комплекса дисциплины. В процессе проектирования курсов требуется подготовить следующую информацию:

– название учебной дисциплины;

– для кого предназначен курс (факультет, специальность, семестр, в котором изучается дисциплина);

– подробный список разделов курса;

– контрольные сроки выполнения каждого раздела (если эти сроки лимитируются);

– текст учебного материала для всех разделов курса в электронном виде;

– рекомендуемая литература, средства мультимедиа и ссылки на веб-сайты для получения дополнительной информации обучаемым;

– программные средства, необходимые учащимся при изучении курса;

– перечень контрольных вопросов и/или заданий по каждому разделу курса;

– для вопросов типа да/нет и вопросов с множественным выбором необходимо подготовить варианты ответов и указать правильные;

– баллы за ответы на вопросы и сумма баллов, необходимая для положительной аттестации учащегося;

– если в курсе предусмотрена коллективная работа, необходимо разработать ее сценарий, темы для обсуждения, обсуждаемые вопросы, коллективные проекты;

– сведения об учащихся: фамилии, имена, отчества, адреса электронной почты, URL (адреса в сети Интернет);

– аналогичные сведения обо всех преподавателях, ведущих курс.

Все эти сведения реализуются в среде МООДУС в процессе работы с «Конструктором курса». После этапа проектирования наступает этап реализации и тестирования. Затем курс сдается «администрации» для проверки.

В АГЗУ на 2006 год предусмотрена реализация курса обучения для подготовки специалистов уровня «Магистр» по специальности «Пожарная безопасность». Продолжительность курса составляет восемьдесят один час. Из них на лекции отведено двенадцать часов, на лабораторные работы -

двадцать восемь. Сорок один час предусмотрен для самостоятельной работы обучающихся.

Курс состоит из двух модулей. Первый – «Системы управления базами данных. Современные компьютерные технологии обработки данных» состоит из следующих тем: 1) создание таблиц в реляционной базе данных; 2) способы работы с информацией в реляционной базе данных; 3) поиск и представление информации в Интернет; электронная почта.

Второй модуль – «Современные системы анализа и обработки статистических данных», в свою очередь, состоит из тем: 1) универсальная система обработки и анализа данных Excel; 2) анализ данных с использованием инструментов Excel.

Во время обучения предусмотрен контроль полученных знаний посредством тестирования. После завершения обучения по двум модулям предусмотрено выполнение контрольной работы в форме теста. Окончательным контролем результатов обучения будет являться экзамен.

Немаловажным аспектом для реализации дистанционного обучения в вузе является привлечение студентов (потребителей), желающих пройти какой-либо курс. Поэтому на первом этапе внедрения системы ДО целесообразно использовать ее в качестве «корпоративной» сети, в рамках основного учебного процесса для работы с удаленными филиалами, совершенствования реализации формы заочного обучения. И только при условии апробации курсов, проверки их качества и при проведении широких рекламных мероприятий возможно использование системы дистанционного Интернет-обучения в коммерческих целях.

Реализация идеи сетевой организации образовательных ресурсов действительно может оказаться очень продуктивной для формирования у выпускников курса способности действовать и быть успешными в ситуации открытого динамично развивающегося общества, где общественный прогресс напрямую зависит от таких качеств личности, как: профессионализм, мобильность, решительность, ответственность, способность усваивать и применять знания в незнакомых ситуациях, способность выстраивать коммуникацию с другими людьми и кооперировать ресурсы для достижения общих целей; а также реализации открытого доступа к качественным образовательным продуктам, программам, взаимодействию с ведущими педагогами.

Литература.

1. Полат Е.С. Дистанционное обучение. Учебное пособие для ВУЗов. – М.: Владос. – 1998. – 192 с.

2. Шматко О.В. Дистанційний on-line курс “Комп’ютерні інформаційні технології (КІТ)” для магістрів управління Академії цивільного захисту України // Проблеми та зміст фундаментальної освіти сучасного інженера: Матеріали Всеукраїнської науково-методичної конференції 24-26 листопада 2004 р. – Харків: НТУ “ХПІ”, 2004. – С. 62-63.

КОРПОРАТИВНА МЕРЕЖА ХМЕЛЬНИЦЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

О.А. Мясіщев, О.П. Кудрявцева
м. Хмельницький, Хмельницький національний університет
alex@tup.km.ua

Структура локальної мережі університету

В теперішній час велика увага на підприємствах, в установах приділяється побудові локальних комп'ютерних мереж та організації їх виходу до всесвітньої мережі Internet. У великих установах, що включають декілька окремо розташованих будівель, де знаходяться підрозділи, офіси та відділи, мережа може нараховувати сотні, а подекуди й тисячі комп'ютерів. Їх об'єднання потребує встановлення та настройки спеціального мережного обладнання, наприклад, мостів, комутаторів, маршрутизаторів. Необхідно також організувати вихід кожного комп'ютеру такої корпоративної мережі в Internet. Розглянемо побудову такої мережі на прикладі локальної мережі Хмельницького національного університету.

На сьогодні інформаційно-комп'ютерний центр університету забезпечує роботу чотирьох каналів зв'язку з двома провайдерами Internet. Загальна пропускна спроможність на прийом складає біля 750 Кбіт/с, на передачу даних (робота web-серверів, e-mail, режим запитів тощо) – 384 Кбіт/с. Зовнішні канали забезпечують доступ до Internet комп'ютерам локальної мережі університету, яких нараховується приблизно 1000 одиниць, що об'єднані у близько 60 дисплейних класів. На рис. 1 представлена загальна схема комп'ютерної мережі університету.

Із схеми видно, що усі 7 учбових корпусів університету зв'язані в єдину мереживу систему. Комунікації між корпусами здійснюються за допомогою оптоволоконного кабелю із швидкістю передачі даних 100 Мбіт/с. Між третім та четвертим корпусами швидкість передачі даних складає 1 Гбіт/с, а між бібліотечним та третім корпусом – 200 Мбіт/с. З використанням мідного кабелю до університету підключені два студентських гуртожитки. Крім дисплейних класів у гуртожитках, самі студенти розширюють мережу, підключаючи свої персональні комп'ютери, встановлені у кімнатах гуртожитків. Так, у 4-му гуртожитку студентами встановлено біля 50 комп'ютерів, що підключені до університетської мережі. За допомогою радіоканалу, із швидкістю 5-10 Мбіт/с організований зв'язок технологічного багатопрофільного ліцею із мережею університету. На території ліцею розміщена одна з кафедр університету. Через Dial-Up організований модемний зв'язок університету із школою №27.

Мереживі технології в середині корпусів університету організовано за допомогою кабелю “кручена пара” із швидкістю 10–100 Мбіт/с. Використана топологія – зірка. У центрі “зірки” встановлені магістральні 100-Мбітні

комутатори, в дисплейних класах 10–100-Мбітні комутатори для робочих груп або концентратори. Для перетворення електричних сигналів (“кручена пара”) в оптичні (оптоволоконний кабель між корпусами) використовуються медіоконвертори та комутатори. В середині локальної мережі адресація комп’ютерів здійснюється з використанням локальних IP-адрес (наприклад, класу В: 172.20.0.0), а адресація сегментів мережі університету, з встановленими в них університетських DNS, web-серверів, поштових, проксі-серверів здійснюється з використанням реальних IP-адрес мереж класу С. Взаємодія між локальними мережами і сегментами університетської мережі з реальними Internet-адресами організовано з використанням статичних маршрутизаторів, згенерованих на базі операційної системи Unix FreeBSD. Для організації пригнічення широкомовного шторму (broadcast) в межах локальної комп’ютерної мережі (наприклад, між бібліотечним корпусом, дослідно-експериментальною базою та 2-м учбовим корпусом) також встановлений комп’ютер маршрутизатор з операційною системою Unix FreeBSD і трьома мережними інтерфейсами на кожну підмережу університету.

В якості розводки локальної мережі по учбовим корпусам, на рис. 2, 3, 4 подані схеми підключення дисплейних класів структурних підрозділів та корпусів. Рис. 2 – в межах бібліотечного корпусу, в межах третього – рис. 3, і четвертого – рис. 4, учбових корпусів. На цих схемах також представлено розподіл IP-адрес між комп’ютерами локальної мережі.

Управління інформаційними потоками в мережі університету

При одночасній роботі великої кількості комп’ютерів в локальній мережі неминучі її перевантаження, особливо коли копіюються файли великого розміру з різних корпусів багатьма користувачами. Проблема для мережі викликають мережні віруси, які сканують інші комп’ютери з метою пересилання небезпечних даних.

Одним із шляхів боротьби з перевантаженням мережі є вибіркова фільтрація трафіку, обмеження потоків передачі даних для окремих підрозділів. З цією метою в мережі на магістральних лініях встановлені керовані комутатори другого рівня, які дозволяють приймати рішення про передачу пакету, ґрунтуючись не тільки на MAC або IP-адресах, але й на параметрах четвертого рівня, таких, наприклад, як порти TCP/IP. В діючій мережі встановлені три таких комутатори фірми D-Link DES3526, що дозволяє обробляти до 800 команд фільтрації по портах TCP/IP. Однак, ці комутатори, що використовують технологію другого рівня комутації, не можуть використовуватись для ефективної фільтрації широкомовного шторму. Програмуючи ці комутатори, вдається відкидати великий потік даних по пересиланню файлів з диска на диск мережі Windows (наприклад, “перекачка” відеофайлів), лишаючи в якості альтернативи роботу з sftp. Налагодження sftp потребує професійних навичок, тому кількість користувачів, що знайомі з цією технологією передачі даних, незначна, а це призводить до різкого зменшенню трафіку локальної мережі.

Загальна схема комп'ютерної мережі університету

(вказана кількість комп'ютерів для учбового процесу)

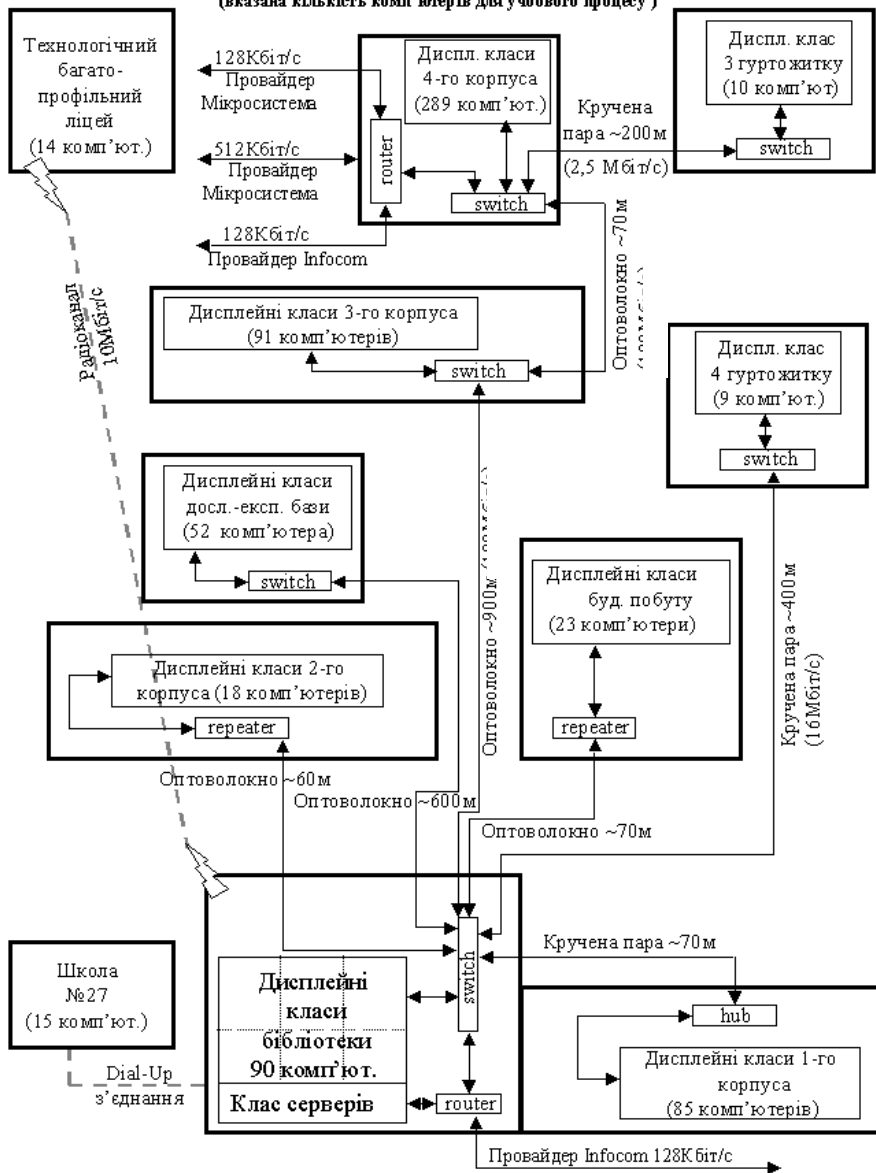


Рис. 1. Загальна схема комп'ютерної мережі університету

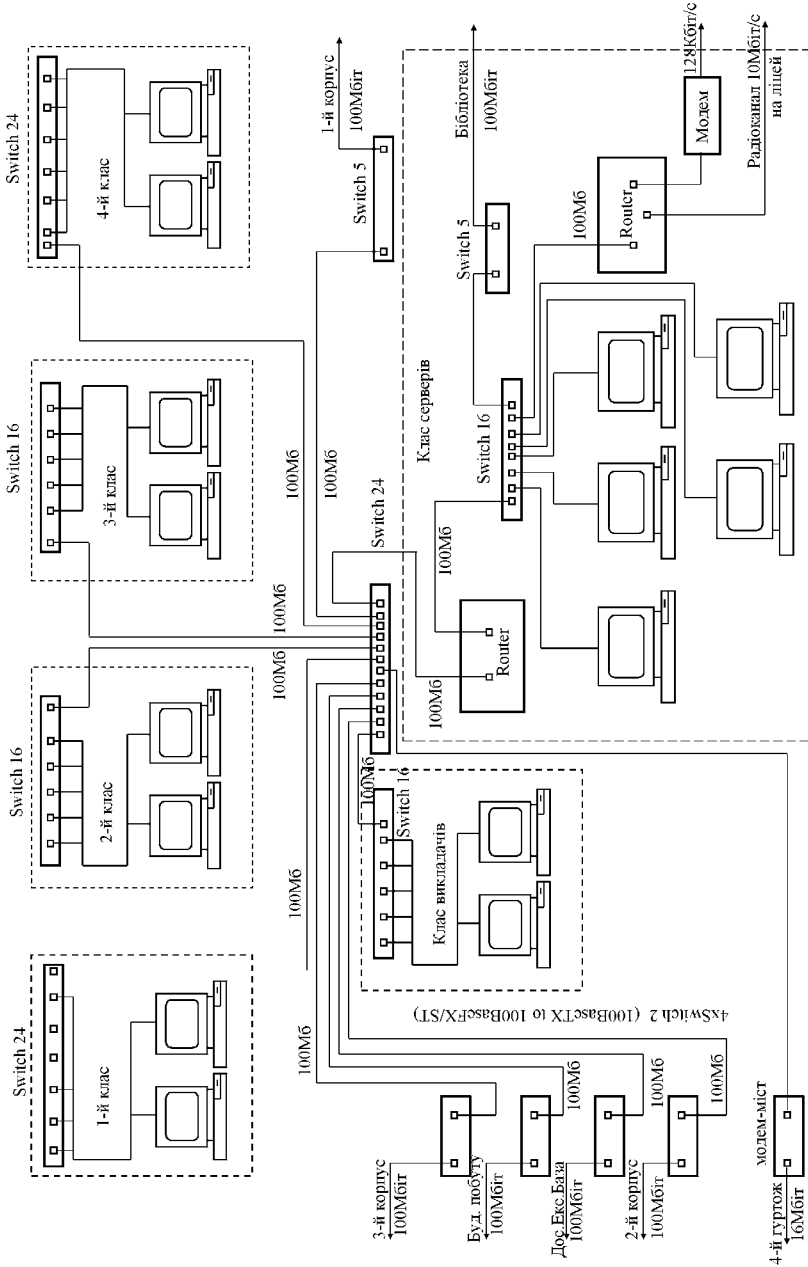


Рис. 2. Спрощена схема 100Мбітної мережі ІКЦ в бібліотечному корпусі

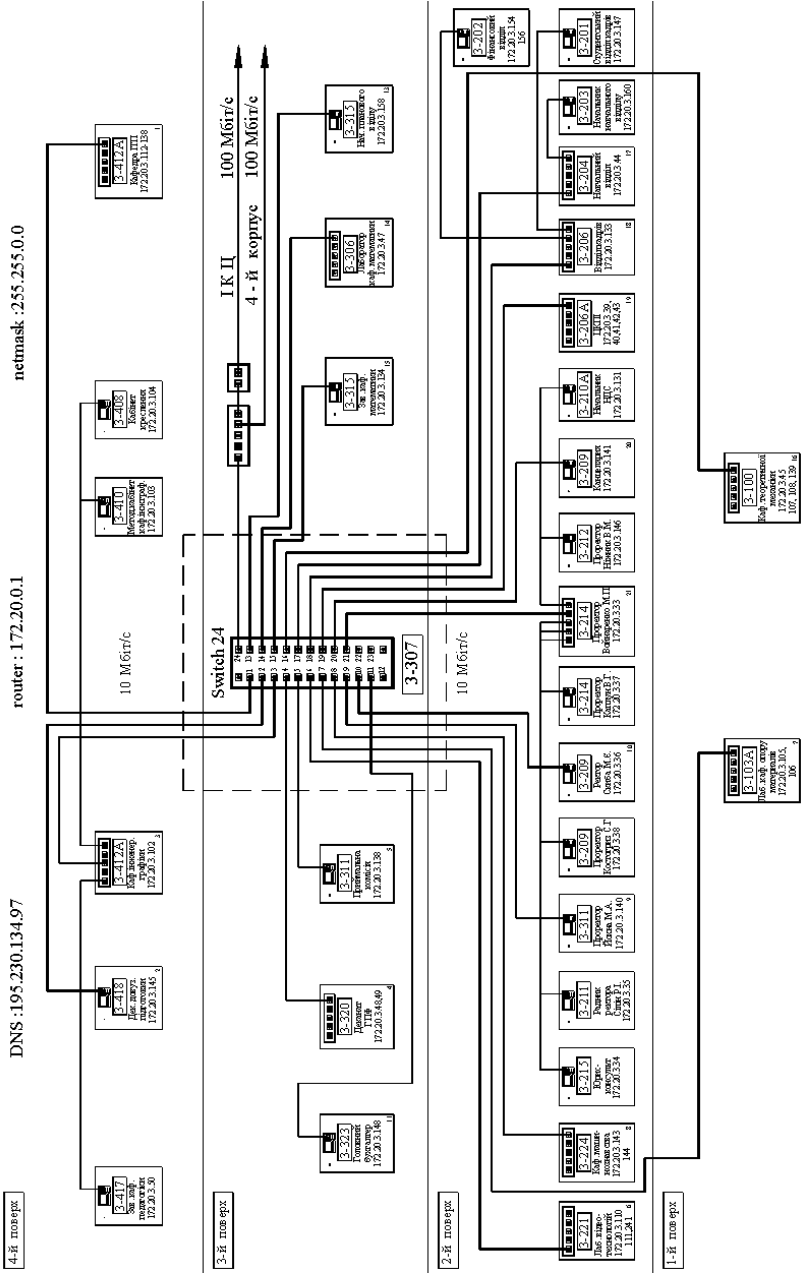


Рис. 3. Підключення дисплейних класів 3-го учбового корпусу до локальної мережі

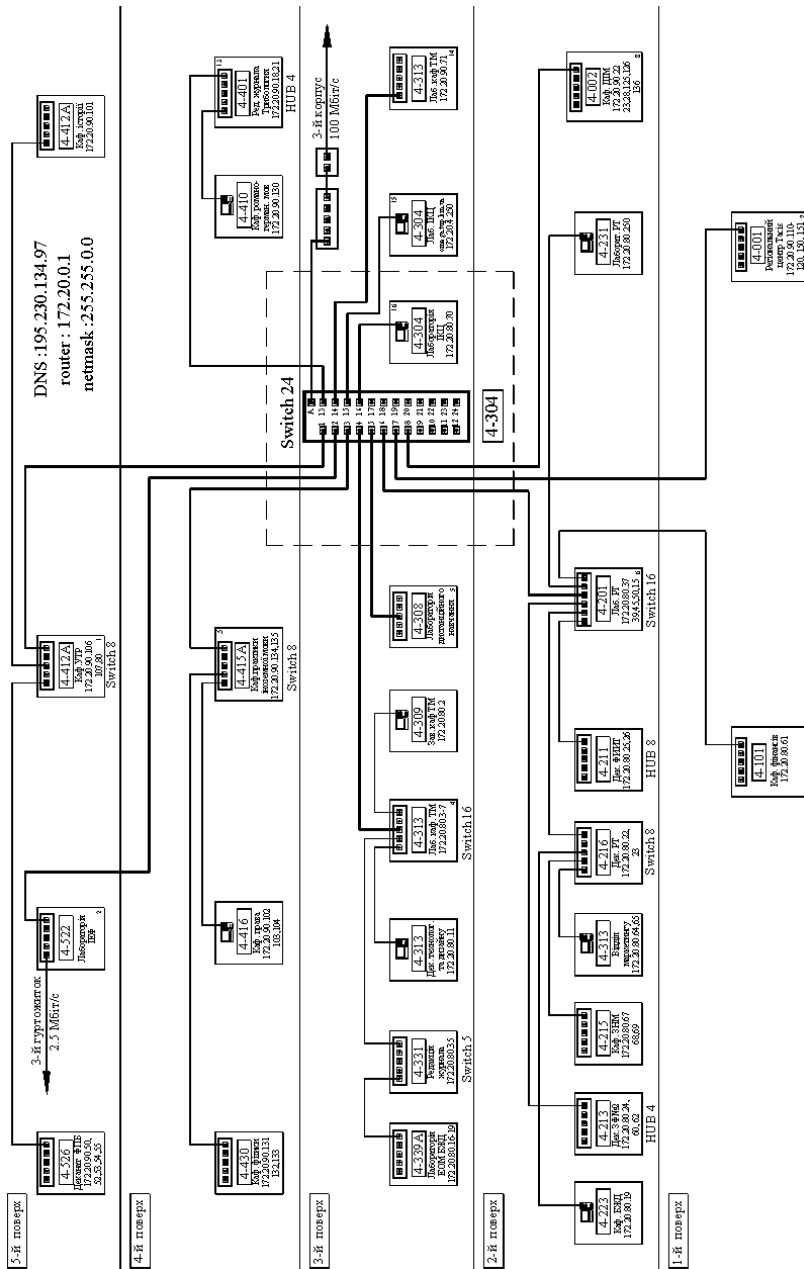


Рис. 4. Підключення дисплейних класів 4-го навчального корпусу до локальної мережі

Так, команди для комутатора DES3526 дозволяють пропускати лише ftp-трафік для усіх IP адрес, решта буде відкинута.

```
Create access_profile ip tcp dst_port_mask 0xFFFF profile_id 15
Config access_profile profile_id 15 add access_id 1 ip tcp
    dst_port 23 port 23 permit
Create access_profile ip tcp src_port_mask 0xFFFF profile_id 16
Config access_profile profile_id 16 add access_id 1 ip tcp
    dst_port 23 port 23 permit
Create access_profile ip source_ip_mask 0.0.0.0
    destination_ip_mask 0.0.0.0 tcp src_port_mask 0x0
    dst_port_mask 0x0 profile_id 20
Create access_profile profile_id 20 add access_id 1 ip source_ip
    0.0.0.0 destination_ip 0.0.0.0 tcp src_port 0 dst_port
    0 port 23 deny
```

Такий підхід дозволив різко зменшити завантаження локальної мережі без переходу до гігабітної технології, яка потребувала б заміну оптоволоконного багатомодового кабелю на одномодовий довжиною ~1000 м.

Не дивлячись на відносно високу пропускну здатність (100 Мбіт/с) магістралей локальної мережі, для якісного забезпечення інформаційними ресурсами близько 1000 підключених до неї комп'ютерів необхідно збільшення її продуктивності. Тому в перспективі планується перехід усієї мережі на швидкості до 1 Гбіт/с для передачі даних між корпусами. Для цього передбачається заміна багатомодового оптоволоконного кабелю між третім та бібліотечним учбовими корпусами на одномодовий, а також установка гігабітного комунікаційного обладнання в інших учбових корпусах. Після здачі в експлуатацію п'ятого учбового корпусу планується його підключення по оптоволокну із швидкістю 1 Гбіт/с і виконання розводки в ньому за технологією "кручена пара" пропускну здатністю 100 та 1000 Мбіт/с.

Для забезпечення університету зовнішніми інформаційними ресурсами локальна мережа має доступ до Internet із швидкістю ~750 Кбіт/с. Проте, при одночасному доступі усіх комп'ютерів до каналів Internet, на кожний комп'ютер, в кращому випадку, біде припадати по 1 Кбіт/с, що звісно є недостатнім для повноцінного отримання інформації. З цією метою, за допомогою розробленого на ІКЦ програмного забезпечення з використанням технології проху-серверу та firewall, організовано обмеження доступу кожного комп'ютеру до Internet по часу, кількості отриманої інформації та швидкості доступу. Так, за допомогою настройки конфігураційного файлу проху-серверу Squid комп'ютери окремих підрозділів мають різні швидкості доступу, наприклад 2,5 Кбіт/с чи 4 Кбіт/с в залежності від поточної необхідності. Причому значення швидкості можна встановлювати заново в будь-який момент. Швидкості також встановлюються і за допомогою firewall на інтерфейсі комп'ютерного маршрутизатора під управлінням Unix FreeBSD. Так, інститут дистанційного навчання та інститут економіки та управління мають прямий доступ до каналів Internet без використання проху-сервера, проте обмеження швидкості здійснюється за допомогою маршрутизатора

Cisco 2600 з використанням команд типу:

```
access_list 101 permit ip any host 212.109.57.236
access_list 102 permit ip host 212.109.57.236 any
traffic_shape group101 48000
traffic_shape group102 48000
```

Ці команди обмежують потік на IP адресу 212.109.57.236 із зовнішнього каналу до швидкості 48 Кбіт/с.

Організація обмежень по часу на набір локальних IP адрес проводиться шляхом перезавантаження проху-серверу з різними конфігураційними файлами для різних IP-адрес в певний час доби засобами операційної системи. Так, IP-адреси студентських комп'ютерів у гуртожитку отримують доступ в Internet через проху-сервер з 21.00 години до 7.00 години, а підрозділи університету через цей же проху-сервер отримують доступ з 7.00 години до 21.00 години. Це досягається додаванням в файл /etc/crontab наступних команд:

```
0021 *** root /usr/local/squid/bin/squid -k interrupt
0121 *** root /usr/local/squid/bin/squid -f
      /usr/local/squid/etc/squid.c
0007 *** root /usr/local/squid/bin/squid -k interrupt
0107 *** root /usr/local/squid/bin/squid -f
      /usr/local/squid/etc/squid.c
```

Робота кожної локальної IP-адреси з Internet організована так, щоб за день комп'ютер міг отримати фіксовану кількість мегабайт інформації. Причому розмір отримуваної інформації може гнучко мінятися адміністратором мережі, а користувач, використовуючи web-браузер, має можливість продивлятися, скільки інформації вже отримано його комп'ютером. Доступ кожного комп'ютера в Internet через проху-сервер здійснюється з використанням логіна та пароля, які відомі співробітнику, що відповідає за комп'ютерну техніку у даному підрозділі.

Механізм відсічки наступний. За допомогою спеціального командного файлу з інтерфейсу проху-серверу зчитується кількість байт інформації, які було передано окремій IP адресі. Кожні 10 хвилин йде співставлення кількості скачаних байт з лімітним значенням для даної IP адреси. Якщо досягнуто перевищення виділеного значення, то відповідний рядок статистики firewall-а підміняється рядком, який забороняє доступ цього комп'ютера до порту проху-сервера. Опівночі відбувається обнуління лічильника отриманих байт інформації і з цього часу комп'ютер знову може отримувати відведений йому ліміт протягом доби.

Крім наукової та освітнянської інформації, в мережі Internet знаходиться велика кількість інформації розважального характеру. Для зменшення завантаження дорогих зовнішніх каналів непотрібною для університету інформацією на проху-серверах організований моніторинг статистики входжень кожної IP-адреси університету у зовнішні сервери і кількість інформації, що отримується звідти. Така статистика може бути переглянута будь-яким ко-

ристувачем університетської мережі через web-браузер при підключенні до web-серверу, який закріплений за конкретним каналом Internet. Статистика зберігається протягом 30 днів для мінімізації завантаження дискових накопичувачів. Періодично системний адміністратор переглядає опубліковану статистику й відмічає зовнішні сервери розважального характеру в конфігураційний файл гроху-сервера для організації заборони доступу до нього для усіх користувачів мережі. Якщо з деяких IP-адрес помічено підключення в основному до таких серверів, розглядається питання про припинення доступу цих комп'ютерів до зовнішніх каналів. Формування статистики доступу виконується програмою SARG, шляхом аналізу log-файлів гроху-серверу Squid.

Основні сервери та лабораторії мережі університету

Інформаційно-комп'ютерний центр забезпечує підтримку серверів, що організовують зв'язок зовнішніх користувачів по модему (дозвонці). На сьогоднішній день встановлено 6 модемів, які дозволяють співробітникам, студентам нашого університету зв'язатися з будь-яким комп'ютером університетської мережі, отримати доступ в Internet, знаходячись вдома, або в організації, що має модем та комп'ютер. Зараз інформаційно-комп'ютерний центр підтримує декілька серверів доменних імен, супроводжує зворотню зону для мережі класу C та зону tur.km.ua. Встановлено два поштові сервери, які працюють незалежно по 2-х каналах зв'язку для їх рівномірного завантаження. На них зареєстровано більш ніж 200 поштових адрес співробітників та студентів університету. Для організації виходу в мережу Internet по чотирьох лініях зв'язку ІКЦ підтримує роботу чотирьох гроху-серверів, що працюють по збалансованій схемі (режим round-robin). Наприклад, якщо один із каналів дуже завантажений, запити перенаправляються до іншого гроху-серверу, що працює з іншим зовнішнім каналом.

В локальній мережі університету встановлений учбовий розрахунковий кластер, що працює під керуванням операційної системи Unix FreeBSD за технологією MPI. На сьогодні в кластер об'єднані два комп'ютери з процесорами Pentium IV, 3 ГГц, що підтримують режим Hyper Threading. Таким чином на кластері запускаються чотирьохпроцесорні задачі. Комп'ютери для організації міжпроцесорних комунікацій об'єднанні гігабітною мережею Ethernet, а з локальної мережі університету по каналу 100 Мбіт/с. Зараз на цьому кластері проводяться учбові заняття із студентами комп'ютерних спеціальностей з дисциплін, які пов'язані з паралельним програмуванням мовами Fortran77 та C.

На рис. 5 представлено фото лабораторії серверів інформаційно-комп'ютерного центру університету.

Центральне місце в локальній мережі університету займають інформаційні сервери. Вони встановлені в університетській бібліотеці, на факультетах, деяких кафедрах, підрозділах. Особливе місце займають сервери ІКЦ, інституту економіки та управління, інституту заочного та дистанційного

навчання, бібліотеки, яким присвоєні реальні IP адреси, DNS імена і тому вони можуть незалежно переглядатись як у межах локальної мережі, так і через Internet.



Рис. 5. Лабораторія серверів ІКЦ університету

На центральному web-сайті Хмельницького національного університету (<http://www.tup.km.ua>, рис. 6.), який підтримується інформаційно-комп'ютерним центром, представлена інформація про структуру університету, керівний склад, інститути, та підпорядковані їм факультети та кафедри, спеціальності за якими ведеться підготовка. На сайті також можна ознайомитись з інформацією про відділ міжнародних зв'язків, про програми, гранти та фонди, школи по вивченню іноземних мов для студентів, викладачів та науковців, про систему освіти за кордоном. Сторінка асоціації випускників надає інформацію служби працевлаштування, є доступ до бази даних вакансій та резюме, а також до бази даних випускників університету різних років.

На сторінці інформаційно-комп'ютерного центру представлена технічна реалізація інформаційного забезпечення учбового процесу, статистика завантаження каналів Internet, учбові посібники, методичні розробки та завдання до лабораторних робіт для студентів.

На сторінці науково-дослідного сектору широко представлена інформація щодо наукових шкіл університету, наукових розробок та технологій, список наукових конференцій на поточний рік. Для бажаючих поступити до

аспірантури корисною буде інформація про терміни вступних іспитів. Деякі наукові підрозділи мають свої окремі сторінки. Так, наприклад, лабораторія композиційних матеріалів за допомогою власної web-сторінки пропагує свої наукові розробки, шукає нових партнерів та замовників.

Хмельницький національний університет

ГОЛОВНА :: КАРТА САЙТУ :: КОНТАКТИ ::

ДЕТАЛЬНІШЕ

- Структура
- Наукова діяльність
- Комп'ютерний центр
- Бібліотека
- Міжнародне співробітництво
- Абітурієнтам
- Асоціація випускників
- Дистанційна освіта
- Інформаційна база даних

НОВИНИ УНІВЕРСИТЕТУ

Міністр освіти і науки України Станіслав Ніколасин побував у Хмельницькому національному університеті

В останні дні весни Хмельницький національний університет відадав міністр освіти і науки України Станіслав Ніколасин. Він перебуваючи на Хмельниччині в рамках проведення Дня депутата, що проходить 27-28 травня в Кам'янець-Подільському. На зустрічі з освітянами, міністр розповів про основні напрямки діяльності Міністерства освіти і науки, щодо покращення ситуації у галузі.

[Детальніше >>>](#)
[Фотогалерея >>>](#)

ЛАБОРАТОРІЯ, ЦЕНТРИ

- Лабораторія композиційних матеріалів
- Подільський науковий фізико-технологічний центр Азотування в тислоцім розряді (Азотування в тислоцім розряді, іонне, іонно-плазменне азотування)
- Навчально-методичне управління

ІНФОРМАЦІЯ

Телефонний довідник ХНУ
Фотогалерея ХНУ
Пошук в Internet
On-line English

КОНТАКТИ

29016 м. Хмельницький,
вул. Інститутська 11,
тел.: (0362)72-80-76
факс: (03622)2-32-65
cent@mailhub.tup.km.ua

На сайті зараз 5 відвідувачів

Червень 2005

Пн	Вт	Ср	Чт	Пт	Сб	Нд
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

18-21 травня у Хмельницькому національному університеті відбулася 4-та міжнародна науково-практична конференція **"Мотивація інноваційно-інвестиційної діяльності підприємств на ринку праці в контексті інтеграції"**

Україна до ЄС* Організаторами конференції були Хмельницька міська рада, Хмельницький національний університет, Інститут економічного прогнозування НАН України, Інститут міжнародних відносин Київського національного університету ім. Т. Шевченка.

Рис. 6. Web-сайт Хмельницького національного університету (<http://www.tup.km.ua>)

Чи не найважливішою складовою web-сайту є сторінка для абітурієнтів. Випускникам шкіл цікавою буде інформація про спеціальності, за якими готують спеціалістів в університеті, а також терміни вступних іспитів та правила прийому до університету. Сторінка бібліотеки містить потужний електронний каталог, пошукову систему, користувачі можуть ознайомитись з новими надходженнями літератури. Також існує і електронна бібліотека ХНУ, де представлена навчальна, наукова та методична література, що може знадобитись студентам в період навчання. На сайті можна користуватись інформаційною базою даних студентів та випускників університету, електронним телефонним довідником підрозділів та служб, дізнаватись про останні події, читаючи сторінку новин.

Сайт створено на основі web-сервера Apache, пакетів програм PHP, та

системи управління базами даних MySQL, операційна система – Unix FreeBSD.

ІКЦ університету підтримує сервер документообігу, куди працівником відділу по контролю та діловодству заносяться усі нормативні документи як внутрішнього, так і загального користування. Відповідальна особа кожного підрозділу, вводячи свій логін та пароль, має можливість ознайомитися з наказами та розпорядженнями керівництва, та доповісти про виконання. Крім того користувачі системи автоматизованого документообігу мають можливість користуватися внутрішнім поштовим сервером для передачі документів та файлів, а також для спілкування між собою.

Як вже зазначалося вище, до комп'ютерної мережі університету на сьогоднішній день підключено близько 60 дисплейних класів, які розташовані на різних кафедрах та факультетах. В інформаційно-комп'ютерному центрі розміщено п'ять дисплейних класів по 14 робочих місць (рис. 7, 8); дві лабораторії серверів, де розташовано обладнання для організації виходу в Internet; одна мультимедійна аудиторія (рис. 9).



Рис. 7. Дисплейний клас ІКЦ

В мультимедійній аудиторії, розрахованій на 90 слухачів, встановлено підключений до мережі комп'ютер викладача з мультимедійним проектором. Маючи доступ до будь-якого внутрішнього та зовнішнього (Internet) інформаційного серверу, викладач проводить заняття з максимальним вико-

ристанням ілюстративного матеріалу. На сьогодні це найбільш популярна та завантажена аудиторія з усіх лекційних аудиторій університету.



Рис. 8. Панорама дисплейних класів ІКЦ



Рис. 9. Заняття в мультимедійній аудиторії

Локальні мережі структурних підрозділів університету

Паралельно до розвитку магістральної частини локальної мережі університету в інститутах та на кафедрах розвиваються власні локальні мережі, які отримують доступ до мережі університету по тій чи іншій технології.

Комп'ютерна мережа **інституту економіки і управління** базується на мережі колишнього інженерно-економічного факультету, створення якої почалося в 1997 році. Основна мета, яка ставилася перед фахівцями факультету, – забезпечення навчального процесу, що використовує сучасні інформаційні і комунікаційні технології; забезпечення ефективної спільної роботи співробітників кафедр і деканату, доступу в Internet і до загальних інформаційних ресурсів університету; забезпечення безпеки і високої швидкості передачі даних між окремими комп'ютерами локальної мережі.

Специфіка роботи користувачів полягає в тому, що ані співробітники, ані студенти, як правило, не мають постійного закріпленого місця за одним з персональних комп'ютерів. Звичайно користувач займає вільний комп'ютер на кафедрі або в учбовому класі. Тому для зручної роботи слід було створити єдине централізоване сховище даних, а також забезпечити централізовану авторизацію при вході в локальну мережу. Для проведення науково-дослідних робіт, а також викладання дисциплін, пов'язаних з клієнт-серверними технологіями, необхідне виділення окремого сервера з установкою відповідного програмного забезпечення.

Таким чином, для забезпечення вказаних функцій слід було організувати замкнуту доменну локальну мережу, яка крім призначених для користувача робочих місць, повинна була мати ряд високопродуктивних серверів, а також шлюз, що забезпечує безпечний вихід у всесвітню мережу Internet і у внутрішню мережу університету. Така структура дозволить обмежити мережевий трафік факультету (інституту) тільки внутрішньою локальною мережею. Середньодобовий трафік, пов'язаний з доступом до внутрішніх серверів університету і мережі Internet, звичайно не перевищує декількох десятків кілобайт в секунду, що повинно сприятливо відобразитися на загальній продуктивності внутрішньої університетської мережевої «магістралі». Дана структура також знижує витрати централізованого адміністрування університетської комп'ютерної мережі з боку інформаційно-комп'ютерного центру, а також підвищує надійність локальної мережі факультету (інституту) при відмовах устаткування, розташованого із зовнішньої сторони шлюзу.

Аналіз розміщення робочих місць для серверного устаткування і призначених для користувача робочих станцій показав, що найбільш ефективним типом мережевої магістралі на даний момент є вита пара, а структура мережі повинна будуватися за двокаскадною комутаційною схемою.

Основний трафік забезпечується центральним комутатором (1 рівень). Експерименти з недорогим устаткуванням середнього класу показали, що на кафедрах і в учбових класах часто виникають “затори”, що приводять до

декількахвилинних пауз при читанні або запису даних на серверах і одночасній роботі більш ніж 12-14 користувачів. Тому для центрального комутатора був вибраний 24-портовий комутатор Саjun 333Т фірми Аwауа (Lucent), що має 100-мегабітні порти з внутрішньою буферною пам'яттю і 8-гігабітною внутрішньою шиною, а також що має можливість розширення за рахунок додаткових модулів. Один високопродуктивний порт центрального комутатора забезпечує прийом-передачу даних однієї кафедри, учбового класу або деканату, оскільки кожен такий підрозділ звичайно розміщується в одному приміщенні. В середині приміщення розводка трафіку по робочих місцях забезпечується комутатором другого рівня, який може мати нижчий клас устаткування (в даному випадку застосовуються 8 або 16-портові комутатори або концентратори фірм 3Com і Planet).

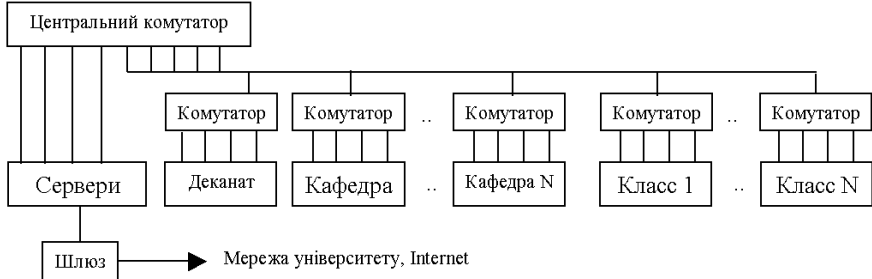


Рис. 10. Двокаскадна комутаційна схема мережі інституту економіки та управління

Для організації локального домену користувачів і мережевих ресурсів використано два сервери, що виконують роль контролерів домену. Два сервери забезпечують резервування бази даних користувачів (на сьогодні їх більше 2000), а також автоматичне балансування навантаження в мережі. Досвід 9-річної експлуатації показав, що така схема забезпечує достатньо високу ефективність і надійність. Один могутній файловий сервер з високошвидкісним дисковим масивом використовується для організації централізованого файлового сховища. На даному файловому ресурсі кожній кафедрі, навчальній групі або підрозділу виділений окремий каталог, усередині якого розміщені особисті каталоги (папки) для зберігання даних окремого викладача або студента.

Вживані апаратні рішення дозволяють виділяти до 200 Мб дискового простору на одного співробітника і до 50 Мб на одного студента. Як шлюз застосовується окремий виділений проксі-сервер з двома мережевими інтерфейсами. Це дозволяє забезпечити необхідну маршрутизацію інформаційних потоків між локальною і зовнішньою мережами з установкою міжмережевого буферного екрану, що забезпечує безпечну роботу користувачів у внутрішній локальній мережі. Додатковий виділений сервер використовується як Web-сервер інституту і сервер електронної пошти. Всі сервери мають мережеві інтерфейси підвищеної продуктивності (фірми 3Com або Intel), які з'єднуються з центральним комутатором. Це забезпечує достатньо комфортну роботу великої кількості користувачів при одночасній роботі в мережі.

При установці комп'ютерного учбового класу в студентському гуртожитку, розташованому на відстані до 1 км від центрального комутатора, було поставлене завдання забезпечити доступ студентам до локальної мережі інституту і можливість виходу в Internet для виконання домашніх завдань і самостійної роботи. Це було досягнуто установкою двох спеціалізованих модемів між центральним комутатором і комутатором учбового класу гуртожитку. Звичайний телефонний кабель забезпечує швидкість передачі даних до 1 мегабіта в секунду. Таке рішення у принципі дає непогану продуктивність при закачуванні файлів з серверів інституту або роботі в Internet.

Тепер локальна мережа інституту економіки і управління охоплює всі поверхи 4-го навчального корпусу від Internet-клубу до методкабінету разом з видаленим сегментом, розташованим в студентському гуртожитку, і забезпечує продуктивну навчальну і навчально-методичну роботу студентів і співробітників інституту, дозволяє ефективно виконувати наукові і дослідницькі розробки.

Варто зупинитися і на інформаційно-комп'ютерному комплексі кафедри інформаційних технологій проектування (ІТП), котрий об'єднує в собі:

- 2 комп'ютерних класи з 13 комп'ютерами кожен;
- лекційну аудиторію на 80 місць, обладнану стаціонарним мультимедійним проектором;
- приміщення для роботи викладачів (кафедру) з 5 комп'ютерами;
- клас з 10 комп'ютерами для самостійної роботи студентів, курсового та дипломного проектування, який розміщений в гуртожитку університету.

Комп'ютерні класи, мультимедійна лекційна аудиторія, кафедра, клас самостійної роботи об'єднані в локальну мережу кафедри та утворюють домен в локальній мережі університету і під'єднані через університетські сервери до Internet. Системним адмініструванням локальної мережі кафедри та всього апаратного комплексу, який складається з сервера, комп'ютерів навчальних класів, класу самостійної роботи, комп'ютерів кафедри, мере-

жевого обладнання, мультимедійного проектора займається технічний персонал кафедри. Всі користувачі кафедрального домену (студенти та викладачі) мають індивідуальний авторизований доступ до мережі кафедри з будь-якого місця локальної мережі університету та відведене для кожного користувача на сервері кафедри місце для збереження необхідної інформації.

Створення інформаційно-комп'ютерного комплексу кафедри дозволяє поєднати в одне ціле основні етапи навчального процесу, а саме:

- методичну та наукову роботу викладачів кафедри;
- проведення лекційних занять з повним використанням інформаційних можливостей комп'ютерної мережі та мультимедійних технологій;
- проведення лабораторних та практичних робіт;
- самостійну роботу студентів в позанавчальний час у гуртожитку.

Всі навчальні та методичні матеріали для забезпечення курсів (електронні варіанти лекцій, лабораторних та практичних робіт, курсового та дипломного проектування, самостійної роботи) зберігаються на сервері кафедри з можливістю отримання інформації в будь-який час і з будь-якого місця локальної університетської мережі. Крім того, на сервері кафедри розміщена електронна бібліотека, де знаходяться підручники, навчальні посібники, довідники, матеріали для поглибленого вивчення тих чи інших дисциплін, наукової роботи та ін., пов'язані з специфікою наукової та навчально-методичної роботи кафедри.

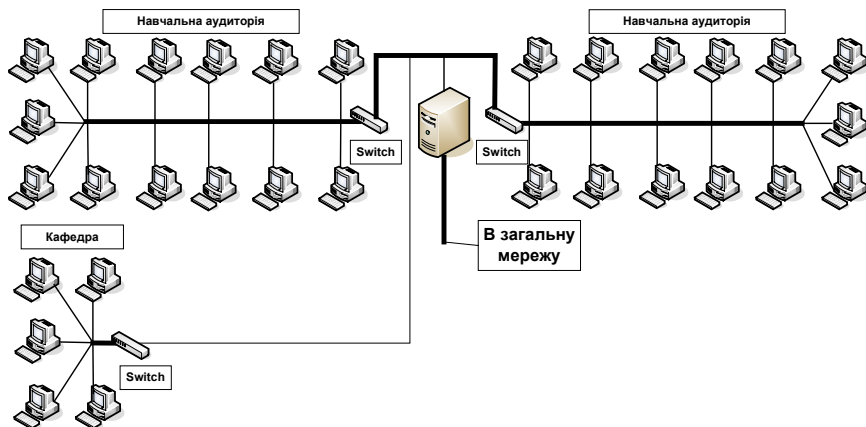


Рис. 12. Схема інформаційного комплексу кафедри ІТТ

Використання навчально-методичних інформаційно-комп'ютерних комплексів дозволяє створювати і всебічно використовувати електронні навчально-методичні посібники, які поєднують декілька дисциплін пов'язаних між собою одним комплексним напрямом. Як приклад такого

методичного посібника, можна навести електронний методичний посібник розроблений доцентом кафедри ІТП М.С. Свірневським по дисциплінам за інтегрованим напрямом «Моделювання інженерних об'єктів і систем автоматизованого проектування», де поєднано три окремі дисципліни: «Основи комп'ютерного моделювання зображень», «Геометричне моделювання у конструюванні інженерних об'єктів та систем» та «САПР конструкторської та технологічної підготовки виробництва».

Створення електронних навчально-методичних посібників дозволяє:

- гармонійно переплітати й нарошувати інформацію при переході від вивчення однієї дисципліни до іншої в міру зростання навичок і знань студентів;
- швидко адаптувати дисципліни під сучасні вимоги, з огляду на особливості кожної дисципліни окремо і специфіки комплексного напрямку;
- швидко відновлювати потрібну інформацію для її використання при вивченні наступних дисциплін, а також виконанні курсових або дипломних робіт з даного напрямку;
- забезпечувати доступ студентів до електронного методичного посібника через локальну мережу;
- підключати учбово-методичну інформацію, представлену в Internet.

На кафедрі машин та апаратів до комп'ютерної мережі університету підключена двухвузлова кластерна система. Вона використовується студентами та співробітниками університету механічних спеціальностей для проведення нелінійного динамічного аналізу методом кінцевих елементів. У якості програмного комплексу використовується пакет Impact. Програма розроблена з використанням мови Java й працює під операційною системою Unix FreeBSD. Комплекс дозволяє знаходити рішення для моделей з пружних та не пружних тіл з лінійними та нелінійними характеристиками. Після розрахунку користувач отримує зміни у часі прискорення, швидкості, переміщення, деформацій в будь-якій точці створеної моделі. Ці дані дозволяють інженеру проводити аналіз елементів та механізмів.

Заключення

Комп'ютерна мережа Хмельницького національного університету – це система, що постійно розвивається. Нові комп'ютери обов'язково підключаються до мережі. Із збільшенням комп'ютерного парку встановлюються сучасні, більш продуктивні мереживі прилади на магістралі мережі, збільшується потужність зовнішніх каналів. Використовуються нові технології для фільтрації трафіка, для організації авторизованого доступу до мережі, проводиться підключення інших освітніх закладів міста, створюються нові інформаційно-пошукові сервери та формуються нові структурні підрозділи. Створюються умови для інформаційного забезпечення освітніх закладів міста в єдину загальноосвітню мережу.

ПОИСК РЕШЕНИЯ ПРОБЛЕМЫ ПЕРЕХОДА СУЩЕСТВУЮЩИХ ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЕЙ К ПОКОЛЕНИЮ 3G

К.Ю. Васильев, Д.В. Сущевский

г. Днепропетровск, Днепропетровский национальный университет

Прошло немногим более 20 лет с момента появления первого мобильного телефона, но мобильная связь уже подверглась существенным изменениям. Системы первого поколения были основаны на аналоговом принципе и использовались исключительно для передачи речи. Системы второго поколения, включающие стандарт GSM, предлагают улучшенную защиту сигнала, дополнительные услуги, низкоскоростную передачу данных, а также автоматическую службу роуминга (стандарт GSM).

Сразу после появления второго поколения мобильных систем началась разработка стандартов систем третьего поколения (3G). Разработки осуществлялись как на региональном, так и на глобальном уровне (ITU-T), результатом чего стало создание в 1985 году инициативной группы, которая в 1996 году была переименована в "IMT-2000". Число 2000 является символом технологии нового тысячелетия и нового частотного диапазона, предназначенного для этой технологии – 2 ГГц. Уже в 1998 году в Европе была разработана система UMTS, которая была предложена ITU-T в качестве основы для создания IMT-2000. Концепция UMTS объединила сети стандарта GSM с наиболее эффективными решениями системы CDMA [1].

Из множества вариантов проектов перехода от существующих сетей к системам 3G специалисты остановили выбор на постепенной интеграции. Именно поэтому важное значение имеет поиск оптимального маршрута модернизации действующих сетей, что и является объектом рассмотрения данной работы.

Для полного уяснения проблемы расшифруем, что же является движущей силой внедрения сетей третьего поколения.

Сотовая связь стоит на пороге массового выхода в Интернет. Это вызвано тем, что традиционные услуги передачи голоса достигают своего насыщения, поэтому для получения новых доходов и компенсации падающего ARPU операторы начинают внедрять дополнительные сервисы, основанные на передаче данных и выходе в Интернет. Распространение этих услуг ограничивается невысокими скоростями передачи данных в действующих сетях сотовой связи. Однако третьему и последующим поколениям телекоммуникационный подход позволит реализовать высокоскоростные приложения и услуги как для бизнес-пользователей, так и для массового потребителя [2].

Улучшенные технические характеристики современных мобильных телефонов позволяют поддерживать:

- речевые вызовы;
- видеотелефонию;

- IP-телефонию;
- видео/аудио потоки: телевидение, видео- и фотосъемку;
- веб-браузинг;
- мобильный офис;
- услуги, основанные на местонахождении клиента: карты и путеводители, ориентацию в незнакомом месте, обеспечение безопасности;
- мобильную электронную коммерцию: оплату билетов, товаров и услуг, поиск и выбор товаров;
- игры.

Перечисленные услуги, на первый взгляд, вызывают сомнение в востребованности конечным потребителем. Они испытывают сильную конкуренцию со стороны других, более дешевых услуг: стационарного Интернета, телевидения, сотовой телефонной связи, GPS, фото- и видеоаппаратуры. Поэтому создается впечатление, что движущей силой разработки систем 3G в мире является не заинтересованность потребителя, как это определяется правилами маркетинга, а интересы производителей аппаратуры сетей и операторов.

Из графика на рис. 1 видно, что рынок сотовой связи в Европе близок к своему насыщению. После довольно благополучных 1999-2000 годов рост общей численности абонентов прекращается.

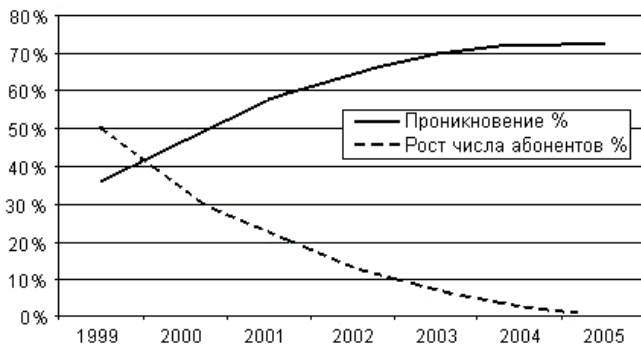


Рис. 1. Распространение GSM в европейских странах

Для операторов сотовой связи это означает прекращение роста доходов, стабилизацию прибыли на достигнутом уровне. Основным мотивирующим фактором для развертывания сетей 3G является возможность увеличить доходы за счет широкого распространения неголосовых услуг, что должно переломить долговременную тенденцию снижения ARPU, характерную для сетей связи второго поколения. В конце 2004 г. доходы от передачи данных составили более 20% от общего дохода 17 ведущих операторов сотовой связи Западной Европы. Предполагается, что за год этот показатель повысится еще на 10%. В настоящее время среднее значение ARPU в стра-

нах Западной Европы равняется 35 евро и продолжает увеличиваться. Другими благоприятными факторами для внедрения сетей нового поколения являются стабилизация финансового состояния предприятий телекоммуникационной отрасли и существенное снижение затрат, требуемых для развертывания новой инфраструктуры. Так, если затраты ввода одного абонентского номера GSM составляют, по разным оценкам 320-350 долл. США, то с учетом эволюционного характера перехода к развертыванию сетей UMTS дополнительное увеличение затрат не превышает 50 долл. [3].

Производителей сотовых телефонов ожидает снижение объемов выпуска аппаратов из-за снижения прироста абонентской базы. Основным направлением их рыночной деятельности становится замена износившихся и устаревших телефонов, срок амортизации которых, по оценкам экспертов, составляет около 1,5 лет. В этой отрасли, несомненно, произойдет спад производства, хотя остаточный его уровень из-за короткого срока амортизации и большой абонентской базы сохранится довольно высоким. Появление новых услуг повлечет за собой спрос на новые модели, поддерживающие эти услуги, поэтому переход на 3G приведет к увеличению доходов отрасли.

Производителям инфраструктуры сотовых сетей уже в ближайший год грозит катастрофическое снижение объема производства из-за прекращения расширения сетей по следующим причинам:

- завершение покрытия территории европейских государств;
- расширение емкости сети не требуется из-за насыщения рынка;
- остаточный уровень производства для замены амортизированного оборудования окажется низким из-за длительного (7–10 лет) срока амортизации.

Таким образом, наиболее заинтересованными во внедрении технологии мобильной связи 3G являются собственники инфраструктуры, выступающие в этом наиболее инициативной стороной.

Учитывая необходимость и неизбежность замены существующих сетей, особую актуальность приобретают вопросы теоретического обоснования принятия тех или иных управленческих решений при проектировании и модернизации сетей, а также проблема их программного моделирования и оптимизации.

Рассмотрим следующую задачу. Вершины представленной ниже сети отображают моменты времени с первого по N -й год. Каждая вершина соответствует моменту времени, когда принимается решение: продолжать использовать данную сеть или модернизировать. Если принимается решение о модернизации, то одновременно должно быть принято решение о длительности ее использования. Дуга от вершины K к вершине $K+1$ представляет решение продолжать эксплуатацию данной сети до конца K года, а затем модернизировать ее. Каждой дуге соответствует число, показывающее суммарные затраты, связанные с решением о модернизации сети. Эти затраты включают в себя дисконтированную (под дисконтированием подразумева-

ется приведение разновременных затрат к их ценности на определенный момент времени) цену на оборудование, текущие затраты, сумму отчислений на ремонт и т.д. Целью рассматриваемой задачи является определение политики модернизации сети за определенный период с минимальными затратами.

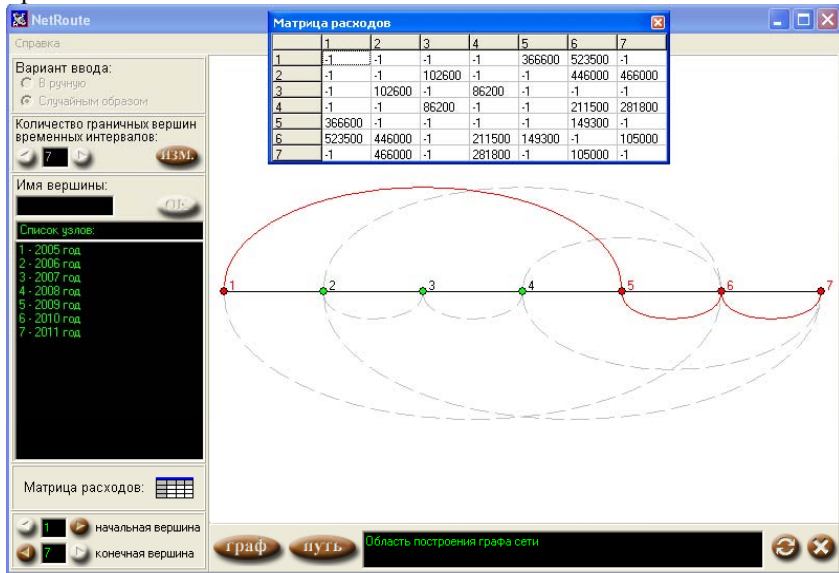


Рис. 2. Окно программы NetRoute

Решение очерченной задачи реализуется учебно-демонстрационной программой NetRoute (рис. 2), созданной в объектно-ориентированной среде программирования Delphi7 на основе алгоритма Дейкстры поиска кратчайшего пути на сети. Подробное описание программы представлено в [4].

Литература:

1. Шиллер Й. Мобильные коммуникации: Пер. с англ. – М.: Издательский дом “Вильямс”, 2002. – 384 с.
2. Ивакин А. Прогрессивное решение от CISCO SYSTEMS – управление контентом на сетях связи. // Мобильные системы. – 2005. – №3. – С. 21-22.
3. Крупнов А.Е., Скородумов А.И. Перспективы формирования рынка услуг нового поколения и операторы виртуальных сетей подвижной связи. // Мобильные системы. – 2005. – №6. – С. 24-25.
4. Васильев К.Ю., Сушевский Д.В. Поиск оптимального маршрута между временными вершинами сети связи. // Матеріали IV Міжнародної науково-практичної конференції “Динаміка наукових досліджень’2005”. Том 50. – Дн-вськ: Наука і освіта, 2005. – С. 13–15.

ПОСТРОЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ КАФЕДРЫ С ИСПОЛЬЗОВАНИЕМ ТЕРМИНАЛЬНЫХ СИСТЕМ

А.В. Шматко

г. Харьков, Академия гражданской защиты Украины

fd.apbu@list.ru

Создание или модернизация информационной системы вуза ставит вопрос о том, как обеспечить оптимальное соотношение стоимости, производительности и надежности. Рассматривая стоимость ИС, надо обязательно учитывать не только стоимость создания, но и стоимость владения системой на интервале ее жизненного цикла. При этом необходимо учитывать стоимость оборудования, программного обеспечения (системного и прикладного), их обновления и обслуживания.

Развитие серверной версии операционной системы Windows привело к появлению в ней терминальных сервисов. То есть, помимо услуг по хранению файлов, поддержке баз данных, выполнению серверных компонент клиент-серверных приложений, теперь у нескольких пользователей имеется возможность выполнять клиентские приложения на сервере в режиме разделения времени. До появления Windows NT Terminal Edition такие возможности были доступны только при установке ПО Citrix WinView или WinFrame, позднее MetaFrame. Стоит заметить, что Unix и Linux системы изначально обладают мультитерминальными возможностями, но их применение ограничено по причине плохой совместимости с Windows-приложениями.

Мультитерминальные информационные системы подразумевают установку на рабочих местах не персональных компьютеров, а Windows-терминалов. На терминале не выполняются клиентские приложения, он служит только для получения доступа к приложениям, работающим на сервере.

В мультитерминальных системах в качестве терминалов можно использовать и персональные компьютеры тоже. Для этого на компьютере достаточно иметь терминальное ПО. В режиме терминала компьютер на своем мониторе просто отображает результаты вывода программ, работающих на сервере, и обеспечивает ввод данных с клавиатуры и мыши. При этом никакая обработка данных на компьютере не производится. Все необходимые пользователю приложения запускаются на сервере в режиме разделения времени.

Понятно, что при этом технические требования к терминалам минимальны. Это могут устаревшие или очень недорогие компьютеры, производительности которых недостаточно для тех приложений, которые запускаются на сервере, но достаточно для запуска терминальной программы.

Для работы информационной системы в терминальном режиме необхо-

дим достаточно мощный сервер, который сможет обеспечить работу нескольких пользователей одновременно с несколькими современными приложениями. Требования к серверу могут быть более высокими, чем к обычному офисному серверу. Во всяком случае, оперативной памяти на терминальном сервере должно быть больше, а система ввода-вывода должна быть быстрее. Эти требования легко выполнить, поскольку компьютерное оборудование постоянно дешевеет, а экономия на остальных элементах системы покрывает дополнительные затраты.

Для обеспечения надежности системы можно использовать несколько серверов с автоматической балансировкой нагрузки. При этом выход из строя одного из серверов не приведет к отказу всей информационной системы предприятия.

Использование терминальной системы вместо сети персональных компьютеров в вузе предоставляет множество выгод:

1. Терминальные технологии снижают общую стоимость владения информационной системой вуза. Понятие общей стоимости владения (ОСВ) информационной системой (Total Cost of Ownership – TCO) включает в себя все расходы на закупку и обновление оборудования и программного обеспечения, проектирование и установку сетевого оборудования, настройку и поддержку системы, обучение пользователей, эксплуатацию (приобретение расходных материалов, ремонт, профилактическое обслуживание), обеспечение безопасности и развитие. Использование терминальных информационных систем позволяет достичь существенной экономии средства с точки зрения TCO, хотя иногда при создании системы с нуля затраты начальных периодов могут немного превышать аналогичные затраты на создание системы на основе клиент-серверных или файл-серверных технологий за счет более высоких требований к серверу. При создании терминальной системы требуется приобрести сервер, несколько терминалов, коммуникационное оборудование. При реконструкции или обновлении существующей системы для перевода ее в терминальный режим достаточно провести обновление или замену сервера. В качестве терминалов можно использовать имеющиеся компьютеры, требования к ним предъявляются минимальные.

Для создания информационной системы с нуля или расширения имеющейся системы более выгодно приобрести специализированное терминальное оборудование. Терминалы имеют ряд преимуществ перед персональными компьютерами при их использовании в корпоративных системах.

В терминале отсутствуют вращающиеся детали, подверженные механическому износу. В терминале нет дисководов и в большинстве моделей терминалов нет вентиляторов, так как используются процессоры с малым выделением тепла. Это означает, что ремонтировать терминал по причине износа не потребуется. Практика свидетельствует, что сроки эксплуатации терминалов без ремонта могут достигать 10 лет и более.

Терминалы потребляют в несколько раз меньше электроэнергии, чем обычный персональный компьютер. С учетом роста тарифов на электроэнергию этот показатель становится важным, особенно при большом количестве рабочих мест.

Основным узким местом обычного сервера является недостаток оперативной памяти. Считается необходимым иметь на сервере от 64 до 128 мегабайт в расчете на каждый активно работающий терминал в системе плюс от 512 до 1024 мегабайт на потребности системы. То есть для системы, обслуживающей одновременно до 20 пользователей, достаточно иметь на сервере от 2 до 4 гигабайт оперативной памяти.

При этом надо иметь в виду, что если основной набор приложений представляет собой Microsoft Office, браузер, почтовый клиент и систему учета типа 1С: Предприятие, то требования к серверу могут быть существенно снижены.

Сетевое оборудование можно приобретать в расчете на небольшой трафик. Если уже имеется оборудование, поддерживающее скорости до 10 Мбит/сек, то его будет вполне достаточно.

2. В терминальных системах приложения работают быстро, производительность труда растет, затраты на сопровождение снижаются. Поскольку в терминале никакого клиентского программного обеспечения не используется (ПО терминала зашито в ПЗУ), обновление ПО для исправления ошибок в терминальной системе производится централизованно только на сервере. Набор и конфигурация программных продуктов под конкретного пользователя определяется записями в профиле пользователя.

Администрирование сервера возможно с единого (в том числе удаленного) рабочего места. То есть, настройка комплекса программного обеспечения непосредственно на рабочем месте пользователя не требуется.

Консультации и обучение пользователей также можно проводить удаленно. Понятно, что для администрирования терминальной системы потребуется значительно меньше трудозатрат, то есть на лицо экономия рабочего времени и персонала.

3. Терминальные системы обеспечивают высокий уровень безопасности при минимальных затратах.

Общая информационная безопасность. Повышение уровня общего уровня информационной безопасности связано с переходом от распределенного к централизованному хранению данных на едином пространстве, доступ к которому легко можно администрировать. Можно исключить доступ к базам данных через локальную сеть, удаленный доступ, так как клиенту передается только изображение экрана, а не информация.

Программы и данные всех пользователей терминальной системы хранятся только на сервере, поэтому даже физическое уничтожение рабочего места пользователя не влечет потерю данных и остановку работы.

Терминальную систему легко защитить от сбоев по питанию. Доста-

точно защитить сервер. Терминалы от сбоев по питанию защищать не надо. Приложения и данные находятся на сервере, защищенном от сбоев. После восстановления питания на терминалах пользователи смогут продолжить работу с теми приложениями и теми данными, которые обрабатывались ими на момент отключения. Фоновые приложения (прием почты почтовым клиентом, формирование отчета в 1С: Предприятии и т.п.) продолжат работу и без подключенного терминала.

Если предполагается длительное отключение питания, то системный администратор терминального сервера может поочередно зайти в сеансы пользователей и закрыть их приложения во избежание потери данных при выключении сервера.

Вирусная устойчивость терминальных систем. Поскольку на рабочих местах нет операционных систем и приложений, то они полностью недоступны для воздействия вирусов и других вредоносных программ. Защищать терминал от вирусов не надо. Защитить же сервер соответствующим анти-вирусным ПО и дисциплиной использования почты, данных и каналов доступа в Интернет можно очень надежно. Отсутствие возможности несанкционированно устанавливать на рабочих местах носители данных, практически полностью исключает этот канал заражения.

Безопасность доступа в Интернет. Можно не пропускать Интернет-трафик в локальную сеть. При этом вся работа, которую надо выполнять в Интернете, может быть выполнена на сервере в терминальном режиме, даже если на каких-то рабочих местах установлены компьютеры. Права доступа в Интернет для пользователя, работающего на терминальном сервере, устанавливаются легко и централизованно.

4. Снижение затрат на модернизация оборудования терминальных систем.

С ростом требований к информационной системе по причине роста объема обрабатываемых данных, обновления программного обеспечения или увеличения рабочих мест возникает необходимость программной и аппаратной модернизации системы. В системах традиционных архитектур необходимость апгрейда относится и к серверам, и к клиентским рабочим станциям, и к сетевому оборудованию. Обновлению подлежит программное обеспечение и на сервере, и на рабочих станциях (персональных компьютерах).

В терминальных системах необходимость апгрейда относится только к серверу и серверному ПО. Терминалы не требуют ни аппаратных, ни программных обновлений. Сеть может потребовать обновления, точнее расширения, только при увеличении количества рабочих мест.

Производительность системы и объем дискового пространства в терминальном варианте легко увеличиваются за счет модернизации (или замены) сервера, заменой и добавлением процессоров, если это было предусмотрено заранее, добавлением или заменой модулей оперативной памяти и

добавлением или заменой дисковых накопителей.

Если сервер изначально допускает такие расширения, то процесс добавления ресурсов в него может быть плавным и многоступенчатым, что не требует от владельца системы крупных единовременных финансовых затрат.

Примером эффективного использования терминалов в учебном процессе может служить система, установленная на кафедре информатики и вычислительной техники Академии гражданской защиты Украины (г. Харьков). Сотрудники кафедры столкнулись с проблемой устаревшего оборудования, а также исключения действий студентов, не связанных с учебным процессом.

В качестве хоста здесь выступает сервер на базе процессора AMD Sempron 2300+ socket-A BUS=333 Mhz 3D Now Pro, оснащенный 1024MB оперативной памяти, накопителем 80GB. Локальная сеть объединяет 12 морально устаревших компьютеров с процессорами Pentium 133-433, с 32-64 MB оперативной памяти. Программный базис решения состоит из ОС Windows 2003 Server SP2. Ресурсами сети преподаватели кафедры пользуются на занятиях по «Информатике и компьютерной технике», «Основах информатики и вычислительной техники» и другим дисциплинам кафедры. В системе прекрасно работают такие программы, как MS Office 2003, WinRAR, Opera, Maple 6, OpenOffice 2.0. Немаловажную роль сыграла дешевизна такой сети. Кафедра получила полноценный работоспособный компьютерный класс с современным ПО, потратив около 2500 грн. на приобретении сервера, а также сетевых карт и кабеля.

Учитывая затраты на модернизацию, техническую поддержку и администрирование терминальной системы, можно с уверенностью сказать, что терминальные информационные системы являются прогрессивными средствами обучения в вузе.

ЗАБЕЗПЕЧЕННЯ СКЛАДСЬКОГО ОБЛІКУ (АРХІТЕКТУРА КЛІЄНТ-СЕРВЕР)

Н.В. Моїсеєнко, Ю.В. Протаскіна
м. Кривий Ріг, Криворізький державний педагогічний університет
n_v_moiseenko@yahoo.com

Однією з найскладніших задач для фірми, що займається торговою діяльністю, є точний і впорядкований облік матеріальних засобів. При великому обороті первинних документів стає дуже складним їх впорядкування. Як правило, багато фірм і досі, при такому стрімкому розвитку комп'ютерної техніки і програмного забезпечення, не мають чітко налагодженого комп'ютерного обліку.

Будь-яка система управління базами даних вимагає адаптації до умов конкретного підприємства, яку організації часто розбивають на дві задачі: проектування даних доручається фахівцям по базах даних, а програмна підтримка виконання транзакцій – програмістам. Реалізація такого підходу, що має, звичайно, свої переваги, пов'язана з розв'язанням ряду серйозних проблем. Треба відверто визнати, що в діяльності розробників баз даних і програмістів існують серйозні відмінності, які визначаються відмінностями в технології і в навичках розробки. Проектувальники баз даних звичайно описують проблемну область в термінах “довгоживучих монолітних таблиць” з інформацією, тоді як програмісти звикли сприймати світ в термінах потоків управління.

Якщо ці два підходи не вдасться об'єднати в рамках одного проекту, то домогтися цілісності проектного рішення для більш менш складної системи буде практично неможливий. Для системи, в якій головне – дані, ми повинні досягти рівноваги між базою даних і програмою. База даних, розроблена без урахування того, як вона надалі використовуватиметься, виявляється, як правило, незграбною і неефективною. У свою чергу, ізольована програма може пред'явити нездійсненні вимоги до бази даних, що призведе до серйозних проблем із забезпеченням цілісності інформації.

Ще в недалекому минулому бізнес-програми виконувалися на великих ЕОМ, що споруджувало для звичайного службовця майже непереборні бар'єри на шляху до потрібної йому інформації. Персональні комп'ютери цю ситуацію змінили: доступні інструменти обробки і зберігання даних укупі з комп'ютерними мережами дозволили з'єднати комп'ютери не тільки усередині офісу, але і між підприємствами, відокремленими один від одного тисячами кілометрів. Одним з основних чинників, що сприяли такій зміні, було упровадження архітектури клієнт-сервер. Різкий перехід до архітектури клієнт-сервер на базі персональних комп'ютерів був викликаний, перш за все, вимогами бізнесу. Перед обличчям збільшеної конкуренції і прискореного циклу випуску нової продукції, виникла потреба в більш швидкому

просуванні товарів на ринок, збільшенні об'єму послуг, що надаються клієнтам, більш оперативному відстеженні тенденцій розвитку ринку, загальному зменшенні витрат.

Крім того, необхідно враховувати, що на створення великої системи скоріше за все буде витрачено декілька років. За цей час сильно зміниться апаратна частина.

Виходячи з вищезазначеного нами була визначена мета, що була досягнута в рамках виконання кваліфікаційної роботи, створити міні-комплекс програмного забезпечення складського обліку в архітектурі клієнт-сервер.

Складський облік – достатньо складна програмна система, що зачіпає всі аспекти, пов'язані з рухом товару на склад і з складу. Для зберігання продукції служить, природно, реальний склад, проте саме програма є його душею, без якої він втратить свою функцію ефективного центру розподілу.

При розробці такої системи замовникам необхідно частково переосмислити весь бізнес-процес і врахувати вже наявні програми, щоб не втратити вкладені засоби. І хоча деяке поліпшення продуктивності ведення справ в компанії може бути досягнуто просто за рахунок автоматизації вже існуючої системи обліку товарів “уручну”, радикального поліпшення можна досягти тільки при кардинальному перегляді ведення бізнесу.

За наслідками системного аналізу можна виділити такі основні функції системи:

1. Облік замовлень: прийом замовлень від клієнтів; відповідь клієнтам за станом замовлень.

2. Ведення рахунків: виставлення рахунків клієнтам і відстеження платежів; прийом рахунків від постачальників і відстеження платежів постачальникам.

3. Відвантаження з складу: складання специфікацій на комплектацію товарів, що відправляються з складу клієнтам.

4. Складський облік: постановка товарів, що прибувають, на облік; зняття товарів з обліку при відправці замовлень.

5. Закупівлі: замовлення товарів постачальникам; відстеження поставок.

6. Отримання: прийняття на склад товарів від постачальників.

7. Планування: випуск звітів, у тому числі що відображають тенденції попиту на окремі види товарів і активність постачальників.

Для роботи потрібно створити базу даних, в якій зберігатиметься вся інформація, наприклад, засобами Access.

У файлі `access.php`, у фрагменті коду `php`, вказується шлях до бази даних, з якою ми працюватимемо. Тут же треба вказати, які поля даної бази може проглянути клієнт. Дані цих полів виводитимуться в окремий файл. Далі в цьому файлі `html`-код описує, які поля будуть створені, а їх кількість повинна відповідати кількості полів, вказаних у фрагменті коду `php`. Ця відповідність необхідна для того, щоб зв'язати поля `html`-сторінки з необхід-

ними полями бази даних. Наступний же фрагмент – фрагмент коду php, який здійснює прив'язку вказаних нами полів. Клієнт, викликаючи кожного разу дану сторінку, має свіжі дані з бази.

Приклад файлу access.php:

```
<?
$db = 'C:\\BAZA\\SERDOLIK.MDB';
$conn = new COM('ADODB.Connection');
$conn->Open("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=$db");
$sql = 'SELECT Марка, ЕдиницаИзмерения, Цена FROM Товары
        ORDER BY Марка';
$rs = $conn->Execute($sql);
?>
<table>
<tr><th>Марка</th> <th>Единица Измерения</th> <th>Цена</th></tr>
<? while (!$rs->EOF): ?>
<tr>
    <td><?=$rs->Fields['Марка']->Value ?></td>
    <td><?=$rs->Fields['ЕдиницаИзмерения']->Value ?></td>
    <td><?=$rs->Fields['Цена']->Value ?></td>
</tr>
<? $rs->MoveNext() ?>
<? endwhile ?>
</table>
<?
$rs->Close();
$conn->Close();
?>
```

Після запуску сервера Apache ми запускаємо програму клієнта. Вона, в свою чергу, звертається до сервера, бази, яка відкривається йому тільки для перегляду і автоматично пересилає йому, вказані у файлі access.php дані.

В ході розв'язання поставлених для досягнення мети роботи задач отримані такі основні результати:

- 1) теоретичний аналіз літератури по проблемі дослідження дає можливість зробити висновок про те, що модель клієнт-сервер може бути використана для системи складського обліку;

- 2) аналіз існуючих серверів приводить до висновку, що найбільш придатним для нашої мети є сервер Apache, який дозволяє працювати в Windows з технологіями PHP, CGI і Perl-скриптами одночасно так само просто і невимушено, як в Unix;

- 3) розроблений програмний міні-комплекс складського обліку, що чітко відображає взаємодію клієнта і серверу, на базі web-серверу Apache з модулем PHP, що інтегрується;

- 4) проведена серія тестів, в ході яких підтверджена правильність і ефективність роботи розробленого програмного забезпечення.

ПРОСТОЙ ПОДСЧЕТ ТРАФИКА ЧЕРЕЗ РОУТЕР CISCO

А.А. Мясичев, В.М. Полозова

г. Хмельницкий, Хмельницкий национальный университет

alex@tup.km.ua

В настоящее время практически все высшие учебные заведения имеют свои локальные сети. Для организации их выхода в Internet обычно используют маршрутизаторы (роутеры). В более развитых университетах для этих целей используют роутеры фирмы CISCO, которые занимают стабильное первое место в мире среди аппаратных решений доступа к Internet.

После установки и настройки маршрутизатора, важной задачей является подсчет трафика, проходящего через него. Приобретение фирменного программного обеспечения подсчета трафика от CISCO очень дорого и покупка его экономически не оправдана. Однако решить эту проблему можно весьма эффективно без затрат на основе заложенных в CISCO свойств, набора несложных программ и использования свободного программного обеспечения.

Предположим, что выход локальной сети университета к провайдеру Internet организован через роутер CISCO 2611 с версией программного обеспечения IOS 12, который имеет один синхронный порт Serial 0/0 (подключен к провайдеру через модем) и один порт Ethernet 0/0 (подключен к локальной сети университета). Задача состоит в том, чтобы сбрасывать на Web-сервер под управлением Unix FreeBSD статистику посещения сайтов конкретными узлами локальной сети университета, которым присвоены реальные IP-адреса с последующим просмотром файлов статистики удаленными пользователями. Обычно под этими IP-адресами работают проху-, почтовые, Web-сервера конкретных подразделений университета. Например, IP-адрес 194.9.68.115 имеет сервер факультета дистанционного образования (dn.tup.km.ua). Он используется как проху-сервер, т.е. к нему подключены рабочие станции всего факультета. Поэтому важно, чтобы информационно-компьютерный центр, который занимается каналами Интернет в пределах университета, имел статистику по каждому подразделению.

Для решения этой задачи, прежде всего, необходимо сконфигурировать роутер CISCO. Ниже приведены команды, которые требуется ввести в конфигурацию роутера CISCO 2611 для возможности снятия статистики.

1. Конфигурируем интерфейс Ethernet 0/0 так, чтобы можно было просматривать исходящие из него пакеты в локальную сеть.

```
Router#configure terminal
Router(config)#interface Ethernet 0/0
Router(config-if)#ip accounting output-packets
```

Теперь маршрутизатор будет собирать исходящие пакеты с этого интерфейса, информацию о которых можно смотреть командой `show ip accounting`.

```
Router# show ip accounting
```

Source	Destination	Packets	Bytes
213.219.242.3	194.9.68.113	3	385
207.46.2.30	194.9.68.115	9	726
213.219.241.3	194.9.68.113	4	492
207.126.111.223	194.9.68.116	11	5558
205.188.1.116	194.9.68.113	2	80
212.158.251.191	194.9.68.123	1	908

```
Accounting data age is 6
```

В CISCO информация о трафике собирается в таблице active, которая постоянно изменяется. Командой show ip accounting просматривается содержимое этой таблицы. Однако вывод требует некоего времени и, следовательно, появляется неучтённый трафик. Для получения всего трафика воспользуемся следующей последовательностью действий

1) clear ip accounting – переводим данные из таблицы active в таблицу checkpointed, где они не изменяются;

2) show ip accounting checkpoint – распечатываем статистику из таблицы checkpointed;

3) clear ip accounting checkpoint – очищаем таблицу checkpointed.

2. Изменения в конфигурации роутера для удалённого получения статистики (удалённого выполнения команд)

```
Router#configure terminal
Router(config)#no ip rcmd domain-lookup
# Разрешаем выполнять удалённые команды (rsh)
Router(config)#ip rcmd rsh-enable
# Локальный пользователь alex на CISCO имеет уровень
# привилегий 8, а
# удаленный пользователь alex на компьютер 172.20.0.149 может
# принимать статистику
Router(config)#ip rcmd remote-host alex 172.20.0.149 alex enable 8
# Разрешить уровню привилегий 8 выполнять следующие команды
# обнулять статистику
Router(config)#privilege exec level 8 clear ip accounting
Router(config)#privilege exec level 8 clear ip accounting check-
point
# собирать статистику
Router(config)#privilege exec level 8 show ip accounting check-
point
# разрешить количество строк в таблице checkpointed равной 10000
Router(config)# ip accounting-threshold 10000
```

Теперь статистику с CISCO можно получать на удаленный компьютер с системой Unix FreeBSD с IP 172.20.0.149 под пользователем alex. Для этого на компьютере достаточно ввести команду:

```
> rsh -l alex 194.9.68.97 show ip accounting
```

Получим распечатку:

Source	Destination	Packets	Bytes
66.98.200.22	194.9.68.113	8	1274
211.43.197.193	194.9.68.115	3	120
207.46.2.30	194.9.68.115	15	1014
193.0.0.193	194.9.68.113	1	128
81.19.70.1	194.9.68.116	76	58071
64.156.215.8	194.9.68.115	21	1219

Accounting data age is 9

На основании изложенного, последовательность работы сервера с IP-адресом 172.20.0.149 может быть следующая. В некотором каталоге создается, например, каждые 30 минут файл `stat` с общей статистикой. Затем этот файл обрабатывается программой, написанной на Perl, и полученные файлы разбрасываются по соответствующим каталогам, к которым имеет доступ Web-сервер. Например, это файл суммарной статистики для всех потребителей локальной сети, файл для факультета дистанционного образования и др. подразделений.

Файл, который должен запускаться из `cron` через определенный интервал времени (например, каждые 30 мин.), имеет вид

```
rsh -l alex 194.9.68.97 clear ip accounting checkpoint
rsh -l alex 194.9.68.97 clear ip accounting
rsh -l alex 194.9.68.97 show ip accounting checkpoint > stat
/usr/home/alex/cis_stat/cod.pl
```

Следующий командный файл может запускаться через `cron` например, раз в неделю для удаления всей статистики (чтобы не переполнять диск).

```
rm /usr/home/alex/public_html/summ/*
rm /usr/home/alex/public_html/dn/*
```

Программа на Perl (`cod.pl`), формирующая общую статистику, помещаемую в каталог `/usr/home/alex/public_html/summ`, и статистику для пользователя с IP-адресом 194.9.68.115, который представляет собой проху-сервер, имеет вид:

```
#!/usr/bin/perl
open(OLD, "stat");
@oldfile=<OLD>;
close(OLD);
$i=0;
foreach $aa (@oldfile)
{
$aa=~s/\\r//g;
($sour, $dest, $pack, $byt) = split( ' ', $aa );
$Source[$i]=$sour;
$Destin[$i]=$dest;
$Byte[$i]=$byt;
$i++;
}
$k=$i;
# Определение времени
```



```

@AD = localtime;
$Date = sprintf("%4d-%02d-%02d_%02d:%02d", $AD[5]+1900, $AD[4]+1,
$AD[3], $AD[2], $AD[1]);
# Создание в каталоге summ файлов суммарной статистики
open(NEW, ">/usr/home/alex/public_html/summ/$Date");
$sum=0;
$host=0;
print NEW "Узел-источник\t\t", "Потребитель\t\t", "Byte\n\n";
for( $i=2; $i<=$k-2; $i=$i+1 )
{
print NEW $Source[$i], "\t\t", $Destin[$i], "\t\t", $Byte[$i], "\n";
$sum=$sum+$Byte[$i];
$host=$host+1;
}
print NEW "\nВсего полученной информации в байтах = ", $sum;
print NEW "\nВсего посещено сайтов = ", $host;
close(NEW);
# Создание в каталоге dn файлов-статистики для dn.tup.km.ua
# (194.9.68.115)
open(NEWdn, ">/usr/home/alex/public_html/dn/$Date");
$sumdn=0;
$hostdn=0;
print NEWdn "Узел-источник\t", "Потребитель-dn.tup.km.ua \t",
"Byte\n\n";
for( $i=1; $i<=$k-1; $i=$i+1 )
{
if ($Destin[$i] eq "194.9.68.115"){
print NEWdn $Source[$i], "\t\t", $Destin[$i], "\t\t", $Byte[$i], "\n";
$sumdn=$sumdn+$Byte[$i];
$hostdn=$hostdn+1;
}
}
print NEWdn "\nВсего полученной информации в байтах = ", $sumdn;
print NEWdn "\nВсего посещено сайтов = ", $hostdn;
close(NEWdn);

```

Результатом работы рассмотренной простой системы по снятию статистики, например для факультета дистанционного образования, является следующая распечатка, снятая с Web-сервера по адресу http://cel.tup/~alex/dn/2006-01-21_11:30:

Узел-источник	Потребитель-dn.tup.km.ua	Byte
207.46.2.30	194.9.68.115	624
210.17.38.21	194.9.68.115	124
80.68.242.22	194.9.68.115	4695
195.161.118.231	194.9.68.115	10037
199.174.114.19	194.9.68.115	102862
81.222.134.12	194.9.68.115	5738

81.19.70.1	194.9.68.115	44274
217.16.19.219	194.9.68.115	11149

Всего полученной информации в байтах = 1890756

Всего посещено сайтов = 73

Литература:

1. Система учета трафика для маршрутизаторов CISCO. Ниеншанц-Телеком. – <http://www.linuxcenter.ru/lib/articles/networking/nnz-cisco.phtml>

ИСПОЛЬЗОВАНИЕ ПРОТОКОЛА STP ДЛЯ УСТРАНЕНИЯ ПЕТЕЛЬ И РЕЗЕРВИРОВАНИЯ СВЯЗЕЙ

А.А. Мясичев, В.М. Полозова
г. Хмельницкий, Хмельницкий национальный университет
alex@tup.km.ua

В настоящее время большое внимание уделяется построению локальных сетей масштаба предприятия с повышенной надежностью каналов связи. Одним из путей повышения надежности – это организация альтернативных каналов связи. Для оборудования, работающего на втором уровне модели OSI (мосты, коммутаторы), реализующего технологию прозрачного моста и не умеющего работать в сетях с петлями, для автоматического перевода в резервное состояние альтернативных связей широко используется алгоритм покрывающего дерева (Spanning Tree Algorithm, STA). В этом случае среди множества путей только один становится активным, и через него передаются данные. Рассмотрим основные принципы технологии STA и практически выполним настройку коммутаторов фирм D-Link (DES-3326SR) и Allied Telesyn (AT-8224XL) для реализации этого алгоритма. Данный материал излагается в учебном курсе “Компьютерные сети” специальности “Компьютерные системы и сети” в Хмельницком национальном университете.

При изложении материала рассмотрим следующие вопросы:

1. Понятие пакета BPDU
2. Технология создания покрывающего дерева
3. Пример построения покрывающего дерева для 3-х коммутаторов
4. Последовательность настройки коммутаторов для реализации STA

1. Понятие пакета BPDU

Протокол, реализующий алгоритм STA называют Spanning Tree Protocol (STP). Обмен информацией между коммутаторами проводится с помощью пакетов Bridge Protocol Data Unit, BPDU (рис. 1). Они содержат следующую основную информацию для построения топологии сети без петель:

1. Идентификатор коммутатора, на основании которого выбирается корневой коммутатор
2. Расстояние от коммутатора-источника до корневого коммутатора (стоимость корневого маршрута)
3. Идентификатор порта

Пакеты BPDU помещаются в поле данных канального уровня. Коммутаторы обмениваются BPDU через равные промежутки времени (1–4 с). В случае отказа моста, соседние коммутаторы, не получив BPDU в течение заданного времени, начинают пересчет топологии сети.

Protocol Identifier	1
	2
Protocol Version Identifier	3
BDPU Type	4
Flags	5
	6
	7
	8
	9
Root Identifier	10
	11
	12
	13
	14
Root Path Cost	15
	16
	17
	18
	19
	20
Bridge Identifier	21
	22
	23
	24
	25
Port Identifier	26
	27
Message Age	28
	29
Max Age	30
	31
Hello Time	32
	33
Forward Delay	34
	35

Рис. 1. Структура пакета BPDU конфигурационного типа

Пакет BPDU включает следующие поля:

1. Протокол ID (Protocol Identifier) – идентификатор протокола.
2. Версия (Version) – поле версии протокола.
3. Тип сообщения – поле имеет нулевое значение при нормальной работе протокола. Оно задается равным 80h при извещениях об изменениях в сетевой топологии.
4. Флаги (Flags) используют только два бита. Первый бит сигнализирует об изменениях в сетевой топологии. Второй бит – подтверждения изменения конфигурации.

5. Корневой ID (Root ID) – идентификатор корневого коммутатора. Поле состоит из восьми байт. Два первых байта – идентификатор коммутатора, оставшиеся шесть байтов – его MAC-адрес.

6. Стоимость пути до корня (Root Path Cost) характеризует суммарную стоимость пути до корневого коммутатора.

7. ID коммутатора (Switch ID) – идентификатор отправившего сообщение коммутатора.

8. Идентификатор порта (Port ID) – идентификатор порта коммутатора, с которого отправлено сообщение.

9. Возраст сообщения (Message Age) – время, прошедшее с момента отправки корневым коммутатором сообщения об обнаружении изменений в сетевой топологии.

10. Максимальный возраст (Maximum Age) – время удаления из обращения текущего сообщения.

11. Время приветствия (Hello Time) – промежуток времени между расылками сообщений корневым коммутатором.

12. Задержка перехода (Forward Delay) – задержка, которую коммутаторы должны выждать перед переходом в новое состояние после изменений сетевой топологии.

Пакет BPDU уведомления о реконфигурации имеет следующие поля:

1. Протокол ID (Protocol Identifier) – идентификатор протокола.
2. Версия (Version) – поле версии протокола.
3. Тип сообщения – задается равным 80h при извещениях об изменениях в сетевой топологии.

2. Технология создания покрывающего дерева

При автоматическом выборе корневым становится коммутатор с наименьшим значением MAC-адреса его блока управления. Далее, для каждого коммутатора определяется корневой порт (root port) – это порт, который имеет по сети кратчайшее расстояние до корневого коммутатора. Для каждого логического сегмента сети выбирается так называемый назначенный мост (designated bridge), один из портов которого будет принимать пакеты от сегмента и передавать их в направлении корневого моста через корневой порт данного моста. Такой порт называется назначенным портом (designated port). Назначенный порт сегмента имеет наименьшее расстояние до корневого моста, среди всех портов, подключенных к данному сегменту. Назначенный порт у сегмента может быть только один. У корневого моста все порты являются назначенными, а их расстояние до корня полагается равным нулю. Корневого порта у корневого моста нет. Также при построении покрывающего дерева важную роль играет понятие расстояния. По этому критерию выбирается единственный порт, соединяющий каждый коммутатор с корневым коммутатором, и единственный порт, соединяющий каждый сегмент сети с корневым коммутатором. Все остальные порты пе-

реводятся в резервное состояние, то есть такое, при котором они не передают обычные кадры данных. При таком выборе активных портов в сети исключаются петли и оставшиеся связи образуют покрывающее дерево. В качестве расстояния в STA используется метрика (или т.н. Path Cost) – она определяется как суммарное условное время на передачу данных от порта данного коммутатора до порта корневого коммутатора. Условное время сегмента рассчитывается как время передачи одного бита информации и измеряется в 10-наносекундных единицах. Так, для сегмента Ethernet на 10 Мбит/с условное время равно 10 условным единицам.

3. Пример построения покрывающего дерева для 3-х коммутаторов

Исходные данные для коммутаторов представлены на рис. 2. В этом примере с помощью STP блокируется соединение между коммутаторами С и В.

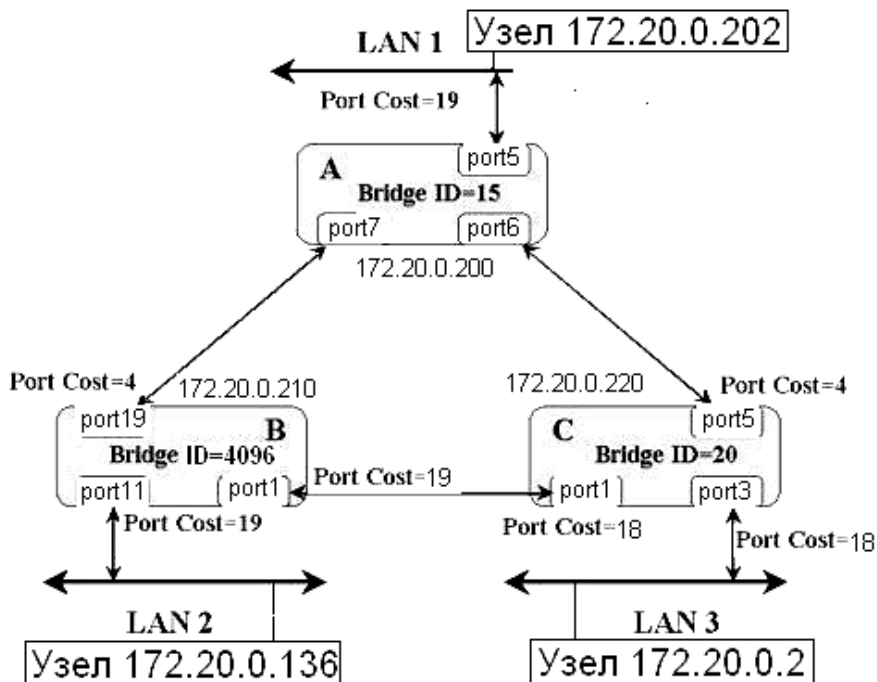


Рис. 2

После инициализации каждый коммутатор сначала считает себя корневым. Поэтому он начинает через интервал hello генерировать через все свои порты сообщения BPDU конфигурационного типа. В них он указывает свой идентификатор в качестве идентификатора корневого коммутатора, расстояние до корня устанавливается в 0, а в качестве идентификатора пор-

та указывается идентификатор того порта, через который передается BPDU. Как только коммутатор получает BPDU, в котором имеется идентификатор корневого коммутатора, меньше его собственного, он перестает генерировать свои собственные кадры BPDU, а начинает ретранслировать только кадры нового претендента на звание корневого коммутатора. При ретрансляции кадров он наращивает расстояние до корня, указанное в пришедшем BPDU, на условное время сегмента, по которому принят данный кадр.

Рассмотрим выборы корневых портов коммутаторами на рис. 2. Поскольку коммутатор А имеет меньший ID, он становится корневым. Коммутатор А посылает BPDU с порта 7 со стоимостью пути, равной 0. Когда В получает пакеты через порт 19, он добавляет стоимость пути к пакету равную 4. Далее мост В запоминает эту стоимость и рассылает пакеты BPDU через порт 1 и порт 11 со стоимостью, равной 4. Мост С на порт 1 получает пакет BPDU со стоимостью от корня, равной 4, добавляет к ней стоимость своего 1-го порта (18) и запоминает ее (22).

Мост А через порт 6 посылает BPDU к мосту С со стоимостью пути, равной 0. Порт 5 моста С добавляет к BPDU стоимость, равную 4, и запоминает ее. Мост С сравнивает стоимость пакетов BPDU, пришедших от А на порт 1 (22) и порт 5 (4), делает заключение, что порт 5 моста С является корневым.

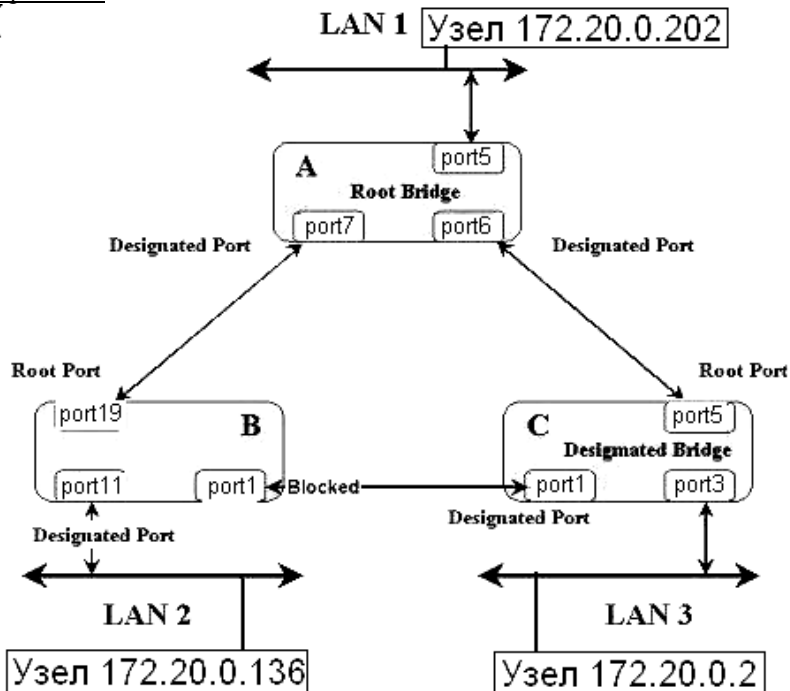


Рис. 3

В свою очередь, мост С рассылает через порт 1 и порт 3 пакеты BPDU моста А со стоимостью 4. Мост В получает на порт 1 от моста С пакет стоимостью 4 и добавляет к ней стоимость своего порта 1 (19), что в итоге равно 23. Коммутатор В сравнивает стоимости BPDU, полученные с порта 19 (4) и порта 1 (23). Делает заключение, что порт 19 корневой.

Рассмотрим выбор назначенных портов.

Мост А посылает BPDU со стоимостью 0 через порт 7 на порт 19 коммутатора В. Порт 19 моста В добавляет стоимость 4. Далее пакет BPDU проходит через порт 1 (В) со стоимостью 4, запоминается там и идет на порт 1 моста С также со стоимостью 4 и там запоминается.

С другой стороны мост А посылает BPDU со стоимостью 0 через порт 6 на порт 5 моста С. Порт 5 добавляет стоимость 4. Далее пакет BPDU проходит через порт 1 (С) со стоимостью 4, запоминается там и идет на порт 1 моста В также со стоимостью 4 и там запоминается.

Мост С сравнивает стоимость на своем порту 1 для пакета BPDU, который он переслал В (4) и получил от В (4). Стоимости оказались одинаковыми. Он тогда определил ID пакета с моста В (4096). Сравнил со своим ID(20). Оно оказалось меньшим и мост С перевел свой порт 1 в состояние назначенного. В свою очередь аналогично мост В перевел свой порт 1 в заблокированное состояние.

Результат работы алгоритма STA приведен на рис. 3.

4. Последовательность настройки коммутаторов для реализации STA

Находясь на любой рабочей станции, имеющей выход в сеть, подключаемся через telnet сначала к компьютеру cel.tup, а затем от него к коммутаторам:

```
telnet 172.20.0.200 -> коммутатор А (AT-8224XL)
telnet 172.20.0.210 -> коммутатор В (DES-3326SR)
telnet 172.20.0.220 -> коммутатор С (AT-8224XL)
```

Для коммутатора А:

1. В основном меню выбираем раздел Bridging и входим в него (рис. 4)
2. Далее выбираем Port spanning tree configuration и входим в него (рис. 5)
3. Поочередно заходим в порты 5, 6 и 7 и устанавливаем Cost=19 (рис. 6)
4. Обратнo последовательно выходим в раздел Bridging, и входим в Spanning tree parameters
5. Устанавливаем здесь Bridge Priority = 15 (рис. 7)

Аналогично настраивается коммутатор С. Для него Bridge Priority =20, порт 1, 3 имеют cost=18, а порт 5 – cost=4.

Коммутатор В фирмы D-Link настраивается в командном режиме.

1. Разрешить STP глобально на коммутаторе
DES-3326SR:4#enable stp
2. Задать приоритет коммутатору

Allied Telesyn AT-8224XL Ethernet Switch: 2.2.0
h4
Main Menu

Port status and configuration
Ethernet statistics
Administration
System configuration
Traffic/Port Mirroring
Virtual LANs/QoS
Bridging
MAC Address Table
Quit

Рис. 4

Bridge Menu
h4

Spanning tree parameters
Port spanning tree configuration
Return to Main Menu ...

Рис. 5

Bridge Menu
h4
Port 5

> **Enable Spanning Tree**
Disable Spanning Tree

Priority: 128
Cost: **19**

Return to previous menu ...

Рис. 6

```

Bridge Configuration Menu
h4
Bridge Identifier (Mac Address : Priority)      003084 8de0f1 : 15
Root Bridge Identifier (Mac Address : Priority) 003084 8de0f1 : 15
Cost to the Root                             0
Port closest to the Root                     0
Max Age                                       20
Forwarding Delay                             15

```

```

Bridge Priority:      15
Max age time:        20
Hello time:          2
Forwarding delay:    15

```

Return to Bridge Menu ...

Рис. 7

DES-3326SR:4#config stp priority 4096

3.Сконфигурировать порты 1,11,19 со значениями cost=19,19,4 соответственно

DES-3326SR:4#config stp ports 1,11 cost 19 state enabled

DES-3326SR:4#config stp ports 19 cost 4 state enabled

Проверить правильность установки портов можно по команде

DES-3326SR:4#show stp ports (рис. 8)

Port	Designated Bridge	State	Cost	Pri	Edge	P2P	Status	Role
1	0014/00308463DDBB	Yes	19	128	No	Yes	Discarding	Alternate
2	N/A	Yes	250	240	No	Yes	Disabled	Disabled
3	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
4	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
5	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
6	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
7	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
8	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
9	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
10	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
11	1000/000D88D7AC20	Yes	19	128	No	Yes	Forwarding	Designated
12	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
13	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
14	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
15	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
16	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
17	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
18	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
19	000F/0030848DE0F1	Yes	4	128	No	Yes	Forwarding	Root
20	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
21	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled

CTRL+C ESC Quit SPACE Next Page Previous Page Refresh

Рис. 8

Здесь видно, что порт 19 является корневым, порт 11 назначенным для сети LAN2, а порт 1 переведен в заблокированное состояние (Status - Discarding).

Проверить правильность настройки STP можно с помощью команды

```

DES-3326SR:4#show stp (рис. 9)

Bridge Parameters Settings
STP Status      : Enabled
Max Age         : 20
Hello Time      : 2
Forward Delay   : 15
Priority         : 4096
STP Version     : RSTP
TX Hold Count   : 3
Forwarding BPDU : Enabled

Bridge Current Status
Designated Root Bridge : 00-30-84-8D-E0-F1
Root Priority           : 15
Cost to Root           : 4
Root Port              : 19
Last Topology Change   : 78103sec
Topology Changes Count : 38
Protocol Specification : 3
Max Age                : 20
Hello Time             : 2
Forward Delay          : 15
Hold Time              : 3

```

Рис. 9

Здесь видно, что корневой коммутатор имеет приоритет (ID) = 15 и его MAC адрес 00-30-84-8D-E0-F1

Изменим исходные данные.

Установим для моста С ID=4097, и по коммутатору В отследим изменения:

```

DES-3326SR:4#show stp ports (рис. 10)

```

Port	Designated Bridge	State	Cost	Pri	Edge	P2P	Status	Role
1	1000/000D88D7AC20	Yes	19	128	No	Yes	Forwarding	Designated
2	N/A	Yes	250	240	No	Yes	Disabled	Disabled
3	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
4	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
5	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
6	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
7	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
8	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
9	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
10	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
11	1000/000D88D7AC20	Yes	19	128	No	Yes	Forwarding	Designated
12	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
13	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
14	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
15	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
16	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
17	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
18	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
19	000F/0030848DE0F1	Yes	4	128	No	Yes	Forwarding	Root
20	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled
21	N/A	Yes	*200000	128	No	Yes	Disabled	Disabled

Рис. 10

Изменения произошли, и порт 1 моста В стал назначенным для сегмента В-С, а порт 1 моста С перешел в заблокированное состояние. Следовательно, если стоимости портов одинаковы, учитывается идентификатор коммутатора.

Литература:

1. Коммутаторы локальных сетей D-Link. Учебное пособие.
<http://www.dlink.ru/technology/index.php>

ИНСТРУМЕНТАЛЬНЫЕ МЕТОДЫ В БИЗНЕС-ОБРАЗОВАНИИ

Б.А. Железко

Беларусь, г. Минск, Белорусский государственный экономический
университет
BorisZh@yandex.ru

Обобщенно существо любого инструментального метода можно представить следующей структурной формулой: *способ решения проблемы + средства его реализации + методика их применения*). Инструментальные методы экономики позволяют оперативно (часто даже в реальном времени) и эффективно (со значительным сокращением затрат финансовых и других видов ресурсов) производить оценки различных альтернативных вариантов решений с целью выбора наилучшего из них [1, 2].

Способы решения экономических проблем, как правило, основаны на разнообразном компьютерном экономико-математическом моделировании, а средства – на элементах информационных технологий, автоматизирующих расчёты в соответствии с данными способами и методикой. Следовательно, инструментальные методы оптимизации поддержки принятия решений (ППР) существенно опираются на использование систем поддержки принятия решений (СППР) [3].

Необходимость построения СППР возникает в процессе разработки или модернизации системы управления, чтобы сделать ее адекватной требованиям современных динамичных условий хозяйствования для достижения стратегических целей (недаром среди самых успешных компаний, входящих в список Fortune 500, более чем 90% используют СППР).

При этом выявлено что, несмотря на разнообразие процессов интеллектуализации процедур поддержки принятия решений известные методы моделирования данных процессов базируются на методологиях совместной формализации жизненного цикла социально-экономических (СЭ) и компьютерных информационных систем и законах их прогрессивного развития, под которыми понимается качественное изменение данных систем, происходящее в результате разрешения технических и СЭ противоречий.

Обобщенно данный процесс включает в себя следующие этапы: определение и формализация потребностей в интеллектуальных компонентах СППР; оценка и выбор технологии повышения степени интеллектуализации СППР; выполнение пилотного проекта; практическое внедрение средств интеллектуализации СППР. Последнее хорошо вписывается в созданный и развиваемый автором на базе концепции ПИР-требований А.Н. Морозевича подход и метаметодику совместного описания комплекса согласованных требований целевых групп специалистов (например, Потребителей – 79%, Продавцов – 14%, Производителей – 5.2% и Проектировщиков – 1.8%), базирующийся на макетировании, моделировании и поэтапной реализации

средствами современных информационных технологий компонентов и подсистем СППР (ММР-методология). При этом проведен анализ и предложены оригинальные модели оценки качества СППР на базе комплексных показателей качества Q и Q_{DSS} , учитывающих как степень значимости мнений и требований целевых групп, так и степень удовлетворения выдвинутых ими требований [2].

Таблица

Сравнительная характеристика СППР

Название, авторы и оценки	Область применения	Реализуемый способ	Основные функции
MultiExpert (совместно с А.Н. Морозевичем, 1998 г.[3]), $Q=0.7$, $Q_{DSS}=0.8$	Решение в интерактивном режиме дискретных многокритериальных оптимизационных задач в слабоформализуемых проблемных ситуациях	Поддержка принятия групповых решений на основе техники парных сравнений с учетом количества и квалификации экспертов	Система позволяет провести независимый опрос экспертов, занести результаты опроса в базу данных и провести их последующую обработку в зависимости от количества и квалификации экспертов, а также от характера принятых предположений о статистической однородности полученных оценок.
Assistant Choice (совместно с Т.А. Ермаковой, 2000 г.[1]), $Q=0.8$, $Q_{DSS}=0.9$	Многокритериальный выбор оптимальных решений из небольшого числа имеющихся альтернатив в проектах по РБП (управленческий реинжиниринг)	Непосредственное оценивание важности каждого элемента иерархии в соответствии с предложенной вербальной шкалой ранжирования.	Оценка каждой альтернативы по заданным критериям, образующим иерархию. Результат оценки – вектор комплексных индексов для каждой альтернативы (рейтинг альтернатив) с рекомендацией варианта выбора решения.
Study Expert (совместно с О.А. Синявской, 2002 г.[4]), $Q=0.7$, $Q_{DSS}=0.85$	Проведение групповой экспертизы по вопросу выбора наилучшей альтернативы	Поддержка проведения групповых <i>дистанционных</i> экспертиз в асинхронном распределенном режиме с учетом количества и квалификации экспертов	Проведение экспертизы в локальной компьютерной сети с разграничением доступа экспертов и администратора экспертизы. Обработка данных, полученных экспертами, и обобщение результатов экспертизы.

МАИН (совместно с А.А. Ахрамёко, 2004 г. [5]), $Q=0.77$, $Q_{DSS}=0.89$	Выбор наилучшей альтернативы из небольшого числа имеющихся в различных сферах деятельности, построение рейтингов (в том числе интерактивных)	Использование неопределенных данных путем применения нечетких чисел в качестве способа формализации размытых экспертных суждений. (Метод Fuzzy+ – АНР)	Построение моделей принятия решений на основе модификации метода Саати, адаптированного к условиям дефицита информации и нестохастической неопределенности данных, различиям в квалификации экспертов и их субъективным особенностям. Система размещена в сети Internet по адресу kivi.bas-net.by/il .
---	--	--	--

Для экспериментального исследования полученных результатов разработан ряд прототипов СППР на основе оригинальных модификаций метода анализа иерархий Т. Саати. При этом предложен новый класс моделей процессов принятия решений, основанный на метафоре динамических проблемных ситуаций (ДПС-моделей), а также *концепция построения мультиобъектных СППР (МО СППР)*, базирующаяся на введенных обобщениях модели ППР (Generalized Model – GM) и понятия объекта (Generalized Object – GO), а также модифицированных нечетких продукционных правилах. Введенные обобщения положены в основу предложенного мультиобъектного метода моделирования и анализа слабоформализуемых динамических проблемных ситуаций (ДПС-метода), в котором информационно-аналитическая поддержка процесса решения проблемы осуществляется путем организации и проведения асинхронной распределенной *дистанционной* экспертизы.

Данные СППР стали де-факто *стандартом* в экономическом, в том числе дистанционном, образовании РБ (в течение ряда лет постоянно применяются в учебном процессе 6-ти ведущих вузов), опубликованы на сайте http://sedok.narod.ru/s_files/belorussia.htm, в четырех учебных пособиях с грифом министерства образования, использованы при разработке шести учебных программ по дисциплинам, связанным с подготовкой кадров экономического профиля в РБ (в том числе одной типовой и одной базовой). Результаты оценки СППР показывают, что все системы имеют достаточно высокое качество (более 70% требований целевых групп удовлетворено, из которых функциональные требования потребителя – 48.3%), причем данные СППР *обеспечивают* повышение эффективности дистанционной подготовки специалистов за счет улучшения их основных эксплуатационных характеристик (некоторые из которых приведены в таблице) [1–3].

При этом выявлено, что сегодня задачу использования информацион-

ных технологий для оптимизации бизнес-процессов считают приоритетной на треть больше опрошенных менеджеров, чем тех, кто главной целью видит учетно-расчетные операции. Еще три года назад оптимизацию считало приоритетом в 1,7 раза меньше специалистов, чем тех, кто ставил во главу угла учет. Если в 2002 году лишь 7% предприятий использовали автоматизированную систему для решения комплексных задач, то сейчас их доля увеличилась до 27%. Сегодня доля нестандартных задач в деятельности руководителя, по мнению опрошенных, составляет 50%. Тремя годами раньше этот показатель составлял около 30%. Налицо усложнение управленческих задач. При этом сегодня 88% опрошенных считают, что компьютер облегчает труд руководителя – раньше так полагало лишь 70%. И хотя в ряде случаев перемены имеют лишь психологический характер, тем не менее, они чрезвычайно важны. Менеджеры начинают понимать, что инструментальные методы в бизнес-аналитике обеспечивают прозрачный контроль директора за работой служб и позволяют снизить риски принятия решений.

Литература:

1. Реинжиниринг бизнес-процессов: монография / Б.А. Железко, Т.А. Ермакова, Л.П. Володько; под. ред. Железко Б.А. – Мн.: Мисанта, 2004. – 214 с.

2. Zhelezko B.A., Siniavskaya O.A., Ahrameiko A.A., Berbasova N.Y. Methodology of the estimation of quality of objects with complex structure under conditions of non-stochastic uncertainty // International Conference on Fuzzy Sets and Soft Computing in Economics and Finance (FSSCEF 2004): Proceedings. – Saint-Petersburg, 2004. – Vol. 2. – P. 360–367.

3. Железко Б.А., Морозевич А.Н. Информационно-аналитические системы поддержки принятия решений. – Мн.: НИУ, 1999. – 140 с.

4. Информационное обеспечение производства: системы поддержки принятия решений: Методические указания к лабораторным работам для студентов экономических специальностей / Б.А. Железко, О.А. Сиянская. – Минск: БГАТУ, 2005. – 36 с.

5. Компьютерные информационные технологии: экспертные системы и системы поддержки принятия решений: Методические рекомендации по выполнению лабораторных работ / Б.А. Железко, О.А. Сиянская, Л.П. Володько, А.А. Ахрамейко. – Минск: БГЭУ, 2004. – 51 с.

РОЗВИТОК ПЕДАГОГІЧНИХ ЗДІБНОСТЕЙ У НАВЧАННІ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИМ ТЕХНОЛОГІЯМ СТУДЕНТІВ ЕКОНОМІЧНИХ СПЕЦІАЛЬНОСТЕЙ

О.В. Клочко

м. Вінниця, Вінницький державний аграрний університет
klochkoob@rambler.ru

Одним із напрямків вдосконалення підготовки студентів економічних спеціальностей є розвиток педагогічних здібностей спеціаліста, що дає йому можливість перетворити отримані знання, вміння та навички в засіб професійної діяльності, можливості задовольняти свої особистісні інтереси та інтереси суспільства. Підвищення ролі цих здібностей пов'язане з посиленням ролі людського чинника у забезпеченні ефективної діяльності підприємства, що викликає необхідність глибоких знань сучасних педагогічних технологій, володіння методами і прийомами, які є важливим інструментарієм ефективної взаємодії між людьми.

Метою даного дослідження було з'ясувати психолого-педагогічні засади формування у студентів педагогічних здібностей, необхідних фахівцям економічного напрямку. Педагогічні спостереження, бесіди з викладачами і методистами, вивчення, аналіз і узагальнення досвіду викладання інформатики у ВНЗ показали, що важливою складовою підготовки фахівця має бути прищеплення йому певного педагогічного хисту. Традиційні функції економіста, бухгалтера, менеджера (планування, ведення обліку, дослідження ринку збуту, тощо) усе більше доповнюються організаційно-управлінськими і педагогічними функціями, які пов'язані з процесом навчання, перенавчання та виховання персоналу, його мотиваційною зорієнтованістю. Адже, наприклад, в районних та місцевих органах самоврядування економіст буде чи не єдиним авторитетним спеціалістом, який володіє краще за інших колег комп'ютерною технікою, оргтехнікою, вміє застосовувати програмне забезпечення для розв'язування професійних задач. Тому він повинен мати педагогічний хист і такт, необхідні для того, щоб вміти зрозуміло і цікаво навчати колег розв'язувати фахові завдання за допомогою комп'ютерних технологій.

На нашу думку, формування педагогічних здібностей у навчанні інформаційно-комунікаційним технологіям набуває ефективності, якщо:

- педагогічними умовами реалізації системи підготовки фахівців буде гармонійне поєднання таких чинників [3]: раціональний добір змісту, особистість викладача, педагогічні технології, організація освітнього середовища, характер навчально-пізнавальної та соціальної діяльності студента;
- при визначенні змісту навчання інформаційно-комунікаційним технологіям враховуються соціальні вимоги до професійних знань, умінь та навичок, а також особистісних якостей фахівців економічного напрямку;

- воно спрямоване на формування у майбутнього фахівця навичок навчання, виховання і розвитку людей з урахуванням їх індивідуально-особистісних якостей, уміння проектувати і здійснювати навчально-виховний процес у різних умовах, зокрема, на робочому місці.

У рамках системного підходу будь-яке педагогічне явище або процес розглядається як система.

У процесі дослідження професійно-педагогічних функцій готовності студентів економічних спеціальностей вищих аграрних навчальних закладів до професійної діяльності, можна виділити наступні компоненти [1]:

- аналітичну – аналіз колективних і міжособистісних стосунків, прийняття оптимальних рішень і втілення їх у практику;
- виховну – становлення і виховання колективу організації;
- комунікативну – розвиток комунікативних умінь;
- мотиваційну – формування інтересу до роботи і цілеспрямованості у членів колективу;
- навчальну – формування у членів колективу знань, умінь і навичок з використання інформаційно-комунікаційних технологій у професійній діяльності;
- розвиваючу – формування у членів колективу уміння переносити знання та навички у нову ситуацію, інтелектуальних і пізнавальних здібностей, готовності до опанування та використання нової комп'ютерної техніки та нового програмного забезпечення, готовності до подальшої самоосвіти в галузі інформаційних технологій;
- управлінську – організація взаємодії членів колективу, окремих груп.

Такий підхід до визначення суттєвості та змісту професійно-педагогічних функцій готовності студентів економічних спеціальностей вищих аграрних навчальних закладів до професійної діяльності дозволяє розглядати ці характеристики особистості в якості показника рівня професіоналізму фахівців економічного спрямування.

Однією з педагогічних умов реалізації формування педагогічних здібностей є підбір професійно-значущого навчального матеріалу [1]: розв'язування педагогічних творчих завдань із професійним змістом.

З метою розвитку логічного і творчого педагогічного мислення у майбутніх фахівців-економістів, а також для пошуку оптимальних управлінських рішень у їх професійній діяльності доцільно застосовувати такі методи навчання: інтерактивні, евристичні, дослідницькі, пошукові, проблемний метод, метод проєктів, ділові ігри, демонстраційний метод.

Педагогічні здібності спеціаліста-економіста, а саме – планування, підготовка до публічних виступів, вміння навчати, пояснювати свою думку, керувати роботою інших формуються під час застосування відповідних форм навчання у процесі вивчення інформаційно-комунікаційних технологій.

У групових формах навчання діяльність є колективною, в процесі дія-

льності між членами колективу створюються відносини взаємної відповідальності, контроль за діяльністю частково здійснюється самими членами колективу [2]. У парному навчанні студенти пояснюють один одному незрозумілий матеріал, оцінюють результати товариша. У груповій роботі над спільним завданням студенти взаємодіють між собою: пояснюють новий матеріал, обговорюють його, оцінюють свою діяльність, готують виступи, наприклад організація ділової гри при вивченні пакетів прикладних програм “ІС:Бухгалтерія”. У груповій формі – студент замість викладача – один чи двоє студентів навчають всю групу, проводять заняття за комп’ютером, здійснюють допомогу при виконанні практичної роботи.

Колективна пізнавальна діяльність більш емоційна і привчає до прояву ініціативи. Наприклад, розглянемо інтерактивний метод “мозкового штурму”, який можна застосувати під час вивчення теми “Налагодження програми” Більшість практичних суперечливих питань, що виникають у реальному комп’ютерному середовищі, можна розв’язувати застосовуючи метод вирішення проблем у кілька етапів:

1. Аналіз помилки: Що трапилось? Чому? Як її можна виправити? Якої інформації бракує для вирішення проблеми? Де її можна отримати?

2. Пошук шляху виправлення помилки: Які способами можна виправити помилку? В чому полягають їх переваги й недоліки?

3. Вибір способу виправлення помилки: Який спосіб вирішення проблеми є найкращим з точки зору одержання практичного й ефективного результату, простоти? Які труднощі можуть з’явитися при його реалізації? Як наступні кроки під час реалізації способу виправлення помилки?

Завдяки колективній формі організації вивчення інформаційно-комунікаційних технологій студенти навчаються таким видам педагогічної діяльності:

- підготовці виступу перед аудиторією, презентації роботи програми; пошук потрібної інформації в мережі Інтернет;

- колективному обговоренню розв’язування поставленої проблеми; за рахунок використання локальної комп’ютерної мережі на етапах розв’язування професійного завдання формуванню у колег уявлення про реальну ситуацію: від постановки задачі до аналізу здобутих результатів;

- у процесі активного навчання у групах студенти обмінюються інформацією, порівнюють варіанти розв’язування завдань, оцінюють один одного, здійснюють вплив один на одного;

- спілкуванню та співпраці, використовувати метод “мозкового штурму” для колективного обговорення проблеми;

Професійні задачі і функції, що відображені в моделі спеціаліста, отримують своє вираження у власне навчальних задачах, задачах квазіпрофесійних і навчально-професійних. Це і є змістовне розв’язання проблеми інтеграції освіти, виробництва і науки.

Компонентами інформаційної культури є фахова компетентність і мо-

дельне мислення, здатність розв'язувати професійні задачі, тобто компонентою є вміння управляти і навчати підлеглих, формувати, пояснювати і відстоювати свою думку. Важливими рисами фахівця-економіста є вміння представили колегам розроблений план діяльності економічного об'єкта, презентувати новий вид продукції, довести до відома підлеглих обсяг завдань. Вирішальними умовами ефективності такої діяльності є проблемність, демонстрація, наочність, варіювання умов задачі з метою виявлення суттєвих спільних властивостей об'єктів і їх відмінностей тощо. Оскільки, згідно з положеннями асоціативно-рефлекторної теорії, утворення в свідомості індивіда різних систем асоціацій, починаючи від простіших і закінчуючи узагальненими мотивовано забезпечує людину орієнтувальною основою дії або діяльності в цілому [2]. Реалізації таких функцій сприяє вивчення студентами на заняттях з інформатики програмних продуктів (MS Word, PowerPoint, Paint, Adobe Photoshop) за допомогою яких можна організувати наочне представлення інформації, створити презентацію. Важливу роль у вирішенні даних задач відіграє системний підхід до вивчення об'єктів, модельне мислення, які формуються у процесі вивчення дисциплін "Методи оптимізації", "Економетрія", "Дослідження операцій", "Логістика", "Методи обробки економічної інформації та прогнозування", "Проектування макроекономічних систем" із застосуванням завдань з професійно орієнтованим змістом.

Однією з особливостей задач управління колективом є вибір оптимального варіанта прийнятого рішення за найбільш доцільної лінії поведінки. Професійно-педагогічні функції управління колективом отримують своє вираження у навчальних задачах через поєднання добору змісту, педагогічних технологій (евристичний, дослідницький, пошуковий і проблемний методи), програмних засобів. Така організація навчання інформаційним технологіям формує у студентів вміння будувати моделі об'єктів, вміння управляти ними та цілеспрямовано навчати підлеглих виконувати відповідні дії для розв'язування поставлених задач, передавати необхідні знання і стратегії. На даний час створене програмне забезпечення, яке дозволяє знаходити оптимальне рішення, а також робити прогнози. До такого програмного забезпечення належать пакет Microsoft Excel, MathCAD, GRAN1, прикладні пакети статистичної обробки даних STATISTICA, S-PLUS, Mathematica і ін. Фахівцеві необхідно правильно сформулювати задачу, формалізувати її (скласти математичну модель), ввести дані й одержавши результат, проаналізувати його. Якщо отримане оптимальне рішення не задовольняє відповідним умовам, то можна модифікувати створену модель й одержати новий результат.

Висновки

Зрозуміло, що при одному і тому ж змісті навчання, навчальних планах і програмах студенти отримують різний рівень освіти, що залежить від багатьох факторів: індивідуальних особливостей студента, особистості педагога,

типу виконуваної навчальної діяльності, рівня особистої активності студента і його включення в процеси пізнання, способів спілкування з викладачем і іншими студентами, засобів навчання.

Сьогодні важливим є не тільки впровадження інформаційно-комунікаційних технологій на робочому місці, а й творче вміння розв'язувати проблеми управління персоналом, досягнення порозуміння з працівниками, підвищення їх мотивації, навчання, виховання і розвиток членів колективу. Формування педагогічних здібностей спеціаліста суттєво підвищить якість його професійної підготовки. Підготовка фахівця-економіста повинна базуватися не тільки на умінні «працювати за комп'ютером», але і на знаннях, що дозволяють швидко і якісно використовувати досягнення нових інформаційно-комунікаційних технологій у своїй професійній діяльності.

Економічна галузь – складна динамічна система, управління якою є комплексною, задачею для розв'язування якої необхідні знання у багатьох наукових і практичних галузях, урахування намірів і дій інших людей – партнерів, союзників, супротивників, здатність збирати інформацію із багатьох джерел і приймати одночасно декілька рішень в умовах обмеженого часу. Через спеціально організоване навчання інформаційним технологіям у студентів формуються системний тип орієнтування у досліджуваній галузі, вміння досліджувати нові об'єкти, вміння управляти ними та цілеспрямовано навчати підлеглих виконувати відповідні дії для розв'язування поставлених задач, передавати необхідні знання і стратегії, прийоми діяльності. У результаті проведеного дослідження можна зробити висновок, що застосування відповідного змісту, методів, форм і засобів навчання студентів економічних спеціальностей інформаційно-комунікаційним технологіям сприяє розвитку їх педагогічних здібностей, їх професійному становленню на сучасному етапі соціально-економічного розвитку країни.

Література:

1. Капітанець О.М. Педагогічна підготовка менеджерів у вищих технічних навчальних закладах: Автореф. дис. ... канд. пед. наук. – Т.: ТДПУ імені Володимира Гнатюка, 2001. – 20 с.
2. Морзе Н.В. Методика навчання інформатики. Частина 1. Загальні питання методики навчання інформатики. – К.: Навчальна книга, 2003. – 189 с.
3. Фоміна М.В. Структурування змісту психолого-педагогічної підготовки майбутніх інженерів машинобудівного профілю: Автореф. дис. ... канд. пед. наук. – В.: ВДПУ імені Михайла Коцюбинського, 2005. – 20 с.

РОЗВИТОК ТВОРЧИХ ЗДІБНОСТЕЙ СТУДЕНТІВ ЕКОНОМІЧНИХ СПЕЦІАЛЬНОСТЕЙ НА ЗАНЯТТЯХ З ІНФОРМАТИКИ

О.Г. Онишко

м. Хмельницький, Хмельницький національний університет

Практика навчання інформатиці, аналіз літературних джерел засвідчують, що в контексті сучасних реформувань змісту освіти, суть їх полягає у спрямованості на виховання творчої особистості, фахівця, спроможного постійно вдосконалювати свій професійний рівень, продуктивно вирішувати актуальні проблеми, спрямовані на формування фахової компетентності спеціаліста. А тому розвиток творчих здібностей студентів, формування їхнього досвіду творчої продуктивної діяльності, самоосвіти, самовдосконалення є важливим завданням вищої школи.

Щоб сформувати такі якості під час навчання інформатиці, необхідно певним чином підібрати зміст, методи навчання, систему педагогічних впливів. Теоретичною основою роботи є праці психологів Д.Б. Богоявленської, П.Я. Гальперіна, В.В. Давидова, В.М. Дружиніна, А.М. Леонтьєва, Н.А. Менчинської, Я.О. Пономарьова, В.А. Роменця, М.А. Холодної та ін.

Питання розвитку творчих здібностей студентів під час вивчення інформатики, різні аспекти впливу інформаційних технологій на розвиток творчого мислення, пізнавальної активності учнів, студентів відображено в працях Ю.К. Бабанського, М.І. Бурди, І.Я. Лернера, М.І. Махмутова і ін.

У той же час, залишаються недостатньо розглянутими проблеми теоретичного обґрунтування і впровадження технології організації навчально-пізнавальної діяльності студентів, спрямованої на розвиток творчих здібностей. Здібності людини спираються на наявні у неї знання, вміння та навички, на системи тимчасових нервових зв'язків, що є засадовими стосовно них, вони формуються й розвиваються у процесі набування людиною нових знань, умінь і навичок [4].

Проте це не означає, що здібності людини зводяться до її умінь, знань і навичок. Здібності – це такі психологічні особливості людини, від яких залежить оволодіння знаннями, вміннями та навичками, але які самі по собі до знань, умінь і навичок не зводяться. Здібності залежать від знань, але здібності визначають швидкість та якість оволодіння цими знаннями. Щодо знань, умінь і навичок здібності швидше виявляються не в їх наявності, а в динаміці оволодіння ними, тобто в тому, наскільки за однакових умов людина швидко, глибоко, легко і міцно опановує знання та вміння.

Важливу роль у структурі здібностей відіграє здатність людини мислити, розкривати не дані безпосередньо зв'язки та відношення [1]. Важливе значення тут мають такі якості мислення, як широта, глибина, якість, послідовність, самостійність, критичність, гнучкість. Так, В.О Крутецький, досліджуючи здібності школярів до математики, виявив важливу роль таких

компонентів [3]:

- швидко й широко узагальнювати математичний матеріал (узагальнення без спеціального тренування);
- швидко згортати, скорочувати процес міркування при розв'язуванні математичних завдань;
- швидко переключатися з прямого на зворотний хід думки у процесі вивчення математичного матеріалу.

Як і всі інші індивідуально-психологічні особливості, здібності не даються людині в готовому вигляді як щось властиве їй від природи. Здібності кожної людини, її індивідуальні особливості є результатом її розвитку.

Крім того, що особливо є важливим з точки зору розвитку творчих здібностей, студенти отримують навички прогнозування та формулювання гіпотез відносно закономірностей, уміння використовувати різні методи при розв'язуванні задач; вчать аналізувати одержані експериментальні дані, формулювати висновки.

Виконати завдання в інтегрованому середовищі MS Office [2] і представити звіт по практиці, оформлений відповідно до зазначених вимог.

Скласти базу даних, що містить відомості про роботу будівельної фірми, до складу якої входить декілька бригад. Визначити суму, нараховану кожному співробітнику за роботу протягом місяця. При нарахуванні врахувати, що співробітникам, що проробили більш 3-х років установлюється надбавка розмірі 10% заробітної платні, що пропрацювали понад 5 років - у розмірі 30% заробітної платні. Прибутковий податок нараховується за наступними правилами: сума менша за 80 гр. податком на обкладається, на суму в межах від 80 гр. до 300гр. нараховується податок у розмірі 10% від нарахованої суми, при сумі, що перевищує 300 гр. податок складає 20%.

При проектуванні бази вихідними даними вважати: ПІБ співробітника, спеціальність, заробітної платні тощо.

Ми виходимо з того, що розвиток творчих здібностей студентів під час вивчення курсу інформатики технічного вищого навчального закладу основну увагу доцільно приділити задачам, що зводяться до проведення обробки інформації за допомогою комп'ютера. Тоді студентам стануть зрозумілими нові властивості процесу, які можуть з'явитися в результаті обробки інформації, у них формується здатність до оволодіння і застосування методологічних принципів у навчально-творчій і пізнавальній діяльності. Студенти усвідомлюють, як кількісні зміни переходять у якісно нову інформацію, доступнішу та активнішу.

Виконання завдань за допомогою комп'ютера розвиває такі складові творчих здібностей, як стиль мислення, прищеплює певні навички розумових дій, вміння планувати структуру дій, визначати послідовність стандартних операцій для досягнення мети; вміння алгоритмічно мислити; вміння організувати пошук інформації необхідної для розв'язування поставленої задачі.

Ефективним засобом формування творчих здібностей є нові інформаційні технології, особливо слід виділити технології використання табличних процесорів.

Табличні процесори є зручними інструментами для відображення, обробки економічної інформації, корисними для тих фахівців, діяльність яких потребує виконання складних розрахунків, та враховувати наявність великої кількості різноманітних чинників, що впливають на прийняття правильного рішення.

Розглянемо методичні питання розвитку окремих компонент творчих здібностей студентів на прикладі побудови бази даних.

Основною проблемою проектування БД є визначення складу і структури даних предметної області. БД є сукупністю моделей даних, які описують логічну структуру і фізичну організацію даних машинних носіях під керуванням СУБД. Основою для проектування логічної структури БД є формалізований опис даних предметної області. Неправильне розуміння даних і інформаційних потреб додатків предметної області призводить до помилок в структурі БД.

Визначити кількість полів та рядків бази даних для обчислення.

Завдання студенти виконують в такій послідовності:

1. Проектують базу даних. Визначають імена, типи, розміри полів, наводять розрахункові формули для полів, наводять ескіз таблиці.

2. Створюють базу даних.

3. Переглядають створену базу, додають і видаляють деякі записи і поля, виконують коригування окремих даних.

4. Виконують найпростішу математичну обробку числових полів.

Під час виконання цього етапу проектування бази даних студенти розкривають спроможність до оволодіння і застосування методів дослідження (експеримент, моделювання, ідеалізації, формалізації, порівняння математичних моделей тощо) в розв'язуванні навчально-творчих і дослідницьких завдань і задач.

5. Виконують просте і складне сортування у полях бази.

6. Доповнюють підсумовані дані декількох полів. Виконують дії зі структурою.

7. Здійснюють аналіз інформації, що міститься в базі даних. Обчислюють статистичні характеристики числових полів.

8. Вибірково виконують добір даних за допомогою автофільтра і розширеного фільтра за визначеними критеріями.

Студент спроможний до продуктивного опанування фаховою культурою у тому випадку, коли йому надається можливість для стимульованого творчого розвитку, в процесі якого студентом поглиблюється досвід досягнення мети, здійснюється осмислене створення продукту, на основі чого формуються вміння самостійного вибору нових задач і цілей. Інакше кажучи, успішний розвиток творчих здібностей студентів можливий за умови

побудови наявного рефлексованого осмислення завдань.

9. Будують діаграми, графіки за вихідною базою і за результатами вибіркового використання даних.

Таким чином, за результатами проектування БД студенти навчаються створювати таблиці БД, задавати їхню структуру, вибирати типи полів та керувати їхніми властивостями, знайомляться з основними типами зв'язків між таблицями, навчаються створювати запити та інші об'єкти БД. Студенти з'ясовують зміст таких понять, як інформаційно-логічна модель предметної області, концептуальна модель БД, внутрішня модель БД, зовнішня модель БД тощо.

Незаперечною перевагою MS Excel є можливість застосування графічного способу рішення багатьох завдань, який дозволяє отримати результат без математичних розрахунків.

В процесі розробки БД у студентів формуються деякі компоненти творчих здібностей: можливість самостійно продумати етапи розробки БД, зібрати дані, запропонувати методику їхнього опрацювання, реалізувати інформаційну систему на комп'ютері, перевірити на практиці результати своєї роботи і побачити можливість застосування отриманих результатів у майбутній професійній діяльності. Якщо ж ця задача вирішується колективно, то, сприймаючи задачу в цілому, студент повинен чітко уявити свою роль під час її розв'язання. Природно, що вирішення такої проблеми потребує великих витрат часу і не може обмежуватися рамками, наприклад, одного лабораторного заняття.

Стосовно самостійної роботи над розробкою БД, то вона потребує розвинутого методичного забезпечення. Вона є природним продовженням аудиторних занять, які проводить викладач, але тільки в тому випадку, коли студент одержує необхідні засоби роботи як із теоретичним, так і з практичним матеріалом, а також при наявності достатньої кількості часу для самостійної роботи за персональним комп'ютером.

Н.В. Морзе [5] виділяє такі типи самостійних робіт, які відображають сучасні підходи до процесу навчання:

1. Відтворення роботи, яка вимагає від студента головним чином репродуктивних дій.

2. Пошукові (евристичні) роботи, що спонукають студентів до активного усвідомлення матеріалу, пошуку варіативних вирішення навчальної задачі.

3. Творчі роботи, виконання яких потребує інтенсивної самостійної діяльності.

Ми розділяємо думку Н.В. Морзе [5] про необхідність доповнювати методи навчання проектуванню БД спеціальними методами навчання, які використовуються в науці. Постановка проблеми відображення методів науки в навчанні обґрунтовується такими двома аргументами: по-перше, цілі навчання включають засвоєння не тільки певної сукупності наукових фак-

тів, але і методів пошуку цих фактів, які використовуються у самій науці; по-друге, оскільки методи наукових досліджень є одночасно методами придбання нових знань у науці, а методи навчання спрямовані на оволодіння новими знаннями в навчально-пізнавальній діяльності, то природно, що методи навчання використанню баз даних відображають методи пізнання, які застосовуються в інформатиці, зрозуміло, у визначеній, пристосованій для навчання, формі.

Такий підхід до розв'язання задач демонструє переваги комп'ютерної графіки, причому використовуються її ілюстративна та когнітивна функції. Ілюстративна функція надає можливість вирішити задачу навіть студентам із слабкою математичною підготовкою, забезпечує економію часу (що є важливим чинником в навчальній діяльності), дозволяє зосередитися на аналізі отриманого рішення, формулюванні відповідних висновків. Когнітивна функція сприяє розвитку пізнання студента і полягає в тому, що за допомогою графічного зображення він може отримати нове знання – про тенденцію явища, процесу, що досліджується, за допомогою тренда прогнозувати значення, формуванню навичок формулювання гіпотез відносно закономірностей, вчатися аналізувати одержані результати експериментів, робити висновки.

Висновки

Таким чином, процес формування у студентів навичок і умінь самовдосконалення буде більш ефективним, якщо він буде організований, як свідомо, цілеспрямована діяльність по розвитку своєї фахової підготовки і професійно значимих якостей і багато в чому визначатиметься рівнем розвитку відповідної мотивації. Майстерність викладача в процесі розвитку мотивації самовдосконалення студента полягає не тільки в умінні з'ясувати внутрішні спонукальні сили студента і сукупність зовнішніх умов, що сприяють позитивній мотивації, але й у спроможності співвіднести їх найоптимальнішою уявою, виходячи з конкретної ситуації.

Література:

1. Богоявленская Д.Б. Психология творческих способностей: Учеб. пособие для студ. высш. учеб. заведений. – М.: Академия, 2002. – 320 с.
2. Гарнаев А.Ю. Excel, VBA, Internet в экономике и финансах. – СПб.: БХВ-Петербург, 2002. – 816 с.
3. Крутецкий В.А. Психология математических способностей школьников. – М.: Просвещение, 1968. – 431 с.
4. Максименко С.Д. Загальна психологія: Навчальний посібник. – Вид. друге, перероб. та доповн. – К.: Центр навчальної літератури, 2004. – 272 с.
5. Морзе Н.В. Система методичної підготовки майбутніх вчителів інформатики в педагогічних університетах. Автореф. дис. ... д-ра пед. наук. – К.: НПУ імені М.П.Драгоманова, 2003. – 39 с.
6. Педагогічні інновації у сучасній школі. – К.: Освіта, 1994. – 88 с.

З ДОСВІДУ ВИКОРИСТАННЯ МЕТОДУ ПРОЕКТІВ У ПРОЦЕСІ НАВЧАННЯ ІНФОРМАТИКИ СТУДЕНТІВ ЕКОНОМІЧНИХ СПЕЦІАЛЬНОСТЕЙ

О.Ф. Клименко^а, Ю.М. Красюк^б

м. Київ, Київський національний економічний університет

імені Вадима Гетьмана

^а OFKlim@online.com.ua

^б Krasyyuk_y@rambler.ru

Потреба “поновлювати” власні знання, набувати нових умінь та навичок властива кожній культурній людині. Однак у сьогоденнішньому інформаційному суспільстві, коли в усіх сферах людської діяльності відбувається стрімке зростання обсягу наукової та технічної інформації, її швидке “моральне” старіння, дана потреба набуває особливого значення, вимагаючи від сучасного фахівця збільшення питомої ваги самостійної діяльності дослідницького характеру.

Підготовка такого спеціаліста, розпочата у школі, гімназії, ліцеї і далі продовжена у вищому навчальному закладі, невід’ємно пов’язана із завданням ефективної організації самостійної роботи студентів у позаурочний час.

Ставлячи перед студентами завдання самостійно засвоїти певний навчальний матеріал, оволодіти відповідними вміннями та навичками, викладач перш за все повинен створити сприятливі умови для успішного його розв’язування. При цьому система планування самостійної поза аудиторної роботи студентів повинна відзначатися:

- методично обґрунтованим добором навчального матеріалу, що пропонується студентам для самостійного опрацювання. Самостійна пізнавальна діяльність студентів повинна проходити на відповідному їм рівні складності в зоні між ЗАР (зона актуального розвитку) і ЗНР (зона найближчого розвитку);

- ретельною діагностикою чинників, що впливають на якість результатів самостійної пізнавальної діяльності студентів, прогнозуванням її ходу та наслідків;

- індивідуалізацією навчального впливу на основі діагностики реальних можливостей студентів окремих індивідуально-типологічних груп, проектуванням конкретних зрушень;

- педагогічно-доцільним та комплексним поєднанням традиційних методик навчання з сучасними інформаційно-комунікаційними технологіями;

- обґрунтованим вибором терміну часу, який необхідний для повноцінного опрацювання навчального матеріалу;

- створенням сприятливих умов для роботи студентів у навчальних лабораторіях вузу (якщо в цьому є необхідність);

- ретельно спланованою системою контролю засвоєних студентами

знань, набутих умінь та навичок.

Окремої уваги заслуговує добір методів навчання, що використовуються під час організації самостійної роботи студентів у позаурочний час. Зокрема, пропонуючи студентам для розв'язування творчі завдання, доцільно застосовувати метод проектів, в основі якого лежить розвиток пізнавальних навичок студентів, умінь самостійно конструювати свої знання, орієнтуватися в інформаційному просторі, розвиток критичного мислення.

Основною тезою методу проектів є: «Все, що я пізнаю, я знаю, для чого це мені потрібно, де і як я можу ці знання застосувати».

Зорієнтований на самостійну діяльність студентів (індивідуальну, парну, групову, яку студенти виконують протягом певного проміжку часу), метод проектів завжди припускає розв'язування деякої проблеми, яка передбачає, з одного боку, використання різноманітних методів, засобів навчання, а з іншої, інтегрування знань, умінь з різних галузей науки, техніки, технології, творчих галузей.

Основні вимоги до використання методу проектів [1, 78-79]:

- наявність значущої в дослідницькому, творчому плані проблеми чи задачі, яка вимагає інтегрованого знання, дослідницького пошуку для її розв'язування, наприклад, проблема використання наявного програмного забезпечення до розв'язування економічних задач та ін.;

- проект розробляється за ініціативою студентів. Тема проекту для всіх студентів може бути однією, а шляхи її реалізації в кожній мікро-групі студентів різні;

- практична, теоретична, пізнавальна значущість передбачуваних результатів (наприклад, рекомендації щодо впровадження та використання конкретних програмних засобів до вирішення поставленої задачі, доповідь у відповідні служби тощо);

- самостійна (індивідуальна, парна, групова) діяльність студентів;

- визначення кінцевої мети спільних / індивідуальних проектів;

- визначення базових знань з різних галузей, необхідних для роботи над проектом;

- структурування змістової частини проекту (з вказуванням поетапних результатів). Проект заздалегідь спланований, сконструйований, але разом з тим допускає гнучкість і зміни в ході виконання;

- використання дослідницьких методів: визначення проблеми, задач дослідження, які впливають із проблем, висування гіпотез щодо способів їх розв'язування, обговорення методів дослідження, оформлення кінцевих результатів, аналіз даних, підведення підсумків, коригування, висновки (використання в ході спільного дослідження методу “мозкового штурму”, “круглого столу”, статистичних методів опрацювання результатів експериментів, творчих звітів та ін.);

- результати виконаних проектів повинні бути матеріальними, тобто оформлені певним чином (відеофільм, презентація, веб-сайт тощо);

– проект реалістичний, орієнтований на ресурси університету.

Розглянемо можливості використання методу проектів у процесі навчання інформатики студентів економічних спеціальностей на конкретному прикладі.

НАЗВА ПРОЕКТУ: “Розрахунок вартості квартири”.

СТИСЛИЙ ОПИС. Студенти розраховують вартість квартир.

ПОВНИЙ ОПИС ПРОЕКТУ. Потрібно проаналізувати ринок нерухомості міста Києва та визначити основні параметри, від яких залежить вартість квартир в столиці. Провівши аналіз вартості квартир в районі станції метро Шулявська (не менше 100 квартир), розрахувати вартість двокімнатної квартири в цьому районі, враховуючи її наступні характеристики: загальна площа – 68 м², корисна площа 38 м², 9-ти поверховий панельний будинок (рік побудови 1996), квартира розташована на 7 поверсі, кухня – 10 м², лоджія – 5,2 м², телефон відсутній. Порівняти вартість цієї квартири з вартістю аналогічних квартир, що розташовані на першому та останньому поверхах. Створити презентацію отриманих результатів Вашого дослідження.

Лозунг проекту:

*“Думай, перш ніж вкладати гроші,
і не забувай думати, коли вже вклав їх”.*

Ф. Дойл

УЧАСНИКИ ПРОЕКТУ: студенти певної студентської групи.

ТЕРМІНИ РЕАЛІЗАЦІЇ ПРОЕКТУ: два тижні після отримання студентами завдання.

ЕТАПИ ПРОВЕДЕННЯ ПРОЕКТУ:

Цілі проекту:

- продовження формування у студентів навичок використовувати ресурси Інтернет для пошуку потрібних відомостей;
- продовження формування у студентів умінь виділяти суттєві ознаки;
- продовження формування у студентів навичок застосовувати методи лінійної та експоненціальної регресії до розв’язування економічних задач;
- продовження формування у студентів навичок використовувати статистичні функції MS Excel для розв’язування задач прогнозування;
- заохочення студентів до використання PowerPoint для створення мультимедійних презентацій;
- розвиток у студентів умінь аргументовано оцінювати отримані результати проекту.

РЕАЛІЗАЦІЯ ПРОЕКТУ ПЕРЕДБАЧАЄ:

Заняття №1.

1. Мотивація самостійної пізнавальної діяльності студентів та представлення проекту.

Студентам після проведення практичного заняття за темою “Використання статистичних функцій MS Excel для розв’язування задач прогнозування” пропонується до розгляду наступна життєва ситуація:

“Сім’я хоче продати власну двокімнатну квартиру, що розташована в районі станції метро Шулявська та має наступні характеристики: загальна площа – 68 м², жила площа 38 м², 9-ти поверховий панельний будинок (рік побудови 1996), розташована на 7 поверсі, кухня – 10 м², лоджія – 5,2 м², телефон відсутній. Запросивши брокера для оцінки вартості своєї квартири, вони залишилися невдоволені запропованою ціною. Як можна самостійно перевірити правильність оцінки брокера?”

Викладач оголошує назву проекту, його завдання та етапи, особливості оцінювання результатів виконання проекту.

(Час – 5 хвилин).

2. Формування малих груп.

Викладач в середині студентської групи формує 5 малих груп (кожна група складається з 4–5 чоловік), використовуючи примусовий метод (формує малу групу, потрібно враховувати рівень підготовки студентів та їх психологічну сумісність).

Серед сформованих малих груп окрему групу складають експерти. Для кожної робочої групи визначається студент-експерт, який буде, при необхідності, надавати допомогу студентам цієї робочої групи, а також оцінювати їх роботу.

Студенти отримують роздатковий матеріал.

(Час – 10 хвилин).

3. Висування гіпотез розв’язання визначеної проблеми.

Застосовуючи метод мозкового штурму, студентам всієї групи спочатку пропонується взяти участь у генеруванні пропозицій з метою:

- визначення основних характеристик квартири, які впливають на її вартість;
- формування системи завдань, що дозволить вирішити поставлену перед групою навчальну задачу.

При цьому доцільно дотримуватися наступних правил [2, 61]:

- кожен студент має можливість вільно висловлювати свої пропозиції;
- учасники висловлюються по черзі, чітко та стисло;
- будь-які пропозиції приймаються та схвалюються учасниками;
- експерти повинні записувати всі висунуті пропозиції та можуть брати участь у генеруванні пропозицій;
- не можна критикувати та коментувати пропозиції;
- можна розвивати попередні ідеї.

(Час – 15 хвилин).

4. Обговорення висунутих гіпотез.

Другий етап мозкового штурму (обговорення, класифікація, добір перспективних пропозицій) передбачає роботу у малих групах, з використанням різних форм дискусії та прийомів обговорення проблем. Студенти кожної робочої групи повинні:

- провести ранжування запропонованих характеристик квартир та ді-

брати ті, що будуть враховуватися під час визначення її вартості;
– вибудувати ієрархічну структуру завдань, що повинні бути розв'язані для досягнення шуканого результату;
– розподілити сформовані завдання між студентами робочої групи;
– визначити засоби, які будуть використовуватися під час розв'язування визначених завдань.

Студенти-експерти спостерігають за процесом обговорення в робочих групах.

(Час – 20 хвилин).

По́зааудиторна робота студентів.

5. Робота у малих групах.

Студенти-експерти визначають методи перевірки прийнятих гіпотез та формують систему оцінювання результатів роботи студентів в робочих групах. Дані результати вони повинні презентувати викладачу через тиждень від початку проекту та обговорити з ним результати їх роботи.

(Час – протягом першого тижня).

У робочих групах: реалізуються поставлені завдання; відбувається оформлення результатів. Якщо робоча група потребує допомоги, то її можуть надати експерти.

За 2 дні до проведення презентації кожна група повинна передати експертам отримані результати свого дослідження для того, щоб експерти могли з ними детально ознайомитися та оцінити змістовну частину.

(Час – протягом двох тижнів).

Заняття №2.

Захист проектів.

Кожній робочій групі пропонуються по 5 хвилин для презентації власних результатів проекту (при цьому кожний студент повинен бути готовий захищати отримані результати проекту).

(Час – 25 хв.)

В кінці виступу представників усіх робочих груп слово отримують експерти для аналізу проведених презентацій, підведенню підсумків та оцінювання результатів роботи.

(Час – 10-15 хв.)

Підсумкове слово викладача.

Підведення підсумку проекту та оцінювання роботи експертів.

(Час – 10 хв.)

ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ ВИКОНАННЯ ПРОЕКТУ.

Максимальна кількість балів, яку може отримати кожний студент за виконання проекту, за умови, що він якісно справився з поставленими перед ним завданнями – 10 балів (у відповідності до особливостей рейтингової системи оцінювання результатів навчальної діяльності студентів при навчанні інформатики в КНЕУ). Бали виставляються експертом, який працював з студентами даної робочої групи.

За кращий проект робоча група може отримати бонусні 10 балів.

Як показує досвід, використання методу проектів у процесі навчання інформатики студентів економічних спеціальностей спрямовує студентів на розвиток власних знань, персональних та фахових навичок, у тому числі навичок колективної роботи та творчого вирішення конкретних проблем.

Література:

1. Морзе Н.В. Методика навчання інформатики: Навч. посібник: В 4 ч. / За ред. М.І. Жалдака. – К.: Навчальна книга, 2003. – Ч. 1: Загальна методика навчання інформатики. – 254 с.

2. Ігри дорослих. Інтерактивні методи навчання / Упоряд. Л. Галіцина. – К.: Ред. Загальнопед. газ., 2005. – 128 с. (Б-ка “Шк. світу”).

ТЕХНОЛОГІЯ РОЗВ'ЯЗУВАННЯ ТВОРЧИХ ФІНАНСОВО-ЕКОНОМІЧНИХ ЗАДАЧ НА ЗАНЯТТЯХ З ІНФОРМАТИКИ

О.Г. Смілянець

м. Вінниця, Вінницький торговельно-економічний інститут
Київського національного торговельно-економічного університету
Smilynez2002@ukr.net

Для підготовки конкурентноспроможних фахівців фінансово-економічної сфери діяльності, які б вміли творчо діяти в сучасних умовах, педагогам вищих навчальних закладів необхідно впроваджувати нові педагогічні технології навчання. У Вінницькому торговельно-економічному інституті КНТЕУ з ціллю вдосконалення навчального процесу виконується держбюджетна науково-дослідна робота з теми “Розробка методики використання творчих фахових задач із застосуванням інформаційних технологій при підготовці спеціалістів фінансово-економічного профілю”. В рамках даної теми розглядається питання технології розв’язування студентами творчих фінансово-економічних задач на заняттях з інформатики засобами комп’ютерної техніки.

Г.С. Костюк зазначав, що розуміння завжди здійснюється на основі попереднього досвіду людини [4, 364]. Про важливість досвіду у мислительній діяльності писав ще І.П. Павлов. Він вказував, що розв’язання будь-якої задачі здійснюється на основі старих засобів, які є в досвіді людини [7].

Рішення задач поліпшується, якщо вироблення відповідних умінь було належно вмотивованим. Цю закономірність підтверджує дослідження П.К. Анохіна, де встановлено, що на стадії інформаційної підготовки рішення “мотивація розпочинає процес вибору із пам’яті, вибору із минулого досвіду всього того, що було в житті людини пов’язане із задоволенням цієї мотивації” [1, 9].

У [5] показано, що існує залежність мислительної діяльності студентів при розв’язанні задач від формулювання і наочного оформлення умови задачі (рисунок, таблиця, граф-схема, діаграма тощо).

У [2] зазначено, що успішне розв’язування задачі економічного змісту пов’язане з умінням переосмислювати допоміжні моделі, підзадачі, включати значення величин (основних і допоміжних) у нові зв’язки. При цьому істотний момент мислення полягає в тому, що, включаючи об’єкт у нові зв’язки, ми усвідомлюємо його в нових якостях, властивостях і, тим самим, одержуємо нові знання про об’єкт. Особливо важливо при розв’язанні цих задач вміти правильно використовувати умову як критерій вибору потрібних зв’язків між відношеннями значень величин. Для цього аналіз умови доцільно здійснювати в трьох формах: 1) аналіз основних і допоміжних величин; 2) аналіз способів їх завдання; 3) аналіз зв’язків значень величин. Проведення такого аналізу передбачає знання змісту економічних понять.

Встановлено, що вміння розв'язувати задачі формуються ефективніше, якщо забезпечуються взаємопереходи від логічних зв'язків між значеннями величин до наочних просторових конструкцій і, навпаки, в результаті яких не лише встановлюються нові зв'язки між поняттями і способами діяльності, а й здійснюється їх синтез – утворюються нові поняття і способи діяльності. Крім того, такі переходи лежать в основі інтуїції – пізнавального акту, в результаті якого з'являється нове знання [3]. Використання комп'ютерних програм під час розв'язування задач допомагає не тільки швидко і правильно виконувати обчислення, але й генерувати певні ідеї стосовно методів розв'язування задач, перевіряти правильність отриманих результатів, обирати більш раціональні шляхи розв'язування. Акцентування уваги студентів на окремих прийомах розумової діяльності під час розв'язування задач за допомогою комп'ютерних програм створює основу для подальшого самостійного оперування системою знань та навичок у нестандартних нових ситуаціях.

У ході дослідження нами використовувалися фахові фінансово-економічні задачі, які розв'язувалися студентами за допомогою комп'ютерної техніки та були до того ж і творчими задачами. Для розв'язання таких задач повинні об'єднуватися технології розв'язування творчих, фахових задач з технологією розв'язання задач за допомогою персональних комп'ютерів. В основу розробки технології розв'язування творчих фахових задач засобами інформаційних технологій лягли технології:

- розв'язання творчих завдань [6, 120];
- розв'язання економічних задач [2, 28-29];
- розв'язання задач на ЕОМ [8, 138-142].

При підготовці з курсів “Інформатика та комп'ютерна техніка” та “Комп'ютерний практикум” студентам пропонувалась наступна технологій розв'язування творчих фахових задач:

1) проаналізувати умову задачі: визначити вхідні дані, з'ясувати, що потрібно знайти, які розрахунки необхідно виконати;

2) сформулювати ціль рішення задачі; врахувати, як це рішення буде використовуватися, як пов'язана дана задача з іншими задачами;

3) з'ясувати, які ще додаткові дані необхідні для розв'язування задачі, визначити засіб знаходження відповідних величин (додаткова література, консультації з фахівцями, Інтернет тощо);

4) спланувати послідовність операцій, спрямованих на розв'язання задачі (алгоритм розв'язання). До даної послідовності дій слід віднести:

– визначення форми представлення вхідних та вихідних даних задачі, форма представлення рішення задачі;

– визначення, у разі потреби, математичної моделі для опису задачі;

– визначення програмних засоби для рішення задачі;

– розбиття задачі на підзадачі, встановлення взаємозв'язку підзадач між собою та з головною задачею;

– визначення зовнішнього виду документа, який необхідно створити у ході рішення задачі, подальшого призначення цього документу, можливі шляхи редагування створеного документу;

5) реалізувати запланований шлях розв'язку;

6) перевірити розв'язок задачі.

Така організація творчої навчальної діяльності студентів дає змогу кожному працювати в оптимальному темпі, не відчуваючи своєї “інтелектуальної неповноцінності”, поступово навчатися узагальнених прийомів розумової діяльності, методів розв'язання широкого кола фахових творчих задач.

Слід зазначити, що такий розгляд етапів технології рішення творчих фахових задач відображає її структуру лише схематично. Жорсткого чергування етапів не існує. Було б помилкою вважати, що кожний наступний етап починається лише по закінченню попереднього. Кожен із етапів – необхідний і цілісний компонент творчого процесу, але вони постійно проникають один в одного, так що на стадії доробки можуть бути внесені корективи в початковий задум.

В [6, 120] зауважено, що одне й те саме завдання може стимулювати розвиток творчих можливостей одних студентів і гальмувати в інших. Тобто для одних це завдання – творче, а для інших – навчальне. Це залежить від рівня розвитку студентів. Майстерність викладача і полягає в тому, щоб усіх студентів залучити до розв'язання різноманітних творчих фахових задач.

При підготовці майбутніх фахівців до розв'язування творчих фахових задач засобами інформаційних технологій перш за все необхідно намагатися розширювати знання студентів з фахових питань та навичок використовувати інформаційні та комп'ютерні технології за фахом. Оскільки в рамках нашого дослідження ми обмежилися дисциплінами “Інформатика та комп'ютерна техніка” та “Комп'ютерний практикум”, то використання творчих фахових задач вимагає розширення у студентів як фінансово-економічних знань, так і можливостей комп'ютерних технологій. Фахові дисципліни найчастіше вивчаються на старших курсах, тому більшість понять та алгоритмів розрахунків фінансово-економічних показників, які використовуються у фахових задачах, студенти можуть або самостійно узнавати, що є видом творчої роботи, або викладач надає необхідну інформацію. В цьому випадку для організації творчої роботи студентам пропонується самостійно визначити алгоритм застосування інформаційних технологій для вирішення фінансово-економічної задачі.

Якщо молодій людині для розв'язання фахової задачі необхідно отримати додаткові знання, це спонукає її звертатися до додаткових інформаційних джерел, або консультуватися з фахівцями чи викладачами з фахових дисциплін. Самостійна робота з інформаційними джерелами сприяє розвитку пошукових і дослідницьких навичок.

В процесі підготовки студентів до розв'язання творчих фахових задач велике значення має виховання критичного ставлення до отриманої інформації. Творчі задачі з надлишковою інформацією примушують студентів критично оцінювати та вибирати необхідну для подальшої обробки інформацію.

Одним із способів підготовки майбутніх спеціалістів до розв'язування творчих фахових задач є обговорення рішення задач у групі студентів. Особливо такий спосіб корисний, коли задачу розв'язали не всі студенти групи, або задача має кілька вірних рішень. Обговорення можна проводити в кінці заняття, або в кінці окремого циклу занять, коли вивчена певна тема.

В процесі підготовки студентів до розв'язування задач велике значення мають позитивні емоції, які можуть викликатися у студентів під час розв'язування естетично привабливих завдань або в процесі естетичного оформлення рішень задач, при експериментуванні та імпровізації з елементами, описаними в умові задачі. Позитивні емоції виникають, коли в процесі рішення задач створюються доброзичливі відношення між студентами групи та між викладачем і студентами. Свобода вибору методів рішення задачі та свобода вибору комп'ютерних програм для рішення задач також надає студентам позитивних емоцій.

Використання засобів нових інформаційних технологій дозволяє полегшити етап розв'язування задачі в середині моделі за рахунок виконання громіздких обчислень. На основі графічних можливостей комп'ютер унаочнює функціональні або статистичні залежності між розглядуваними величинами, що допомагає усвідомити зв'язки між відповідними явищами.

Методика підготовки студентів фінансово-економічних спеціальностей до розв'язування творчих фахових задач засобами інформаційних технологій має враховувати дидактичні цілі, організаційні форми, загальні методи і психолого-методичні закономірності педагогічного процесу вищої школи. Формування вмінь розв'язувати задачі залежить від усвідомлення задачі, мотивів діяльності, розуміння способів діяльності та установки (спрямованості) на повноту, точність діяльності.

Таким чином, ознайомлення майбутніх фахівців фінансово-економічного профілю з технологією розв'язування творчих фахових задач та використання даної технології сприятим ефективному засвоєнню фахових та інформаційних знань, розвитку творчих і дослідницьких здібностей у молоді.

Література:

1. Анохин П.К. Проблема принятия решения в психологии и физиологии // Проблемы принятия решения: Сб. / Отв. ред. П.К. Анохин, В.Ф. Рубахин. – М.: Наука, 1976. – С. 7-11.
2. Дудка Г.Я. Формування вмінь студентів розв'язувати прикладні задачі при навчанні математики і коледжі економічного профілю: Дис. ...

канд. пед. наук: 13.00.02 / Институт педагогіки АПН України. – К., 1998. – 187 с.

3. Кармин А.С. Интуиция и ее механизмы / Проблемы методологии науки и научного творчества / Под ред. В.А. Штоффа, А.М. Мостепаненко. – Л.: Изд-во Ленинградского университета, 1977. – С. 44-64.

4. Костюк Г.С. Вопросы психологии мышления / Психологическая наука в СССР / Под ред. Б.Г. Ананьева, Г.С. Костюка, А.Н. Леонтьева и др. – М.: Изд-во АПН РСФСР, 1959. – Т. 1. – С. 357-440.

5. Мансуров Н. С. Зависимость решения от формулировки и наглядного оформления задач / Процесс мышления и закономерности анализа, синтеза и обобщения: Сб. / Под общ. ред. С.Л. Рубинштейна. – М.: Изд-во АПН СССР, 1960. – С. 153-167.

6. Освітні технології: Навч.-метод. посіб. / О.М. Пехота, А.З. Кіктенко, О.М. Любарська та ін.; За заг. ред. О.М.Пехоти. – К.: А.С.К., 2001. – 256 с.

7. Павловские среды. – М.-Л.: Изд-во АН СССР, 1949. – Т. 1. – 490 с.

8. Экономическая информатика и вычислительная техника: Учебник / Г.А. Титоренко, Н.Г. Черняк, Л.В. Еремин и др.; Под ред. В.П. Косарева, А.Ю. Королева. – Изд. 2-е, перераб. и доп. – М.: Финансы и статистика, 1996. – 336 с. – С. 138-142.

ДИСЦИПЛІНА «ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ» ЯК ПОДОЛАННЯ РОЗРИВУ МІЖ НАВЧАННЯМ ТА РЕАЛІЯМИ ЖИТТЯ

О.М. Боско, Н.В. Гринь
м. Кривий Ріг, Інститут ділового адміністрування

Сучасний етап розвитку економіки України вимагає наявності на ринку праці спеціалістів, які б професійно володіли навичками роботи в інформаційних системах, призначених для автоматизації конкретної предметної області. Особливі вимоги останнім часом пред'являють до фахівців різних рівнів в галузі економіки, зокрема бухгалтерів та фінансистів, які повинні досконало володіти теоретичною базою свого фаху і при цьому мати навички роботи з комп'ютерним забезпеченням, поширеним на даному етапі технічного забезпечення в окремому регіоні і в Україні в цілому [1].

Нами багаторазово підкреслювалась роль такої дисципліни, як «Прикладне програмне забезпечення», яка має подолати прірву між практичним застосуванням та теоретичним навчанням. Її унікальність полягає в можливості швидкого перетворення змісту лабораторних робіт стосовно практичних потреб програмного забезпечення фахівців різних напрямів народного господарства. Настала пора вести розмову про міждисциплінарні зв'язки цієї дисципліни з іншими.

В нашому навчальному закладі студенти фаху «Облік та аудит» та «Фінанси» вивчають декілька дисциплін, пов'язаних з навчанням на комп'ютері. Це такі дисципліни, як «Інформатика та комп'ютерна техніка» (I-III семестр), «Прикладне програмне забезпечення» (V-VII семестр), «Інформаційні системи в обліку та аудиті», «Інформаційні системи в фінансово-кредитних установах» (VII семестр) та «Організація інформаційних систем» для студентів фаху «Облік та аудит» (IX семестр). Отже, як ми бачимо, наші студенти мають можливість досить глибоко здобути та закріпити навички роботи на комп'ютері. Звертаємо увагу на те, що дисципліна «Прикладне програмне забезпечення» викладається саме в той час, коли студенти розпочинають фахове навчання з дисциплін «Облік та аудит», «Економіка підприємства». А це означає, що саме цей етап є максимально сприятливим для підготовки студентів, як майбутніх користувачів АРМів за фаховим напрямом.

Впродовж декількох років ми розглядали різні схеми комплектів лабораторних робіт саме на основі тих дисциплін, які вивчають студенти за фахом і які є теоретичною основою їхньої майбутньої професії. Найбільш вдалою схемою на даний момент, на наш погляд, є показана на рис. 1.

Розгалуженість схеми виникає саме в той момент, коли студенти фаху «Фінанси» виходять на останній етап вивчення комп'ютерних дисциплін. Прикладом гнучкості дисципліни «Прикладне програмне забезпечення» є

розроблена нами методика вивчення програми “БЕСТ-ЗВІТ” для студентів спеціальності “Облік та аудит”. Актуальність її в наш час визначається підтримкою таких державних установ, як податкова інспекція, де згідно з останніми вимогами необхідно кожен звітну форму оформити як у друкованій формі, так і в електронному вигляді в базовому форматі, запропонованому “БЕСТ-ЗВІТ ПЛЮС”. Детальне вивчення програми було покладене в основу лабораторних завдань, які разом утворюють розв’язання комплексної задачі. Студенти наочно розглядають основні форми звітності по підприємствам, які до цього розглядалися лише в теоретичному вигляді в інших дисциплінах. Це дає змогу майбутнім фахівцям поглибити практичні навички роботи.

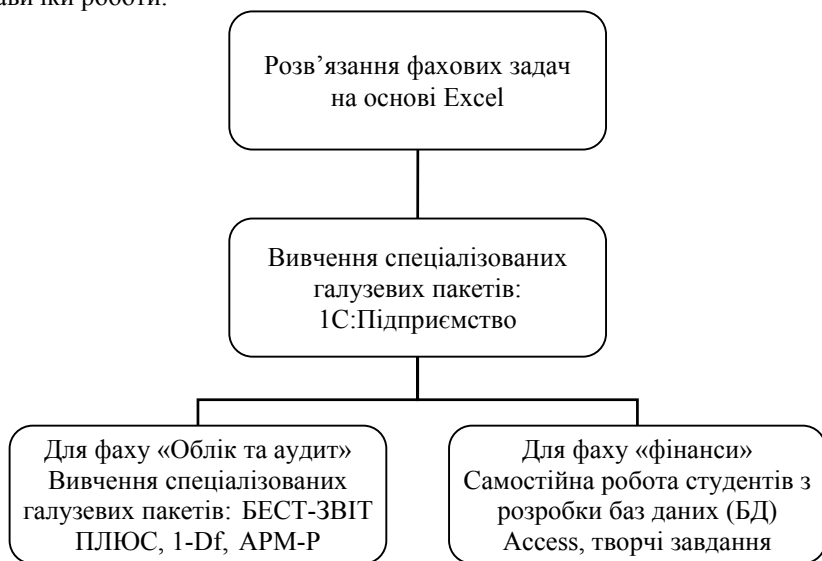


Рис. 1.

В третьому семестрі “фінансисти” мають можливість детально ознайомитися з базами даних на прикладі СУБД MS Access. Методика викладання відрізнялась від загальноприйнятої тим, що, особлива увага, насамперед, була звернута саме на прояв індивідуальної творчості студентів.

Отже, реалізація міждисциплінарних зв’язків дисципліни «Прикладне програмне забезпечення» очевидна. Вона підтверджує необхідність гнучкого підходу до розробки лабораторних робіт з метою якісної підготовки майбутнього фахівця. Доводиться постійно поновлювати склад задач на основі змін чинного законодавства, щоб навчати студентів в умовах, максимально наближених до реальності. Адже завдяки такому підходу розрив між теоретичним навчанням та реаліями повсякденного життя суттєво зменшується.

Це підтверджується результатами щорічного анкетування студентів наприкінці вивчення дисципліни у групах студентів четвертого курсу спеціальності «Облік та аудит» і «Фінанси». Відповіді на питанні анкети свідчать про те, що загалом студенти задоволені підбором практичних програм і вважають їх необхідними для своєї подальшої роботи. Анкетування проводилось на основі анонімності, лише треба було вказати свою групу для визначення року навчання, але масове виказання своїх прізвищ та імен також свідчать про зацікавленість та позитивне ставлення студентів до вивчення цієї дисципліни.

Такі заходи значно поліпшують умови викладання «комп'ютерних» дисциплін, запобігають схоластичному вивченню лекційного матеріалу, надають динамізму практичним заняттям, що є виправданим з огляду на студентський склад, в якому панує молодь до двадцяти трьох років.

Література:

1. Твердохліб М.Г. Інформаційне забезпечення менеджменту: Навч. посібник. – К.: КНЕУ, 2000. – 208 с.
2. Кавторев С. Бухгалтерський облік за допомогою Excel. Повний практичний посібник для сучасного бухгалтера. – Х.: Фактор, 2005. – 206 с.

СОЗДАНИЕ МЕТОДИЧЕСКИХ МАТЕРИАЛОВ ПО TELELOGIC TAU G2 С ПОМОЩЬЮ MICROSOFT PRODUCER

А.В. Бабич

г. Полтава, Полтавский государственный педагогический университет

им. В.Г. Короленко

alexander.v.babich@acm.org

Не так давно в Полтавском политехническом колледже НТУ «ХПИ»[1] появилась специальность 5.080405 «Программирование для ЭВТ и АС», что повлекло за собой введение ряда спецкурсов. Один из них – «Технологии разработки ПО» – практически полностью соответствует курсу SE201 Introduction to Software Engineering стандарта ACM/IEEE Computing Curriculum Software Engineering 2004 [2] и «Проектирование автоматизированных информационных систем», основанный на материалах раздела «Управление информацией (ИМ)» образовательного стандарта Computing Curricula 2001: Computer Science [3]. Первый из упомянутых курсов преподается по материалам курса SE201, созданного коллективом преподавателей в ходе пилотного образовательного проекта «Виртуоз» [4], который проводился в августе-декабре 2004г. в Нижегородском государственном университете (материалы курса распространяются на бесплатной основе и могут быть свободно загружены с сайта IT Lab ННГУ [5]). Материалы же второго из упомянутых курсов опубликованы в MSDN Academic Alliance Repository [6], откуда также могут быть свободно получены.

Обе этих дисциплины связаны тем, что в рамках «Технологий разработки ПО» достаточно глубоко изучается UML [7], а «Проектирование АИС» служит, по сути, полигоном для его применения. UML на данный момент является одной из самых «модных» технологий в области программной инженерии и позволяет системным архитекторам представлять свое видение системы в виде набора стандартных диаграмм, которые, к тому же, служат отличным средством коммуникации в команде разработчиков и прекрасным помощником в общении с заказчиком. И при всем этом, UML – это достаточно логичная и простая для изучения нотация, навыками использования которой, без сомнения, должен овладеть любой специалист, собирающийся работать в области программной инженерии.

Подготовка лекций по UML – не проблема, тем более что, как уже было сказано выше, имеются материалы курса SE201, да и «бумажной» литературы по UML можно найти немало. В качестве же практического метода обучения UML-проектированию мы используем P-modeling [8] – авторскую методику В.Л. Павлова, которая уже доказала свою эффективность в различных вузах и IT-компаниях России и Украины. Но ведь недостаточно учить UML сам по себе – нужно учить студентов использовать прикладные средства UML-моделирования, которые им несомненно придется использо-

вать при реальном проектировании реальных программных систем.

Первая проблема, которая перед нами возникла – выбор такого инструмента моделирования. На данный момент существует огромное количество CASE-средств, поддерживающих UML [9]. Нами были перепробованы многие из них – Borland Together, Poseidon, Visio, Dia, SmartDraw, но ни один из этих весьма достойных программных продуктов по разным причинам нам не подошел. Долгое время нашим фаворитом был продукт под названием Gentleware Poseidon [10]. Наряду с неоспоримыми достоинствами, этот продукт имеет также несколько весьма существенных недостатков, так что, используя Poseidon, мы не прекращали поиски подходящего инструмента. В данный же момент мы осваиваем продукт TAU G2 компании Telelogic [11]. Это средство моделирования сочетает в себе мощь и простоту использования, а также предоставляет уникальную возможность начальной верификации создаваемых моделей. Интерфейс программы удобен и интуитивно понятен для студентов. TAU позволяет создавать все виды диаграмм UML 2.0, проверять их корректность и синтаксическую правильность, экспортировать и печатать диаграммы и многое другое.

Каким бы интуитивно понятным не был интерфейс программного продукта, сначала нужно научить студента работать с самим инструментом. При этом следует учесть временные ограничения, налагаемые учебным планом по специальности. Можно, конечно, предоставить студентам оригинальную документацию по TAU для самостоятельного изучения, но тут препятствием становится низкий уровень владения техническим английским, что является второй проблемой. Поэтому возникла идея создать собственный пакет интерактивных учебно-методических материалов на родном языке, доступных не только в интрасети, но и в Интернет [12].

Третьей проблемой стал выбор технологии для создания учебно-методических материалов. Первая мысль – использовать flash, как наиболее «модную» технологию для создания образовательного контента. Впрочем, никто из нас не владел этой технологией на достаточном уровне, несмотря на то, что уже имелся некоторый опыт создания flash-презентаций в PowerBullet [13].

Второй вариант, который мы рассматривали – это создание flash-контента с использованием пакета Wink [14]. Wink – хорошее средство для создания обучающих материалов по использованию ПО, обладающее простым интуитивным интерфейсом. Создание обучающего контента в Wink сводится к видеозаписи сеанса работы с ПО, использованию которого мы собираемся учить и последующей обработке в форме добавления комментариев, всплывающих подсказок и диалоговых окон. Результат – нечто вроде видеоролика с примечаниями (рис. 1).

Microsoft Producer [15] известен тем, что это почти единственный на рынке продукт для создания презентаций в формате «живая голова». У нас уже есть некоторый опыт создания лекций в этом формате [16]. Как и Wink,

Producer может быть бесплатно загружен с сайта производителя. Многие учебные заведения и IT-компании приняли MS Producer в качестве корпоративного стандарта для дистанционного образования. В таком качестве Producer используется в юридическом образовании США, а также в Microsoft, Intel и ряде других крупных компаний. Выбранный формат позволяет сделать учебные материалы действительно интерактивными, мультимедийными и готовыми к публикации в Интернет или интранет без какой бы то ни было дополнительной обработки.

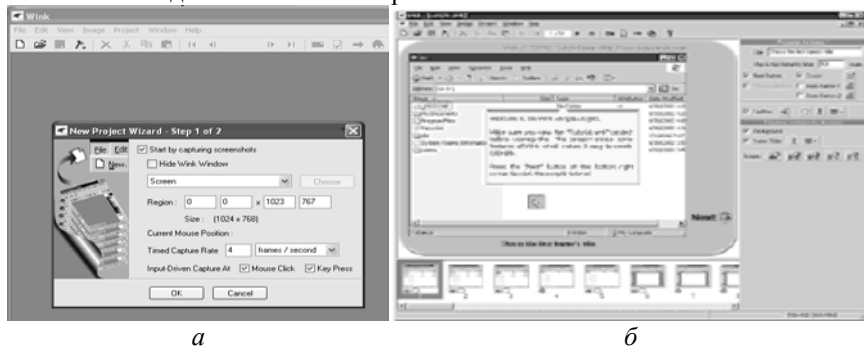


Рис. 1. Среда Wink (а) и созданный в ней ролик (б)

В данном случае, это будет скорее не «живая голова», а «живое видео», т.е. видеозапись действий, производимых с ТАУ (впрочем, можно смонтировать видео и вставить фрагменты с преподавателем). Интерфейс Producer очень похож на Windows Movie Maker, поэтому, если вы хотя бы раз монтировали домашнее видео с помощью этого продукта, создание «живой головы» не должно вызвать никаких затруднений (более того, эти два продукта часто используют совместно). Элементы управления практически такие же и находятся на привычных местах. Если мы создаем именно лекцию, можно импортировать видеозапись и синхронизировать ее затем со слайдами презентации PowerPoint. Для создания же учебных материалов по работе с ПО следует применять «Мастер захвата» MS Producer, который надо использовать в режиме «Захват экранного видео и аудио с микрофона» (рис. 2). Producer позволит синхронизировать полученное видео со слайдами и сгенерирует меню, чтобы слушатели сами могли выбрать необходимые им моменты видеозаписи. Для того, чтобы получить готовую «живую голову», достаточно нажать кнопку Publish в панели инструментов (рис. 3).

Таким образом, довольно легко можно создать набор интерактивных видеопрезентаций, которые будут наглядно иллюстрировать выполнение типичных операций в Telelogic TAU G2 и которые можно будет выложить в электронной библиотеке колледжа, дать студентам домой для самостоятельного изучения, поместить на сайте методической поддержки дисциплины и выложить на создаваемом сейчас в ППК портале дистанционного об-

разования. Такие материалы, думаем, будут очень полезным помощником для студента в освоении этого замечательного продукта, ведь применение TAU в учебном процессе – отличная возможность дать студентам значимый опыт в области программной инженерии и объектно-ориентированного проектирования и дизайна.

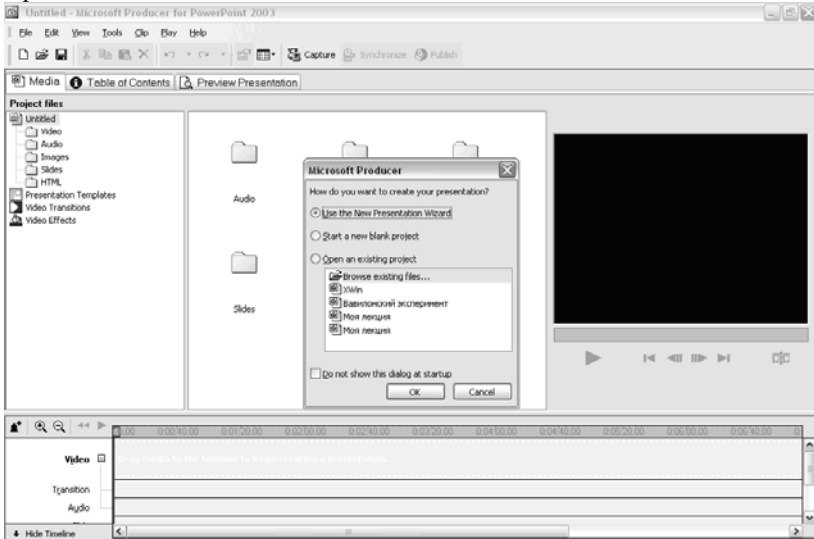


Рис. 2

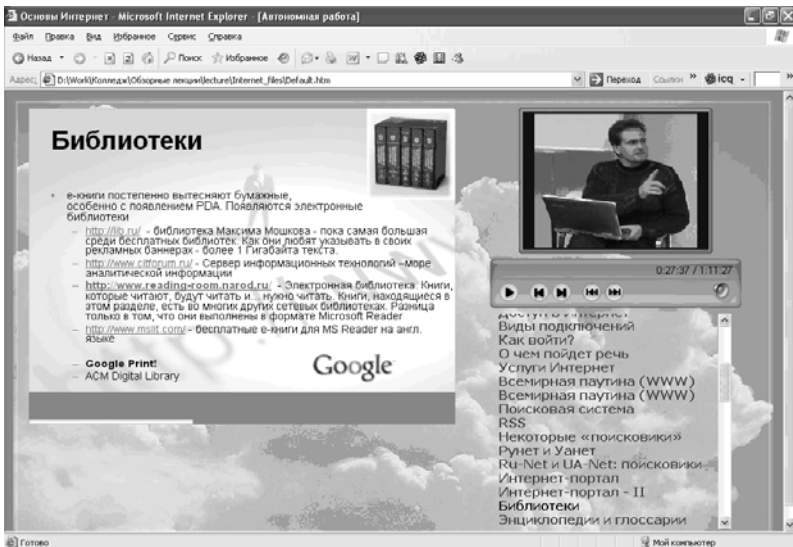


Рис. 3

На данный момент созданы видеопрезентации для таких тем, как «Основы интерфейса ТAU» и «Создание канонических диаграмм». Producer можно использовать для создания любых онлайн-презентационных материалов – это и видеолекции, и задания для лабораторных работ, рекламные материалы, что угодно. Как уже говорилось выше, «живая голова» – очень удобный и практичный формат, который, без сомнения, со временем станет популярным и в странах СНГ.

Есть еще идеи использовать продукты xatshow и xatquiz [17] для создания учебных DVD/VideoCD, а также тестовых заданий, которые можно выполнять дома даже при отсутствии компьютера (просто вставив диск в DVD-проигрыватель), но это уже тема для другой статьи.

Литература:

1. Официальный сайт ППК НТУ «ХПИ» <http://www.ppc-kpi.ukrbiz.net>
2. Association for Computer Machinery <http://www.acm.org/>
3. Русский перевод Computing Curricula 2001 <http://se.math.spbu.ru/cc2001>
4. Официальный сайт проекта «Виртуоз» <http://curricula.ru/virtuoso/>
5. Сайт IT-Lab ННГУ <http://itlab.unn.ru/>
6. MSDN Academic Alliance Curriculum Repository <http://www.microsoft.com/Rus/Msdnaa/Curricula/Default.aspx>
7. OMG UML Page <http://www.uml.org>
8. Vladimir Pavlov, Anton Yatsenko. Using Pantomime in Teaching OOA&OOD with UML // Proceedings of the 18th IEEE Conference on Software Engineering Education and Training (CSEE&T) - Ottawa (Canada), IEEE, 2005
9. Бабич А.В. Обзор CASE-средств для построения диаграмм UML // Теорія та методика навчання математики, фізики інформатики: Збірник наукових праць. Випуск V: в 3-х томах. – Кривий Ріг: Видавничий відділ НМетАУ, 2005. – Т.3: Теорія та методика навчання інформатики. – С. 8.
10. Gentleware Inc. Homepage <http://www.gentleware.com/>
11. Telelogic Corp. Homepage <http://www.telelogic.com/>
12. Бабич О.В. Сторінка для студента <http://barhan.poltava.ua/lug/Students>
13. Powrbullet homepage <http://www.powerbullet.com/>
14. Wink Homepage <http://www.debugmode.com/wink/>
15. MS Producer Homepage <http://www.microsoft.com/office/powerpoint/producer/prodinfo/default.aspx>
16. ППК НТУ «ХПИ»: Пресс-релизы <http://ppc-kpi.ukrbiz.net/prtext.ukrbiz?prnum=47107>
17. Xat: The Internet Technology Homepage <http://www.xat.com/>

КОМПЬЮТЕРНЫЕ ПРЕЗЕНТАЦИИ В ПРЕПОДАВАНИИ ИНФОРМАТИКИ В ТЕХНИЧЕСКОМ ВУЗЕ

А.В. Шматко, И.А. Яковлева
г. Харьков, Академия гражданской защиты Украины
fd.apbu@list.ru

Использование технических средств и информационных технологий в процессе обучения стало неотъемлемой чертой современного образования. Применение компьютерной техники порождает новые и необычайные возможности по совмещению текстовой и аудиовизуальной информации, которую можно организовать для самых различных учебных условий. В зависимости от специфики изучаемых дисциплин данные возможности применяются практически в каждой учебной ситуации.

Нетрадиционные возможности открываются уже при возможности использования традиционных универсальных программ. Особенно выделяется программы презентаций. Среди программ, входящих в пакет Microsoft Office, к таким можно отнести приложение PowerPoint. Оно позволяет создавать компьютерные (электронные) презентации (слайд-шоу), способные оживить даже самую апатичную аудиторию. Слушатель перестает быть пассивным приемником информации и становится активным участником процесса.

Готовая презентация может быть представлена в нескольких вариантах:

- на экране вашего и ученического мониторов;
- на экране через мультимедийный проектор;
- на экране с помощью эпидиаскопов, используя прозрачную пленку;
- на экране с помощью диапроекторов, используя 35-миллиметровые слайдофильмы;
- распечатана на бумаге и роздана присутствующим.

Под электронной презентацией понимается особый документ, который отличается комплексным мультимедийным содержанием и особыми возможностями для управления воспроизведением. В презентации могут быть вставлены программные элементы, подготовленные в других программах (текстовых и табличных процессорах, графических редакторах, средствах обработки звуковой и видеoinформации и другие).

Презентация представляет собой набор слайдов, которые могут представлены как в черно-белой гамме, так и с использованием различных цветовых схем и видов оформления, разработанных профессиональными дизайнерами или самим автором презентации. Слайды могут содержать текст, таблицы, диаграммы, рисунки, организационные диаграммы, видеоклипы, звуковое сопровождение, гиперссылки на другие слайды и документы (презентации, таблицы, диаграммы и т.д., находящиеся на данном компьютере или в Internet). Отдельные элементы слайда могут иметь эффекты анимации.

Использование презентаций в учебном процессе дает возможность преподавателю обеспечить удобное и наглядное представление учебного материала. При этом можно использовать различные виды презентаций, которые будут отличаться способом их реализации на компьютере. Воспроизведение презентации может быть автоматическим или интерактивным, в том числе и дистанционным. В зависимости от способа воспроизведения различают такие их виды:

- презентации со сценарием;
- интерактивные презентации;
- автоматические (самовыполняющиеся) презентации.

Презентация со сценарием относится к наиболее распространенным мультимедийным презентациям. Она напоминает традиционную презентацию со слайдами, дополненную средствами показа цветной графики и анимации с выводом видеоматериала на большой экран или монитор. В такой презентации обеспечивается возможность во время показа вносить изменения в процесс демонстрации. В слайды презентации включают титры, которые плывут по экрану и содержат дополнительные объяснения. Использование анимационного текста в сочетании с анимационными диаграммами, графиками и иллюстрациями позволяет сосредоточить внимание слушателя на основных положениях рассматриваемой темы и способствует лучшему запоминанию информации. Озвучивает материал, как правило, сам преподаватель.

Презентация со сценарием подходит как для проведения лекций, так и для практических занятий и семинаров. Она позволяет все время поддерживать интерес к изучаемой теме. В ходе занятия, отвечая на вопросы студентов, преподаватель может изменить последовательность предоставления материала или настроить презентацию с учетом конкретной аудитории перед началом занятия.

Интерактивная презентация обеспечивает диалог пользователя с компьютером. Во время изложения материала, преподаватель может принять решение, какая информация для него наиболее важна, и осуществить ее выбор. Выбор производится нужного объекта при помощи щелчка мышью или нажатием соответствующей клавиши на нужном объекте экрана. Запрашиваемая информация сразу же появляется на экране.

Интерактивная презентация позволяет осуществлять поиск нужной информации, углубляясь в нее на столько, на сколько это было предусмотрено разработчиком презентации. Например, изучение основного окна какой-либо прикладной программы начинается с изучения общих сведений о его элементах, которые отображаются на экране. Затем после нажатия на гиперссылку, на кнопку или значок отображается более подробная информация о конкретном назначении каждого из элементов. При этом порции информации могут быть представлены в разном виде:

- в графическом или текстовом;

- при помощи анимации или видеоклипов;
- прочтением текста «от автора» с использованием звуковых эффектов.

Очевидно, что можно использовать все перечисленные виды подачи информации в различных сочетаниях.

Для интерактивной презентации характерным является то, что в ней заложена возможность захватывать слушателя и поддерживать в нем заинтересованность материалом на протяжении всего времени изложения информации.

Используя такие презентации, студенты могут осваивать материал самостоятельно. Кроме того, их можно распространять через сеть Internet.

Автоматическая презентация представляет собой законченный информационный продукт. Такие презентации позволяют охватывать ту или иную тему в простой и наглядной форме. При этом презентацию можно перенести на видеопленку, дискету, компакт-диск и распространять среди студентов дневной формы обучения или рассылать по заказу студентов заочной формы обучения.

В презентации огромное значение имеет композиция слайда. Через нее можно раскрыть и определенную точку зрения, и вызвать чувство, и, наконец, выразить понятие.

Главным средством соединения слайдов презентаций является текст. Именно через него протягивается связь от кадра к следующему изображению, текст готовит к восприятию содержания соседнего кадра. Следовательно, преподаватель может выстраивать слайды в определенном порядке и скреплять в зависимости от логики своего изложения. Однако такой «монтаж» кадров – искусство: глубокое и точное владение им дается после длительной и тщательной тренировки. Текст в презентациях управляет ритмом восприятия презентации. Величина текста непосредственно влияет на длительность рассматривания, на восприятие и осознание зрительного ряда. Чем динамичнее и короче во времени действие, срез с которого дан в изображении, тем короче, но эмоционально насыщеннее должен быть текст.

Таким образом, презентация имеет свой выразительный язык. Этот язык не идентичен языку слов, он является своеобразным синтезом изображения и слова при явном приоритете изображения. Отсюда не следует, что в презентации не может быть отдельных слайдов-текстов, но их должно быть мало, иначе презентация превратится в статью на экране.

Выбор темы и подбор материала для презентации остаются творческими процессами автора и не автоматизируются. Автоматизации подлежат лишь процессы воплощения авторских идей в готовый продукт и процессы его публичного воспроизведения.

Особенность презентаций также состоит в том, что презентация – это своеобразная модель, специально сконструированная для определенных целей. В этом случае может быть интересен опыт, полученный преподава-

телями Академии гражданской защиты Украины при преподавании дисциплин «Информатика и компьютерная техника» и «Основы информатики и вычислительной техники».

Какие учебные функции может выполнять презентация? Ответ на этот вопрос в значительной мере зависит, конечно, от специфики предмета преподавания. Так при изучении информационных технологий намного проще продемонстрировать на экране компьютера те или иные действия, чем пытаться рассказать о них. Очень большую помощь в освоении материала оказывает использование анимационных изображений тех или иных действий, выполняемых при настройке программ.

Необходимо отметить, что отличительной особенностью дисциплин, связанных с изучением информатики и компьютерных технологий, является большое количество иллюстративного материала (диалоговые окна, окна приложений и т.д.) который трудно передать обычными средствами – мел, доска, плакаты.

Особенно актуальным является использование презентаций при изучении новой темы в форме лекции. В этом случае они служат планом лекции. Наиболее значительные эпизоды выделяются, к ним даются особые вопросы. Те слайды, которые находятся в конце презентации, направлены на обобщение материала. В ходе решения этих задач преподаватель не должен сковывать инициативу учащихся, специфика экранного изображения позволяет практически все задания сделать предметом массового обсуждения. Это одна из важнейших положительных сторон экранных пособий. Педагог, разумеется, оценит и возможность держать на экране изображение сколько угодно долгое время, определяемое своеобразием задания, характером его решения, манерой работы аудитория.

Таким образом, презентация может решить следующие проблемы с обучением студентов:

- пробуждение и возбуждение усиленного интереса к учебной теме и ориентировка в ней;
- иллюстрирование объяснения;
- изложение новой учебной информации – в союзе со словом, слова преподавателя, анализом текста и т.д.;
- формирование и формулировка условий познавательной задачи;
- обобщение учебного материала, повторение с углублением и расширением темы.

Отметим, что показ презентаций не должен быть самоцелью. Включать их нужно и можно только тогда, когда отчетливо понимается ее необходимость и методическую целесообразность. В противном случае демонстрация станет инородным действием, будет мешать усвоению учебного материала. Казалось бы, истина эта тривиальна, но, к сожалению, ее необходимо повторить: аудиовизуальные средства действуют на обучаемых очень сильно, влияя и на их эмоции, и на ход мысли, и на всю учебную деятельность.

ПОДХОДЫ К РАЗРАБОТКЕ ВИРТУАЛЬНОЙ ЛАБОРАТОРИИ ДЛЯ ДИСЦИПЛИНЫ “ТЕОРИЯ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ”

Н.С. Жуков¹, Л.Л. Жукова²

¹ г. Кривой Рог, Криворожский металлургический факультет
Национальной металлургической академии Украины

² г. Кривой Рог, Криворожский учебно-методический центр
Запорожского национального университета
lab415@etex.dp.ua

Полноценное понимание и усвоение изучаемого материала достигается после практического опробования теоретических знаний, поэтому учебными планами всех фундаментальных дисциплин предусмотрены практические или лабораторные работы. Однако не все учебные заведения имеют соответствующую материальную базу. Кроме того, некоторые работы практические не могут быть выполнены в лабораторных условиях. В таких случаях широко используются различные виды моделирования, создают тренажеры и виртуальные лаборатории, базирующиеся на современных математических методах и компьютерных технологиях.

Сегодня у научных и инженерно-технических работников имеется большое количество программных продуктов, в основном – зарубежного производства (MathCAD, Matlab, Electronics Workbench и др.). Полноценное их использование в силу не полной документированности или дороговизны отдельных составляющих модулей (библиотек) затруднительно. У некоторых моделирующих программ (среди наиболее известных – MathConnex в пакете Mathcad 2000, и VisSim) имеются интерфейсы для расширения их возможностей конечным пользователем. При этом возможны несколько технологий расширений. Первая – внедрение ActiveX-объектов других программных продуктов (MathConnex). Вторая – динамический обмен данными между программами посредством DDE или OPC серверов/клиентов. Третья – создание библиотечных модулей с новыми для пакетов процедурами и функциями. Подключение внешних динамически загружаемых библиотек позволяет существенно расширить возможности пакета (например, VisSim). Пользователь может, используя известный ему язык программирования, создавать новые линейные, нелинейные, дискретные, частотно зависимые элементы, расширяя библиотеку стандартных; разрабатывать собственные процедуры и функции измерений, анализа, работы с базами данных; подключать как собственные, так и стандартные платы сбора данных (DAQ-board) или OPC-серверы для исследования, мониторинга, управления внешним объектом.

Это предоставляет исследователю практически неограниченные возможности. Но студент при этом ограничен той средой, в которой работает. С одной стороны, он формально моделирует математические зависимости,

не ощущая при этом полной связи их с физическими объектами (работая, например, в пакете MathCAD или VisSim). Или, выполняя работу с «электронными объектами» (работая, например, в пакете Electronics Workbench), не имеет методической направленности работы (если не считать, что перед ним находится методическое пособие, в которое можно посмотреть). Преподаватель в обоих случаях должен использовать дополнительные средства оперативного контроля знаний. Ведь зачастую бессмысленно тратится время, если студент не готов к работе, а преподаватель отвлекается на выяснение этого, вместо активной работы с другими студентами.

Для лучшей организации проведения лабораторных работ используют различные интегрирующие среды, которые позволяют непринужденно и логично переходить из одной среды в другую. Например, в текстовом режиме представляется информация о предстоящей работе: тема, задание, порядок выполнения. Затем с помощью тестов производится проверка готовности студента к выполнению работы. Это происходит в удобном для конкретной темы виде. При этом используют рисунки, графики, анимацию. Если студент готов, ему автоматически предоставляется среда, соответствующая теме работы. В случае неготовности, он такой возможности лишается. После выполнения одного задания, он получает возможность перейти к другому и т.д.

Для создания такой виртуальной лаборатории используют Web-технологии или разрабатывают интегрирующую среду, используя средства визуального проектирования интерфейсов и объектно-ориентированного программирования. При квалифицированном подходе оба варианта позволяют получить практически равноценный результат. Однако, по мнению авторов, для реализации доступа к реестру, организации взаимодействия между различными процессами, логической обработки информации и других операций непосредственное программирование проще, чем их реализация с использованием сценариев VBScript или JScript и элементов ActiveX.

Кратко рассмотрим основные возможности и их реализацию с использование средств объектно-ориентированного программирования в среде Delphi. Как отмечалось выше, при выполнении практической работы необходимо обеспечить переход из одной среды выполнения в другую, т.е. необходимо из одного процесса породить другой дочерний. Для этого рекомендуется вместо функции WinExec использовать более мощную CreateProcess.

Например, после получения допуска выполнить работу (после прохождения теста на готовность), для ее продолжения необходимо перейти в приложение Electronics Workbench. Для его запуска из текущего процесса (по событию Go_to_EWClick), использована процедура с рассматриваемой функцией:

```
procedure TForm_Lab1.Go_to_EWClick(Sender: TObject);  
var  
    lpStartupInfo:TStartupInfo ;
```

```

    lpProcessInformation:TProcessInformation ;
begin
    FillChar(lpStartupInfo,SizeOf(lpStartupInfo),#0);
    lpStartupInfo.cb:=SizeOf(lpStartupInfo);
    lpStartupInfo.dwFlags:=StartF_UseShowWindow;
    lpStartupInfo.wShowWindow:=SW_ShowNormal;
    if not CreateProcess(
        'E:\Program Files\EWB512\WEWB32.EXE',
        'E:\Work_delphi\Virtual_Lab\RC.ewb',
        nil,
        false,
        Create_New_Console or
        Normal_Priority_Class,
        nil,
        nil,
        lpStartupInfo,
        lpProcessInformation)
    then
        ShowMessage(IntToStr(GetLastError))
    else
        begin
            WaitForSingleOb-
ject(lpProcessInformation.hProcess,10000);
            CloseHandle(lpProcessInformation.hprocess);
            end;
end;

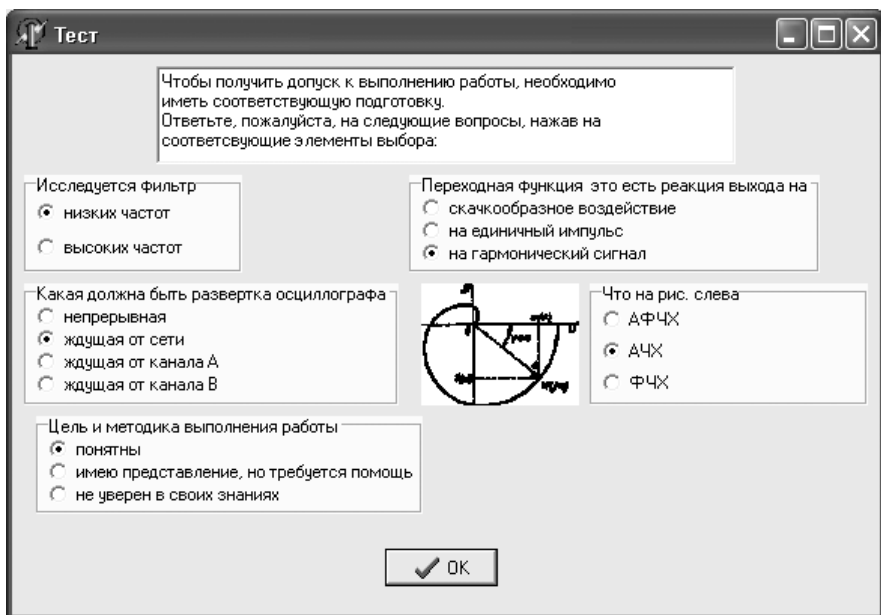
```

Здесь флаги `Create_New_Console` и `Normal_Priority_Class` означают, что новое консольное приложение (запускаемый файл `WEWB32.EXE`, которому передается параметр `'E:\Work_delphi\Virtual_Lab\RC.ewb'`), будет запущено в новом окне с нормальным приоритетом. Это окно будет иметь нормальные размеры (`lpStartupInfo.wShowWindow:=SW_ShowNormal`). Если процесс по какой-либо причине не сможет выполниться, то об этом будет передано соответствующее сообщение: `ShowMessage(IntToStr(GetLastError))`. Если процесс будет запущен, то дальше ожидается его завершение, о чем будет сообщено через параметр `lpProcessInformation.hProcess`. В этом примере для лучшего понимания параметров функции `CreateProcess` конкретно указаны запускаемый файл и передаваемые параметры. Реально это реализуется автоматически. Если известно, какое приложение надо использовать для данной работы, его полное имя находится в реестре и записывается в соответствующий указатель. Параметром в этом случае будет заранее подготовленный файл, например, в `Electronics Workbench`, который содержит средства, для выполнения работы. Студенту необходимо будет только соответствующим образом их соединить и настроить параметры. По ходу выполнения работы у него всегда «под рукой» находится подробное описание задания. Для этого удобно использовать компоненты `TRichEdit` и `TImage`. Текстовые файлы в `RTF`-формате и рисунки находятся в соответствующих папках и по мере

папках и по мере необходимости в указанные компоненты. Для получения дополнительной оперативной информации, которая может потребоваться во время работы, предусмотрена контекстно-чувствительная справочная система. Она включает основные сведения по дисциплине «Теория автоматического управления» (разделы «Линейные непрерывные и дискретные цепи»), а также о работе с пакетами MathCad, Matlab (пакет SimuLink), Electronics Workbench – в объеме, необходимом для выполнения работы. Для удобства работы, предусмотрена возможность выбора языка общения: русского или украинского.



При первом вхождении в программу нажимают одну из них. При этом осуществляется настройка на обращение к данным одной из папок: Rus и Ua, каждая из которых, в свою очередь, содержит папки с тестами (Test) и с файлами другой текстовой информации (Text) на соответствующих языках. Как уже отмечалось, для допуска к работе необходимо ответить на некоторые вопросы по данной теме. На рисунке приведен пример тестового окна к лабораторной работе по исследованию фильтров. Предварительно (в другом окне) было сформулировано задание лабораторной работы и приведен рисунок со схемой электрической принципиальной. После этого предлагается следующее задание. Руководитель работы может заранее определить, на сколько вопросов и на какие необходимо дать положительные ответы, чтобы без его вмешательства произошел переход в среду виртуальной лаборатории, а при каких допуск невозможен или требуется дополнительное собеседование.



При формировании концепции лабораторных работ основной акцент сделан на том, чтобы студент мог перейти от физического восприятия основ дисциплины к абстрактному. С этой целью работы, связанные с изучением свойств динамических звеньев, выполняются с использованием моделирования физического подобия элементов (пакет Electronics Workbench), а проверка устойчивости, точности регулирования – с использованием структурных схем (пакет SimuLink) или символьных преобразований и вычислений (пакет Mathcad).

В настоящее время разработана оболочка, проверено взаимодействие всех составляющих, производится наполнение необходимой информацией.

Литература:

1. Фаронов В.В. Delphi 6. Учебный курс. – М.: Изд. Молгачева С.В., 2003.
2. Гультяев А.К. Matlab 5.3. Имитационное моделирование в среде Windows: Практическое пособие. – СПб.: Корона принт, 2000. – 400 с.
3. Теория систем автоматического управления / В.А. Бессекерский, Е.П. Попов. – Изд. 4-е перераб. и доп. – СПб, Изд-во «Профессия», 2003. – 752 с.
4. Воронов А.А. Теория автоматического управления. Часть первая. – М.: «Высшая школа», 1986. – 367 с.

ОТ РУЧНОГО УПРАВЛЕНИЯ К ПРОГРАММНОМУ

Р.М. Балабай, Д.В. Рябчиков

г. Кривой Рог, Криворожский государственный педагогический университет
Dimail@list.ru

В настоящее время множество отечественных предприятий требуют автоматизации в том или ином виде. В современных условиях рынка выживают те, кто наиболее конкурентоспособен, кто умеет организовать свои производственные процессы с наибольшей эффективностью и наименьшими затратами. Очевидно, что современному предприятию, для того, чтобы расти и развиваться, необходима «новая кровь» в виде новых технологий и умелых специалистов. Компьютерные системы все больше охватывают производственный сектор, на рабочих местах операторов появляются эффективные средства для оперативного контроля и управления технологическими процессами. Грамотно построенная автоматизированная система управления технологическими процессами (АСУ ТП) – это набор программных и аппаратных средств, создание которых требует больших усилий целого коллектива разных специалистов. Принципы управления промышленными объектами можно переносить и в управление моделями.

Ниже мы схематично представим вариант ознакомления студентов педагогических вузов физических специальностей с элементами теории автоматического управления и её практического применения.

В методике обучения анализ исторического развития предмета исследования является очень важным и часто используемым приемом. Последуем ему и мы.

Автоматика (А.), отрасль науки и техники, охватывающая теорию и принципы построения систем управления, действующих без непосредственного участия человека. Как самостоятельная область техники А. получила признание на 2-й Мировой энергетической конференции (Берлин, 1930), где была создана секция по вопросам автоматического и телемеханического управления. В СССР термин «А.» получил распространение в начале 30-х гг.

А. как наука возникла на базе теории автоматического регулирования, основы которой были заложены в работах Дж.К. Максвелла (1868), И.А. Вышнеградского (1878), А. Стодолы (1899) и др. В самостоятельную научно-техническую дисциплину окончательно оформилась к 1940. История А. как отрасли техники тесно связана с развитием автоматов, автоматических устройств и автоматизированных комплексов. В стадии становления А. опиралась на теоретическую механику и теорию электрических цепей и систем и решала задачи, связанные с регулированием давления в паровых котлах, хода поршня паровых и частоты вращения электрических машин, управления работой станков-автоматов, АТС, устройствами релейной защи-

ты. Соответственно и технические средства А. в этот период разрабатывались и использовались применительно к системам автоматического регулирования. Интенсивное развитие всех отраслей науки и техники в конце первой половины 20 в. вызвало также быстрый рост техники автоматического управления, применение которой становится всеобщим [1; 2]

Вторая половина 20 в. ознаменовалась дальнейшим совершенствованием технических средств А. и широким, хотя и неравномерным для разных отраслей народного хозяйства, распространением автоматических управляющих устройств с переходом к более сложным автоматическим системам, в частности в промышленности – от автоматизации отдельных агрегатов к комплексной автоматизации цехов и заводов. Существенной чертой этого периода является использование А. на объектах, территориально расположенных на больших расстояниях друг от друга (например, крупные промышленные и энергетические комплексы, системы управления космическими летательными аппаратами и т.п.). Для связи между отдельными устройствами в таких системах применялись средства телемеханики, которые совместно с устройствами управления и управляемыми объектами образовывали телеавтоматические системы. Большое значение при этом приобретали технические (в т.ч. телемеханические) средства сбора и автоматической обработки информации, т.к. многие задачи в сложных системах автоматического управления могут быть решены только с помощью вычислительной техники. Наконец, теория автоматического регулирования уступила место обобщённой теории автоматического управления, объединяющей все теоретические аспекты А. и составляющей основу общей теории управления [3].

В сфере обучения теоретическим основам и практическим приемам автоматизации существуют определенные трудности. Если речь идет о подготовке специалиста в области автоматизации, то наличие в учебной программе соответствующих дисциплин, специально оборудованных аудиторий, производственной практики позволяет успешно их преодолевать. Тогда как попытка в рамках курса «Радиоэлектроника и основы автоматики» для специальности «Физика и информатика» педагогических вузов реализовать постановку лабораторной работы такой направленности вызывает серьезные затруднения, как в силу значительных финансовых затрат на приобретение оборудования, так и в силу ограниченного времени на изучение соответствующих курсу дисциплин да и самого курса.

Одним из возможных путей решения указанных проблем является постановка лабораторных работ, являющихся малыми моделями АСУ ТП. Ведь модель является выражением упрощенного представления об оригинале, уточняемом в процессе её конструирования. По мере проведения экспериментов с моделью знания об оригинале возрастают.

В среде специалистов, занимающихся проблемами автоматизации, существуют устоявшиеся близкие по смыслу термины, обозначающие анало-

гичные по функционированию системы: АСКУ, АСУ ТП, телемеханика. АСУ ТП – автоматизированная система управления технологическим процессом, АСКУ – автоматизированная система контроля и управления. И, наконец, классическое название подобного класса систем, породившее одноименное научное направление – телемеханика [4]. Но как ни определять систему, цель у нее одна – контроль параметров, характеризующих течение процесса, и своевременное оказание воздействий, удерживающих его в требуемых рамках. И при разработке реальных автоматизированных систем и модельных приходится проходить следующие этапы [5]:

- изучение процесса для достижения его оптимального протекания;
- выявление существенных параметров процесса и установление связи между ними;
- измерение контролируемых параметров процесса;
- ввод измеренных параметров в вычислительную технику;
- обработка информации с целью формирования критериев протекания процесса;
- передача воздействий, выработанных программным образом (или оператором) на процесс;
- фиксирование и оповещение о возникновении аварийных ситуаций и состоянии программно-аппаратных средств;
- формирование информации о течении процесса.

Реализация каждого из названных этапов – это сложная задача, требующая привлечения различных электронных устройств, интерфейсных элементов их согласующих, специального программного обеспечения.

Мы попытались реализовать этапы «наивным образом», не прибегая к специальным программируемым процессорам и интерфейсным элементам, а используя персональный компьютер, язык программирования, устройство аналого-цифрового ввода-вывода, которое осуществляет связь внешних устройств с компьютером по шине PCI и поддерживает наиболее распространенные языки программирования. В качестве управляемого процесса мы выбрали отслеживание траектории движения тележки, которую необходимо провести по заданному маршруту [6]. Отклонения от указанного направления движения должны компенсироваться системой управления.

Для создания системы управления нам необходимы два датчика, которые следят за линией, задающей траекторию движения. В случае ее исчезновения из «поля зрения» любого из датчиков передается сигнал на компьютер. Задачей компьютера является обработка в соответствии с программой полученных от датчиков сигналов (входных данных) и подача на органы управления транспортным средством регулирующих напряжений.

Каждый из блоков, узлов разрабатываемой системы управления движения по заданной траектории на практике требует своей детализации, требующей знания в области радиоэлектроники, электротехники, программирования. Итак, и при разработке и при выполнении этой лабораторной рабо-

ты студент проходит все этапы проектирования реальной АСКУ от компонентов, обеспечивающих ввод-вывод информации, до текста компьютерной программы, обеспечивающий контроль, управление и мониторинг процесса, при этом получая новые знания и проверяя на практике имеющиеся.

Литература:

1. Теория автоматического управления. / Под ред. Воронова А.А. – М.: Высшая школа, 1975. – 570 с.
2. Воронов А.А. Основы теории автоматического управления. Ч. 3. – М.-Л., 1970. – 620 с.
3. Романенко Л.В., Игнатенко В.К., Адаптивное управление технологическими процессами на базе МикроЭВМ // Современные технологии автоматизации. – 2004. – №3. – С. 46.
4. Незнанов В.Н. WEB и SCADA. Угроза или приглашение к сотрудничеству? (продолжение) // Промышленные измерения. Контроль. Автоматизация. Диагностика. – 2003. – № 1-2. – С. 10–12.
5. Незнанов В.Н. WEB и SCADA. Угроза или приглашение к сотрудничеству? (продолжение) // Промышленные измерения. Контроль. Автоматизация. Диагностика. – 2003. – № 3-4. – С. 14–15.
6. Шварце Х., Хольцгрефе Г.В. Использование компьютеров в регулировании и управлении. – М.: Энергоатомиздат, 1990. – 176 с.

ЭТАПЫ ВЕРИФИКАЦИИ КОНФИГУРИРУЕМЫХ НА КРИСТАЛЛЕ ЦИФРОВЫХ СИСТЕМ

М.К. Аль Мади^α, В.А. Андриенко^β, В.Г. Рябцев^γ

г. Черкассы, Черкасский государственный технологический университет

^α survas@maktoob.com

^β Andrienko22@ukr.net

^γ Volodja18@ukr.net

Современная электронная технология позволяет реализовать на одном кристалле основные компоненты цифровых систем (ЦС): микропроцессоры, контроллеры запоминающих устройств и программируемые пользователем логические элементы. Проектирование таких сложных ЦС возможно только с помощью средств автоматизированного проектирования, обеспечивающих достоверную верификацию и тестирование основных компонентов [1; 2].

С целью упрощения средств диагностического обслуживания проектируемых на кристаллах ЦС целесообразно разделить процесс верификации моделей на три этапа. Структуры моделей объектов и средств верификации для различных этапов приведены на рис. 1.

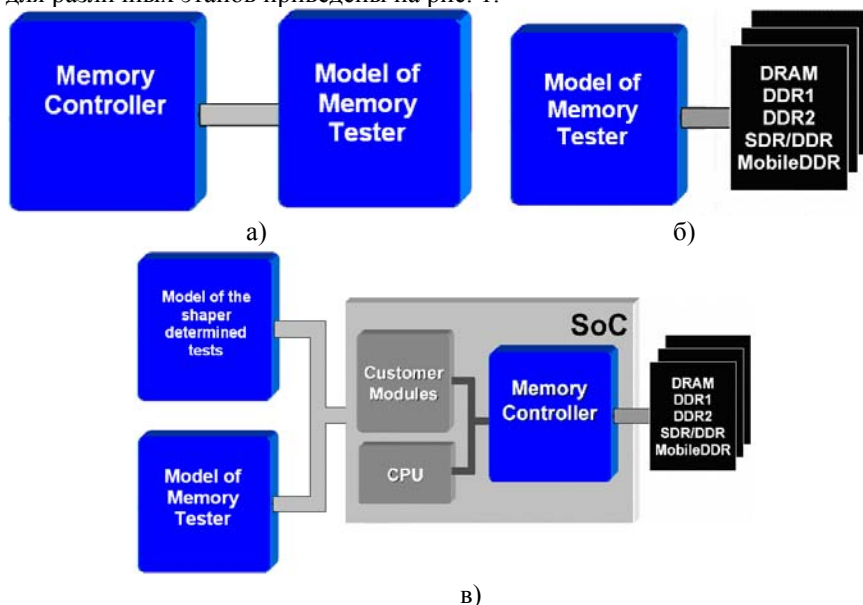


Рис. 1. Этапы верификации моделей цифровой системы:

а) верификация модели микросхемы памяти;

б) верификация модели контроллера;

в) верификация модели цифровой системы, конфигурируемой на кристалле.

На первом этапе осуществляется проверка правильности выполнения всех возможных режимов работы моделью применяемого запоминающего устройства. Выполняемые тесты должны обеспечивать проверку операций записи и считывания пакетов данных всех существующих размеров и возможности формирования допустимых значений параметров управляющих сигналов. Набор тестов создает условия для увеличения спроса на микросхемы памяти, так как они наглядно демонстрируют их возможности, кроме того, уточняются требования к проектируемому контроллеру запоминающего устройства.

Трудоёмкость разработки алгоритмов и программ тестов для верификации моделей микросхем памяти значительно снижается, если их разрабатывать на объектно- и машинно-зависимых языках программирования, которые учитывают особенности структур диагностирующих устройств и микросхем полупроводниковой памяти [3; 4].

В общий проект разрабатываемого изделия предлагается включить модели тестера и исследуемой микросхемы памяти, при этом в данных моделях объединяются порты адреса, данных, кодов операций и синхронизирующих сигналов. При выполнении операций записи и считывания данных в одноименных процедурах формальные операнды заменяются на фактические.

Например, процедура записи данных в запоминающие ячейки объявляется следующим образом:

```
Procedure writes (n: integer; signal CE_b, OE_b, WE_b: out
                 std_logic; signal ADDRESS: out std_logic_vector
                 (addrBusWidth-1 downto 0); signal DATA: inout
                 std_logic_vector (dataBusWidth-1 downto 0));
```

Вызов процедуры с фактическими операндами обеспечивает следующую запись в программе теста:

```
Write (0, CE_b, OE_b, WE_b, A, DQ);
```

Здесь 0 – номер операционного процессора, формирующего тестовые воздействия, CE_b, OE_b – сигналы выборки кристалла и разрешения выдачи данных соответственно, A, DQ – коды адреса и данных соответственно.

Для автоматизированного синтеза программ тестов предлагается набор инструментальных средств, приведенный на рис. 2.

Данные средства позволяют, используя описания алгоритмов, проектировать и отлаживать программы базового набора тестов для микросхемы ёмкостью 256 бит. При этом обеспечивается визуализация процесса выполнения программ путём выдачи на экран монитора текущего состояния запоминающих ячеек.

Из разработанных мобильных программ можно при помощи специализированного генератора автоматизированным способом получать программы тестов на языке Prover для микросхем заданной емкости [5].

Транслятор Prover/VHDL обеспечивает автоматическое преобразование программ тестов на язык VHDL. Сформированные программы тестов могут

использоваться в системе автоматизированного проектирования цифровых систем, например в среде Active-HDL.

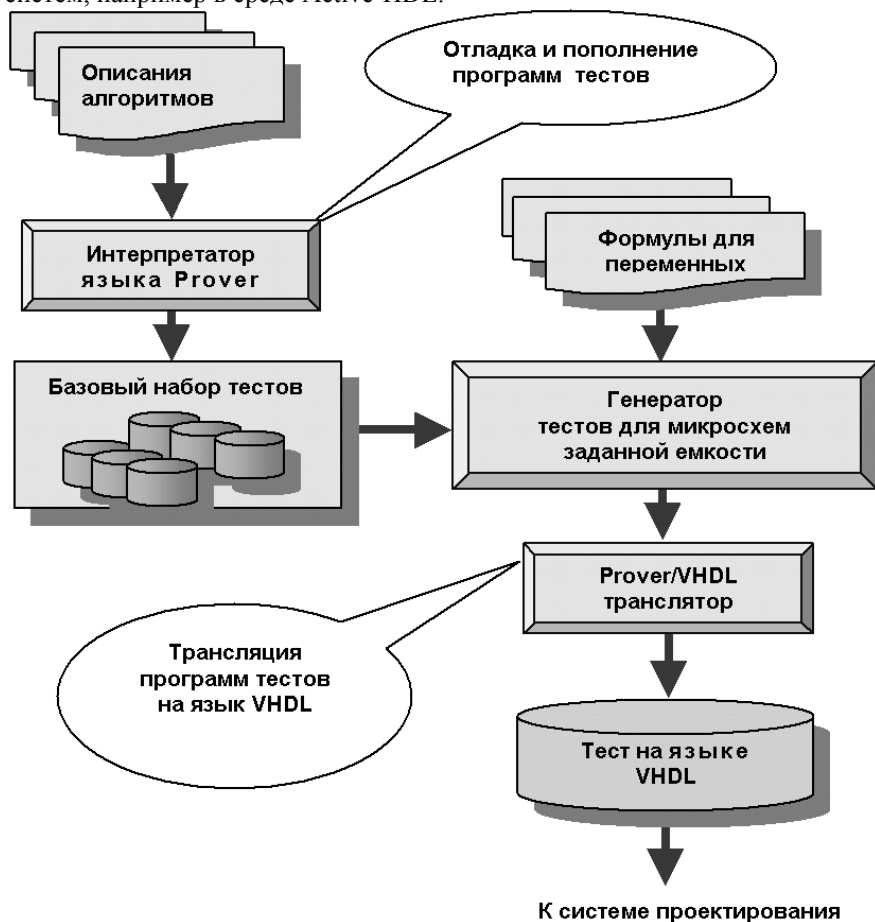


Рис. 2. Инструментальные средства для проектирования программ тестов памяти

На втором этапе проверяется возможности контроллера запоминающего устройства обеспечивать запись кодов в регистр режимов работы микросхем памяти, формировать и принимать пакеты данных. Для проверки выполнения данных операций можно использовать так же модель тестера, который обеспечивает проверку адекватности модели контроллера необходимым режимам работы.

Заключительным является третий этап, на котором осуществляется верификация всей цифровой системы, включая запоминающее устройство.

Эффективная верификация такой сложной системы возможна благодаря использованию алгоритмических и детерминированных тестов, которые формируются моделями соответствующих устройств.

Выводы. Автоматизация программ для верификации моделей конфигурируемых на кристалле цифровых систем возможна за счёт объединения в проекте математических моделей составляющих компонентов, а также средств диагностирования и моделирования их совместной работы. Сформированные тесты обеспечивают проверку правильности взаимодействия логических и запоминающих элементов, что особенно эффективно при отладке структуры контроллеров памяти.

Транслятор Prover/VHDL можно использовать при разработке проектов новых цифровых систем, содержащих встроенные запоминающие устройства, имеющие сложный интерфейс связи с процессором.

Литература:

1. Бибило П.Н. Основы языка VHDL. – М.: Солон-Р, 2000. – 200 с.
2. Семенец В.В., Хаханова И.В., Хаханов В.И. Проектирование цифровых систем с использованием языка VHDL. – Харків: ХНТУРЕ, 2003. – 492 с.
3. Мельников А.В., Рябцев В.Г. Контроль модулей памяти компьютеров. – К.: Корнійчук, 2001. – 172 с.
4. Проектирование и диагностика компьютерных систем и сетей / М.Ф. Бондаренко, Г.Ф. Кривуля, В.Г. Рябцев, С.А. Фрадков, В.И. Хаханов. – К.: НМЦВО, 2000. – 306 с.
5. Рябцев В.Г. Проектування мобільних програм діагностування сучасних мікросхем пам'яті // Вісник Черкаського інженерно-технологічного інституту. – 2002. – № 2. – С. 25-29.

РАЦИОНАЛЬНЫЙ ПОДХОД К ОПРЕДЕЛЕНИЮ КОЛИЧЕСТВА ИНФОРМАЦИИ

К.Ю. Васильев, Д.В. Сущевский, Ю.С. Алпатова, М.Г. Охременко
г. Днепропетровск, Днепропетровский национальный университет
maxim-@mail.ru

На сегодня приоритеты клиентов-потребителей при оценке количества информации являются доминирующими в научно-технической политике при разработке информационной инфраструктуры [1]. Однако далеко не всегда необходимо перекодировать информацию из одной формы, традиционно используемой в специализированных областях, в другую форму презентации данных, подстроенную под индивидуальные знания, умения и привычки конкретного пользователя.

В связи с этим считаем перспективным отдать предпочтение разновидности меры количества информации в виде, предложенном К. Шенноном в 1948 году, по сравнению с полемичным вариантом использования двух альтернативных параметров – парой Галенко [2].

Общая схема работы информационного устройства, которое осуществляет сюръективное отображение сообщений на векторном входе \bar{X} в соответствующий ему вектор \bar{Y} на выходе без ошибок и искажений, состоит в следующем. В терминах К. Шеннона количество передаваемой получателю информации связано с неопределенностью относительно состояния источника сообщений. При передаче независимых и несовместимых сообщений x_i с вероятностью $p(x_i)$ количественной мерой информации будет величина, обратная априорной вероятности сообщения, то есть $1/p(x_i)$, $i = \overline{1, n}$. Для исключения ряда неудобств [1] применяется логарифмическая мера количества информации $I(\bar{X})$ и количественная оценка неопределенности $H(\bar{X})$ сообщения, а именно, при усреднении по всем событиям имеем:

$$I(\bar{X}) = -\sum_{i=1}^n p(x_i) \log_a(p(x_i)), \quad (1)$$

$$H(\bar{X}) = -\sum_{i=1}^n p(x_i) \log_a(p(x_i)), \quad (2)$$

где $H(\bar{X})$ – энтропия, которая с поступлением информации о состоянии системы снижается. Указанная взаимосвязь подчеркивает дуализм информации и энтропии.

Единицы измерения информации (1) и энтропии (2) определяются выбором системы счисления. При оптимальном кодировании информации можно использовать экономический критерий выбора. Под экономичностью системы счисления [3] понимается запас чисел y , которые можно записать в данной системе с основанием a при помощи заданного количества знаков n .

Необходимым условием максимума функции $y(a)$ в точке a_0 является обращение в нуль ее первой производной. В рассматриваемом случае

$$y(a) = a^{n/a}. \quad (3)$$

По формуле дифференцирования для сложной показательной функции u^v имеем:

$$y' = \frac{dy}{dx} = vu^{v-1}u' + u^v \ln(u)v'.$$

Таким образом, производная функции (3) равна

$$\frac{dy}{da} = \frac{n}{a} a^{n/a-1} \cdot 1 - a^{n/a} \ln(a) \frac{n}{a^2} = na^{n/a-2} (1 - \ln a). \quad (4)$$

Приравнивая (4) к нулю, получаем: $\ln(a_0) = 1$, т.е. $a_0 = e$ – основание натурального логарифма. Поскольку слева от $a_0 = e$ значение (4) положительно, а справа – отрицательно, в точке a_0 функция (3) достигает максимума.

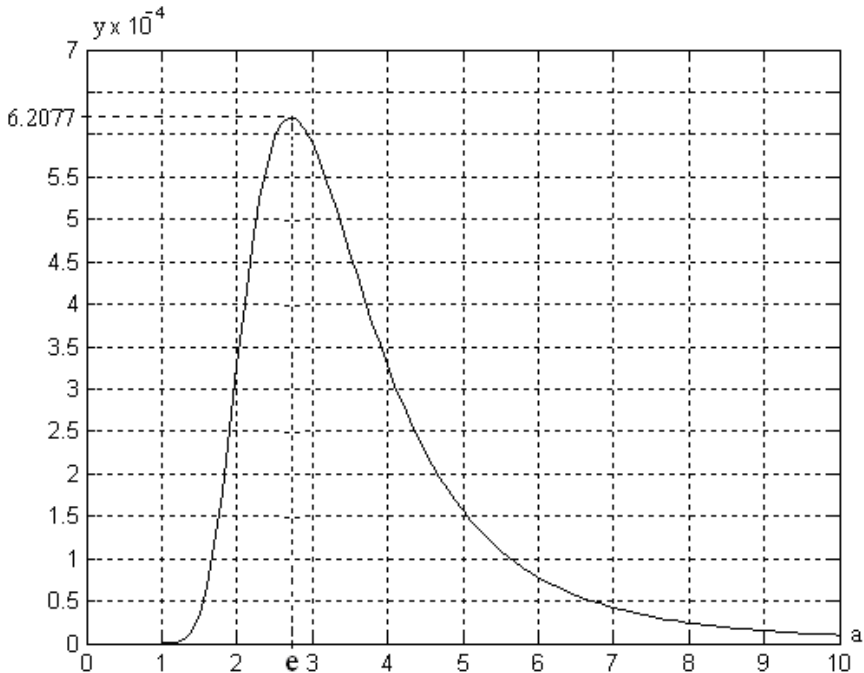


Рис. 1. Зависимость запаса чисел y от системы счисления

Графическая иллюстрация получена посредством системы компьютерной математики Matlab 6.5 [4]. Код программы:

```
a=1:0.1:10;      % ось абсцисс: основание системы счисления a
n=30;           % количество точек отсчета
y=a.^(n./a);    % ось ординат: запас чисел y
```



```
plot(a, y);  
grid;
```

Следует отметить, что при использовании $a=10$ количество информации и энтропии определяется в дитах. При использовании двоичной системы счисления количество информации и энтропии определяется в битах. В математических выражениях рекомендуются натуральные единицы – ниты.

Троичная цифровая техника, например «Сетунь» [5], базируется на трехзначных сигналах и трех стабильных элементах памяти – тритах. Добавление третьего значения значительно усовершенствует двузначную логику, так как тройка ближе других целых чисел к основанию натурального логарифма, кроме того, три цифры – это минимум, позволяющий обозначить положительное, нулевое и отрицательное числа. Высокая надежность троичной цифровой техники была подтверждена соответствующими испытаниями, результаты которых свидетельствуют о высокой стабильности работы и надежности.

В заключение отметим, что количественная оценка информации и энтропии в тритах является естественной и рентабельной. Она выкристаллизовалась в процессе многолетних исследований и наблюдений физической природы окружающего мира и необходима для дальнейшей эволюции информационно-виртуальной среды обитания человека.

Литература:

1. Основы теории информации и кодирования / Кузьмин И.В., Кедрус В.А. – К.: Вища шк., 1986. – 238 с.
2. Галенко В.П. Споживацький підхід до кількості інформації // Праці Луганського відділення МАІ. – 2003. – №1 (6). – С. 30–32.
3. Фомин С.В. Системы счисления. – М.: Наука, 1987. – 48 с.
4. Дьяконов В.П. MATLAB 6/6.1/6.5+Simulink. – М.: СОЛОМОН-Пресс, 2002. – 768 с.
5. Брусенцов Н.П. Пороговая реализация трехзначной логики электромагнитными средствами // Вычислительная техника и вопросы кибернетики. – М.: МГУ. – 1972. – Вып. 9. – С. 3–35.

К ВОПРОСУ УРОВНЕВОЙ ДИФФЕРЕНЦИАЦИИ ИЗУЧЕНИЯ КУРСА ИНФОРМАТИКИ В ВЫСШЕЙ ШКОЛЕ

Т.А. Холошня

г. Павлоград, Западнодонбасский частный институт экономики
и управления
m1962@list.ru

На протяжении ряда лет автором ведется наблюдение за уровнем подготовки студентов первых курсов в области информатики и информационных технологий.

Наблюдения проводились на основе анкетирования студентов первого курса Западнодонбасского института экономики и управления. Для наглядности результаты опросов представлены на диаграммах (рис. 1, 2), в которых отражены основные разделы школьного курса информатики, и по каждому разделу дисциплины представлено процентное отношение респондентов от их общего количества.

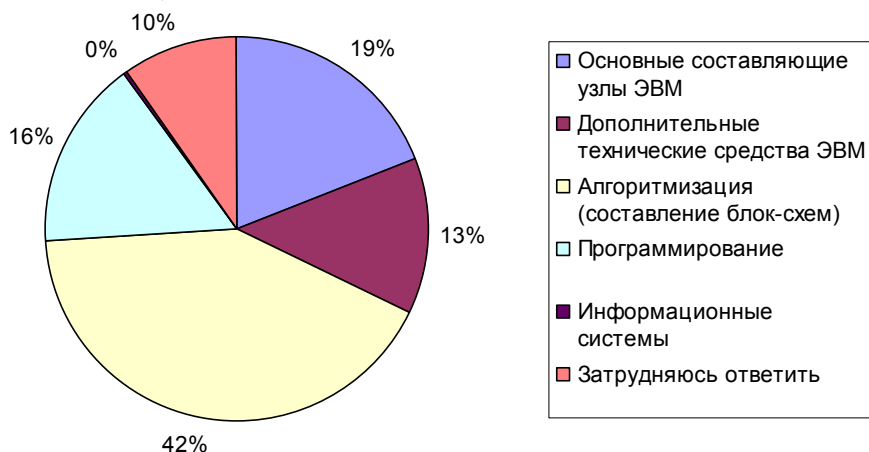


Рис. 1. Основные разделы общеобразовательного курса информатики (опрос 1999–2000 учебного года)

На рис. 3 представлена сводная диаграмма по данным рис. 1, 2. Исходя из сравнительной диаграммы, можно сделать выводы о существенном изменении содержательной части курса информатики в средней школе – больше внимания теперь уделяется изучению тем, посвященных информационным технологиям, а алгоритмизации значительно меньше, в тоже время программирование занимает достаточно весомую часть в общем объеме рассматриваемой дисциплины. Этот факт, бесспорно, является положительным, т.к. необходимость изменения структуры курса дисциплины затраги-

валась и нами в статье [1]. Кроме того, необходимо отметить пропорциональность соотношения разделов рассматриваемой дисциплины, что также является позитивным моментом.

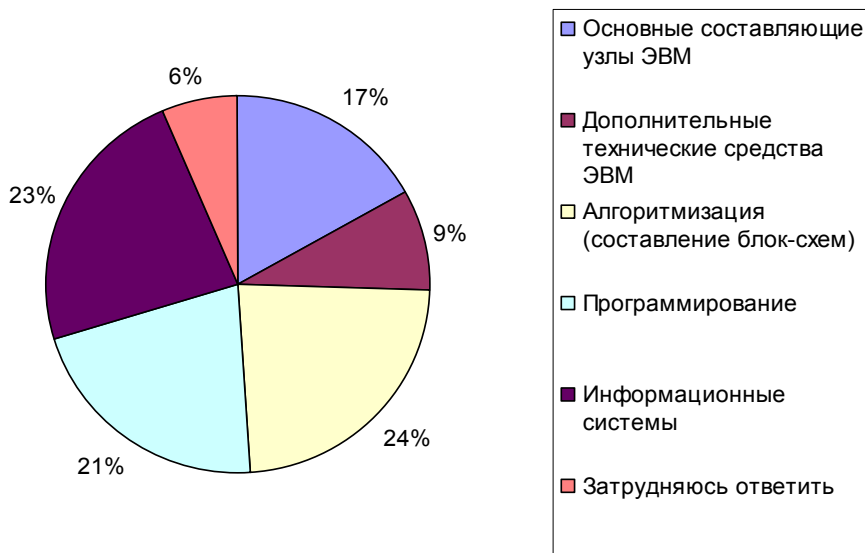


Рис. 2. Основные разделы общеобразовательного курса информатики (опрос 2004–2005 учебного года)

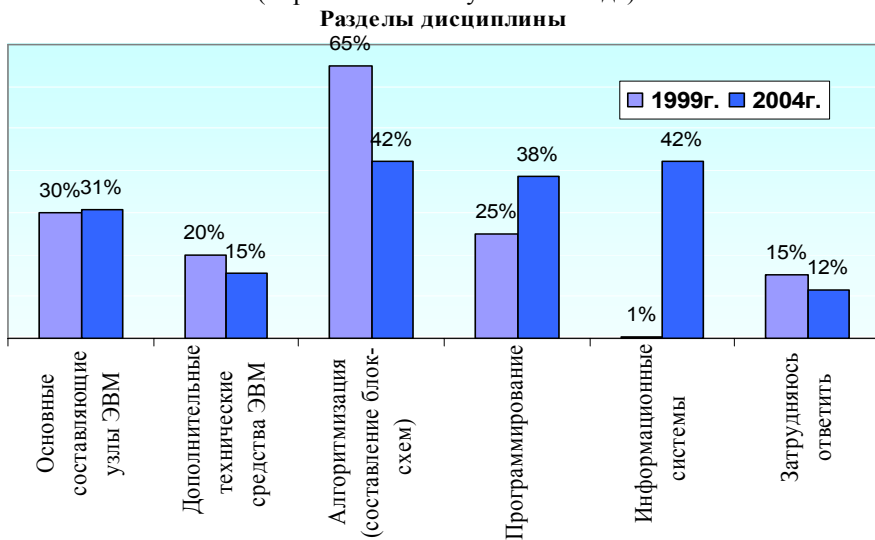


Рис. 3. Сравнительная диаграмма по данным 1999–2000 учебного года и 2004–2005 учебного года

Относительно тем, посвященных изучению аппаратной части ЭВМ («Основные составляющие узлы ЭВМ» и «Дополнительные технические средства ЭВМ»), произошедшие изменения незначительны. Таким образом, упоминание в [1] о необходимости увеличения количества часов на изучение технической составляющей ЭВМ остается актуальным. На современном этапе эта актуальность может быть обусловлена не только высоким процентом наличия компьютерной техники в школах (рис. 4), но и ее однотипностью.

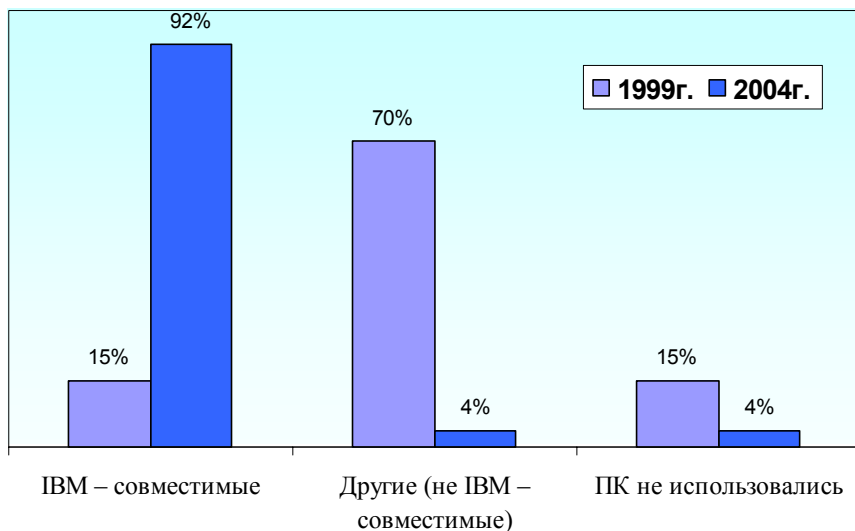


Рис. 4. Тип ЭВМ, используемый при изучении информатики в средней школе

Следует сказать и о том, что в результатах анкетирования на протяжении всего периода наблюдений, отражается факт наличия 10-15% студентов, у которых базовые знания по информатике и информационным технологиям практически отсутствуют, либо они незначительны (т.е. респонденты не изучали вышеуказанную дисциплину в системе общеобразовательной школы). Чаще всего такие студенты являются жителями сел или сельских районов. Не будем останавливаться на сложностях, с которыми сталкиваются такие студенты на занятиях по вышеуказанной дисциплине в вузе, но следует отметить, что этот факт является немаловажным. На наш взгляд, необходимость внедрения пропедевтического этапа изучения информатики на первом курсе вуза для указанной категории студентов обусловлено требованием времени.

В заключении хотелось бы выразить наше согласие с авторами [2], строящими освоение курса информатики по восходящему принципу («ди-

дактической спирали»). При этом имеется в виду не только наращивание числа изучаемых понятий, но и изучение одних и тех же понятий на разных уровнях детализации и научной строгости. Такой подход, на наш взгляд, наиболее оправдан на современном этапе.

Важным дидактическим принципом построения учебного курса является и принцип целостности [3]. На каком бы этапе ни изучалась информатика, в ней в том или ином объеме рассматриваются одни и те же объекты: информация, ЭВМ, алгоритмы, информационные модели, информационные технологии. Изучение различных тем предмета последовательно проходит через этапы, преследующие разные педагогические цели: дать представление; ввести понятие и определить его место в системе понятий; сформировать знания, основанные на понимании связей, структур и механизмов. И если первые два этапа, как вводные, рассматриваются в курсе информатики общеобразовательной школы, то третий, основополагающий и заключительный, необходимо тщательнейшим образом изучать в вузе, учитывая вышеуказанные педагогические принципы.

Литература:

1. Холошня Т.А. Вопросы преемственности в изучении курса информатики в средней и высшей школе // Теорія та практика трудової та професійної підготовки молоді. Сборник научных работ, выпуск 2. – Кривой Рог: Криворожский педагогический университет, 1999. – 177 с.
2. <http://ito.edu.ru/index.html>
3. Семакин И.Г. Об уровне дифференциации изучения информатики (ПГУ) // <http://ito.edu.ru/index.html>

ИЗЛОЖЕНИЕ РАЗДЕЛА «СИСТЕМОЛОГИЯ» ДИСЦИПЛИНЫ «ИНФОРМАТИКА И СИСТЕМОЛОГИЯ»

С.Н. Чаплыгина

г. Днепропетровск, Приднепровская государственная академия
строительства и архитектуры

Marinapn@ukr.net

Для специальности «Экология» в строительной академии на первом курсе читается дисциплина «Информатика и системология». Если относительно содержания дисциплины «Информатика» уже имеются установившиеся представления и набор изучаемых тем в зависимости от объема курса, то относительно системологии этого сказать нельзя. Более того, термины теория систем (системология) и системный анализ, несмотря на период их использования в несколько десятков лет, все еще не нашли общепринятого, стандартного истолкования. Видимо, причина этого заключается в возможности использования системного подхода практически в любой задаче.

Даже в определении самого понятия система можно обнаружить достаточно много вариантов, часть из которых базируется на глубоко философских подходах, а другие используют обыденные представления.

Исходя из сказанного видно, что отбор материала в раздел системология может быть произведен многими способами. Было бы неплохо знать, что конкретно имели в виду в Министерстве образования и науки относительно содержания этого раздела, но разъяснений к этой дисциплине или плана ее изложения академия не получала.

Когда этот раздел читался впервые, мы хотели каким-либо наглядным и несложным способом продемонстрировать примеры систем, их свойства и выполнение некоторых действий с системами.

Большая часть учебников по теории систем относится к абстрактному направлению, изложение в них ведется на основании теории множеств. Излагать этот раздел подобным образом на первом курсе строительного вуза не представляется возможным.

В нашем случае названный раздел был представлен тремя частями. В первой части давались основные определения теории систем, рассматривались свойства систем, их классификация, методы и модели описания систем. Особое внимание уделялось особенностям системного подхода и моделированию как методу системного анализа. Были рассмотрены различные виды моделей. При этом особое внимание уделялось математическим моделям, поскольку широкое распространение получило математическое моделирование на ЭВМ, называемое сейчас цифровым моделированием. Здесь можно проиллюстрировать применение при решении задач обеих наук – системологии и информатики. Получив систему уравнений, являющуюся математической моделью некоторого явления, нужно либо запрограммиро-

вать ее решение самому, либо использовать имеющееся программное обеспечение.

Во второй части основные элементы системологии излагались согласно Р. Эшби путем наглядного представления множества и совершения над ним преобразований. Это самый доступный способ продемонстрировать сущность происходящих в системе преобразований. Так были рассмотрены изменения в системе, повторные изменения, детерминированные системы, системы со входом, с обратной связью, изоморфные и гомоморфные системы, черный ящик и другие вопросы.

Третья часть была посвящена моделированию в экологии. Признание экосистемы предметом изучения экологии с необходимостью приводит к тому, что методологической основой этой науки становится системный подход, выдвигающий новые принципы в изучении объектов. Здесь следует обратить внимание на то, что в зависимости от характера огрубления реальной системы для одного и того же оригинала можно получить несколько различных моделей. Понятно, что в дисциплине, читаемой для экологов, должны быть рассмотрены примеры системного подхода, в частности моделирования, именно экологического смысла. Тем более, что в этой науке, действительно, практически в каждой задаче действует множество взаимосвязанных факторов. Однако реальные задачи экологии сложны и громоздки. Видимо, по этой причине в учебниках для экологов, таких как «Принципы моделирования та прогнозування в екології» В.В. Богобоющего и «Моделирования та прогнозування стану навколишнього середовища» П.І. Ковальчука, приводятся системы уравнений без их вывода и решения представлены без их получения. Это снижает интерес к восприятию дисциплины и не показывает главного – как же получить математические модели и решение задачи.

Для нас представлял особый интерес рассмотреть модель, сводящуюся к несложному дифференциальному уравнению. Вывод этого уравнения должен происходить наглядно и весь процесс должен быть понятным студентам. Желательно также, чтобы можно было получить решение этого уравнения и проанализировать его. Исходя из сказанного, был рассмотрен пример роста популяции, для него было выведено дифференциальное уравнение и получены начальные условия. Решение этого уравнения производилось известным студентам способом.

Последним в этой части был рассмотрен более сложный пример с учетом внешних и внутренних факторов, действующих на систему. Математическая модель сводилась к более сложному дифференциальному уравнению. Далее было продемонстрировано, как, вводя некоторые ограничения, можно последовательно упрощать математическую модель. В результате, как и ранее, было получено и проанализировано решение уравнения.

Такова была структура раздела системологии при первом чтении дисциплины.

Впоследствии было высказано мнение, что вторая часть раздела является абстрактной и не имеет непосредственной привязки к методам, используемым в экологических задачах. По этой причине она была заменена главой, посвященной методам математической статистики, а в третьей части добавлены примеры построения статистических моделей распространения загрязнений в реке и изменения радиоактивности воды в озере после сброса АЭС, а также двухкомпонентные модели.

В заключение следует отметить, что и этот вариант не является окончательным и при каждом новом чтении раздела приходится изменять его содержание и добавлять новые методы и примеры.

МАТЕМАТИЧНЕ І КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ЯК ФАКТОР ОПТИМІЗАЦІЇ НАВЧАЛЬНОГО І НАУКОВО-ДОСЛІДНОГО ПРОЦЕСІВ

Н.В. Шаповалова, Т.В. Ломасва

м. Київ, Національний педагогічний університет імені М.П. Драгоманова
shaponv@rambler.ru

Стрімкий розвиток сучасних наукових досліджень тісно пов'язаний з широким застосуванням математичного моделювання. Моделювання сьогодні стало важливим методом наукового пізнання в процесі досліджень. Цей метод використовується на всіх етапах наукового пізнання. В той же час комп'ютерні досліди з моделями об'єктів дозволяють, спираючись на потужність сучасних обчислювальних методів і технічних засобів інформатики, детально і глибоко вивчати об'єкти в достатній повноті, що є недоступним для чисто теоретичних підходів. Бурхливий розвиток математичного моделювання охоплює все нові і нові сфери – від розробки технічних систем і управління ними до аналізу складних економічних, соціальних та міжнародних процесів.

Для створення математичної моделі деякого об'єкту або явища необхідно виробити чіткий план дій, який умовно можна поділити на три етапи: 1) модель; 2) алгоритм; 3) програма.

На першому етапі обирається або будується «еквівалент» досліджуваного об'єкту, який в математичній формі відображає найважливіші його властивості – закони, яким він підпорядковується, зв'язки, які його притаманні та ін. Математична модель або її частини досліджуються теоретичними методами, що дає можливість отримати попередні знання про об'єкт.

На другому етапі розробляється або обирається алгоритм для реалізації моделі на комп'ютері. Модель зображається у формі, зручній для використання чисельних методів, визначається послідовність обчислювальних і логічних операцій, які потрібно виконати, щоб знайти шукані величини із заданою точністю. При цьому обчислювальні алгоритми не повинні спотворювати основні властивості моделі і, відповідно, вихідного об'єкту, мають бути економічними, такими, що легко і швидко адаптуються до особливостей розв'язуваних задач і можливостей комп'ютерів.

На третьому етапі створюється програми, які «переводять» модель і алгоритм на мову комп'ютера. При цьому створюється «електронний» еквівалент досліджуваного об'єкту, властивості, особливості і поведінку якого можна вивчати за допомогою комп'ютера. При необхідності процес моделювання уточнюється і покращується на всіх його етапах.

Більш детально процес побудови моделей може бути умовно розподілений на наступні етапи.

1. Конструювання моделі починається із словесно-змістовного опису

об'єкту чи явища. Крім відомостей загального характеру про природу об'єкту та цілі його дослідження, цей етап може містити також деякі припущення і гіпотези. На цьому етапі формулюється попередня модель.

2. Завершення ідеалізації об'єкту. Відкидаються всі фактори і ефекти, які вважаються не досить суттєвими для його дослідження. Якщо можливо ідеалізуючі припущення записуються в математичній формі з тим, щоб їхня справедливність піддавалась кількісному контролю.

3. Формулювання або вибір закону, принципу, закономірності, якому підпорядковується об'єкт і його запис в математичній формі. При необхідності використовуються додаткові відомості про об'єкт, які записуються математично, враховуючи, наприклад, геометричні, фізичні, хімічні, біологічні, економічні, соціальні та інші закономірності та особливості.

4. Задаються відомості про початковий стан об'єкту для визначення поведінки об'єкта. Формулюється мета дослідження моделі, наприклад, знайти закон, дослідити закономірні зміни, розвитку, визначити вимоги до даної конструкції та ін.

5. Побудована модель вивчається за допомогою всіх доступних науковцю-досліднику методів, причому із взаємною перевіркою різноманітних підходів. Більшість моделей не підлягає чисто теоретичному аналізу, і тому необхідно широко використовувати обчислювальні методи, це особливо важливо при вивченні нелінійних об'єктів, оскільки їх якісна поведінка наперед, як правило, невідома.

6. В результаті дослідження моделі не лише досягається поставлена мета, а й встановлюються всіма можливими способами її адекватність, тобто відповідність об'єкту та сформульованим припущенням і гіпотезам. Неадекватна модель може дати результат, що відрізняється від істинного, і повинна бути відкинута або відповідним чином модифікована.

Універсальність математичних моделей відображає єдність оточуючого нас світу і способи його зображення. Тому методи і результати, розроблені і накопичені при математичному моделюванні одних явищ, відносно легко «за аналогією», можуть бути перенесені на значний клас інших процесів. Одним із прикладів універсальності математичних моделей є параболічні рівняння. Вони описують широке коло процесів цілком різної природи, наприклад, динаміка розподілу влади в ієрархії. Найчастіше параболічні рівняння пов'язують із хаотичними невпорядкованими явищами (теплообмін, дифузія та ін.), однак вони застосовуються і до багатьох процесів, які розглядаються як детерміновані (рух ґрунтових вод, фільтрація газу та ін.).

Особливе значення метод «аналогії» відіграє особливу роль при математичному моделюванні важкоформалізованих об'єктів, для яких фундаментальні закони, варіаційні принципи та інші загальні і математично строгі твердження або невідомі, або взагалі не існують. До таких об'єктів відносяться, наприклад, системи із значним залученням людського фактору, зокрема економічні і політичні системи.

Однією з важливих механіко-економічних аналогій є аналогія між рівновагою матеріальної частки в потенціальному полі зовнішніх сил і вибором оптимального плану виробництва.

Розглянемо найбільш прості математичні моделі економічної рівноваги. Будь-який учасник ринкового економічного процесу діє у відповідності зі своїми індивідуальними інтересами – отримання прибутку, покращення умов праці, мінімізація ризику, економія ресурсів та ін. Найпростіший варіант такої системи – економіка з досконалою конкуренцією, коли кожен суб'єкт економічно мізерний і не чинить безпосередній вплив на рівень виробництва, ціни, зарплату та інші макропоказники. В той же час роз'єднані дії економічних агентів може складатися через існуючі в системі відносини купівлі-продажу в сукупну узгоджену картину дій працедавців і найманих працівників, фінансистів і вкладників та ін.

Якщо в результаті такої колективної взаємодії загальне виробництво товарів і послуг в системі узгоджене з загальним попитом на них, то такий стан економіки називається *рівновагою*, а ціни, що встановлюються при цьому – *рівноваженими ринковими цінами*. Баланс між попитом і пропозицією має місце, за припущенням, не при довільних, а саме при цих ринкових цінах, що означає, зокрема, платоспроможність попиту.

Одна з найважливіших задач економічної науки – визначення умов рівноваги економіки, в тому числі врівноважених ринкових цін. Найбільш прості математичні моделі економічної рівноваги будуються за наступних припущень:

- 1) досконала ринкова конкуренція, що передбачає відсутність як великих виробничих корпорацій (і тим більше, монополій), так і об'єднань робітників, які можуть диктувати свої умови для всієї системи;
- 2) незмінність виробничих можливостей системи: обладнання, виробничі приміщення, технологій не змінюється з часом;
- 3) незмінні у часі економічні інтереси партнерів: підприємці не намагаються збільшити свій прибуток, робітники – заробітну платню, інвесторів задовольняють проценти, які надходять за цінними паперами тощо.

Моделі, які відповідають цим припущенням, описують досить окремих випадок «застиглої» у часі ідеальної ринкової економіки. Однак, вони дають відповідь на питання про можливість існування економічної рівноваги, що формується з ринкового хаосу, і крім того, пов'язують між собою основні показники макроекономічної системи.

Однією з головних задач теорії економічного зростання є знаходження оптимальних способів розподілу виробленого продукту на споживчу і накопичувану частини.

Оскільки математичне моделювання є методологією, то воно не підміняє собою математику, фізику, біологію та інші наукові дисципліни, а виконує щодо них синтезуючу роль.

Розв'язуючи проблеми інформаційного суспільства, було б помилкою

розраховувати лише на потужність комп'ютерів та інших засобів інформатики. В цій сфері методологічним імперативом є постійне вдосконалення всіх етапів математичного моделювання та їх впровадження в сучасні інформаційно-моделюючі системи. Лише його виконання дає можливість виробляти необхідну нам високотехнологічну, конкурентноспроможну і різноманітну матеріальну і інтелектуальну продукцію.

Одна з фундаментальних властивостей природних, технологічних, багатьох економічних і соціальних об'єктів – симетрія (подібність, повторюваність, репродуктивність) знаходить своє відображення в їх математичних моделях. Наявність будь-якого виду симетрії у вивчаємого явища означає більш зручний аналіз даного об'єкту в порівнянні з його менш симетричним аналогом. На цьому засновані широко застосовувані методи спрощення математичних моделей і, як наслідок, методи спрощення їх аналізу. Вони полягають в зниженні порядку системи рівнянь, що утворюють модель, в зменшенні кількості змінних, від яких залежать шукані величини, або кількості постійних параметрів, які визначають процес тощо.

Найбільш розроблений і широко застосовуваний підхід, що використовує подібність моделей, заснований на так званому *інваріантно-груповому методі* дослідження диференціальних рівнянь. Більшість диференціальних рівнянь, які є складовою частиною математичних моделей багатьох явищ, залишаються незмінними (*інваріантними*) при деяких перетвореннях незалежних змінних і шуканих функцій, що входять до них.

Розглянемо перетворення змінних і різні їх комбінації та узагальнення, які відносяться до класу так званих точкових перетворень або перетворень Лі (похідні від величин не перетворюються). Нехай, до того ж, вони утворюють групу, тоді диференціальне рівняння, що допускає групу Лі, може бути спрощене: або понижується його порядок, або зменшується число незалежних змінних, що визначають шукані функції. Однією з цілей групового аналізу є визначення всіх груп перетворень, які допускають дане рівняння або система рівнянь, і відповідних цим групам так званих *інваріантних розв'язків* рівнянь. З точки зору дослідження математичних моделей важливі перш за все не складна і громіздка процедура групового аналізу, а його кінцеві висновки стосовно розглядуваного рівняння.

Обчислювальний алгоритм повинен задовольняти досить жорстким і суперечливим вимогам, структура алгоритму повинна бути достатньо простою і враховувати архітектуру обчислювальної системи та можливості обробки інформації комп'ютером і при цьому містити вичерпну інформацію про кількісні і якісні властивості досліджуваного об'єкту.

Проблеми чисельного моделювання не зникають з появою більш потужних комп'ютерів, тому розробка ефективних обчислювальних алгоритмів та динамічних моделей завжди залишається однією з ключових задач математичного моделювання.

Математичні моделі завжди містять недостатньо розкриті характерис-

тики досліджуваних об'єктів, що заважає досягненню абсолютної точності і адекватності даних моделей реальним процесам, але не зменшує їхньої наукової цінності як інструментів аналізу, спостереження, порівняння і прогнозування різного роду явищ у всіх сферах суспільного життя.

Ускладнення самих досліджуваних об'єктів стимулює науковців до розробки та вдосконалення математичних моделей, які застосовуються для їх аналізу. З плином часу постає необхідність впроваджувати більш комплексні десементовані синергетичні моделі реальної дійсності, побудовані на основі комбінування і синхронізації суспільних процесів в ході наукового пізнання, що є найбільш актуальним завданням сучасної науки.

Органічне поєднання і взаємозв'язок математичного і комп'ютерного моделювання в підготовці студентів є необхідним елементом навчального процесу і дослідницької діяльності. Набуття студентами вищих навчальних закладів вмінь самостійно розробляти моделі для застосування у навчальному та виробничому процесах, розробляти методику проведення занять з використанням комп'ютерного моделювання, створювати нові моделі та вдосконалювати існуючі в своїй дослідницькій діяльності є невід'ємним елементом освітньої підготовки майбутніх фахівців.

Література:

1. Самарский А.А., Михайлов А.П. Математическое моделирование: Идеи. Методы. Примеры. – М.: Наука. Физматлит, 1997. – 320 с.
2. Овсянников Л.В. Групповой анализ дифференциальных уравнений. – М.: Наука, 1978. – 400 с.
3. Жалдак М.І. Педагогічний потенціал комп'ютерно-орієнтованих систем навчання математики // Комп'ютерно-орієнтовані системи навчання: Збірник наукових праць. – К.: НПУ імені М.П. Драгоманова. – Випуск 7. – 2003. – С. 3–16.

ВИКОРИСТАННЯ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ ПРИ ВИКЛАДАННІ КУРСУ «МОДЕЛЮВАННЯ ЕКОНОМІЧНОЇ ДИНАМІКИ»

Л.Е. Гризун

м. Харків, Харківський національний педагогічний університет

ім. Г.С. Сковороди

Lgr@mail.ru

Моделювання розглядається сьогодні як невід'ємна частина будь-якої цілеспрямованої діяльності, оскільки ціль можливо розуміти як модель деякого бажаного стану системи, якого потрібно досягти. Під моделлю деякої системи або об'єкта, явища розуміється штучна система або об'єкт, що в певних умовах може замінити оригінал шляхом відтворення його властивостей і характеристик, які цікавлять дослідника, якщо цей замінювач дає істотні переваги зручності дослідження (доступність випробувань, легкість оперування, наочність тощо). З означення випливає, що модель є досить широким поняттям. Моделі можуть бути не тільки матеріальними, а й ідеальними, абстрактними.

Як метод наукового пізнання моделювання набуває особливого значення при вивченні явищ та закономірностей різноманітної природи та походження, дослідження яких в реальних умовах стикається із труднощами, пов'язаними із складністю реального об'єкта дослідження, тривалістю у часі його життєдіяльності, прихованими специфічними властивостями тощо. Завдяки своєму пізнавальному потенціалу, в сучасних умовах моделювання стає одним із головних інструментів дослідження динамічних економічних систем.

Економічна динаміка – відносно новий напрям в економічній теорії, який охоплює різноманітні концепції і парадигми пояснення складних процесів та явищ, що виникають в соціально-економічних системах на макрота мікрорівнях. Економічна динаміка, на відміну від статички, вивчає поведінку економічних систем та розвиток процесів у часі. При динамічному підході дослідника цікавить не один, а цілий спектр станів системи в продовж певного відрізка часу. Предметом дослідження, таким чином, виявляється поведінка динамічних економічних систем, її характер та стабільність, економіко-математичні методи і моделі, що дозволяють описувати і досліджувати складні явища і процеси в динамічних соціально-економічних системах. Такого роду дослідження дозволяють не тільки визначити перспективи та можливі сценарії розвитку досліджуваного об'єкта, а й розробити комплекс адаптивних впливів на нього, виявити можливі резерви, скоригувати політику, що реалізується в реальній економічній системі.

В своїх дослідженнях економічна динаміка спирається на методичний апарат методів математичного аналізу, диференціального та варіаційного

числення, графічних методів, теорії катастроф тощо.

Особливості моделювання економічної динаміки впливають із основних властивостей складних економічних систем та їх характеристик [1]. До головних особливостей математичного моделювання економічної динаміки можна віднести такі аспекти:

1) для практичного використання розроблених математичних моделей необхідним є їхнє наповнення конкретною і якісною інформацією, що висуває до неї вимоги коректності, точності, повноти, співставленості, оперативності тощо, що загострює проблему якості економічних вимірювань;

2) в залежності від модельованих об'єктів та призначення моделей вихідна інформація має сутнісно різноманітний характер і походження: це економічні спостереження, стосовно минулого розвитку та сучасного стану об'єктів, та прогнозовані дані про майбутній розвиток, які також можуть бути отримані за допомогою моделювання;

3) моделювання в економіці має спиратися на масові дані, оскільки більшість економічних процесів характеризуються закономірностями, які можуть бути виявленими тільки на основі аналізу значної кількості спостережень;

4) в процесі економічного моделювання виникає взаємодія первісних економічних показників та нових, одержаних в результаті моделювання, що актуалізує проблеми коригування первісних показників, їх зворотного зв'язку із новими, їх взаємопов'язаного аналізу.

У зв'язку з цим особливо актуальним в сучасних умовах є комп'ютерне моделювання економічної динаміки, результатом якого є віртуальний аналог, що має іншу фізичну природу, але описується такими ж математичними співвідношеннями, що й оригінал. До переваг комп'ютерного моделювання в цій галузі перш за все слід віднести можливість проведення віртуальних експериментів, які у реальному житті потребують значних матеріальних та часових затрат, а також забезпечення більш глибокого вивчення сутності економічних процесів за рахунок специфічних можливостей комп'ютерних технологій.

Значення комп'ютерного моделювання економічної динаміки посилюється ще й тим, що дослідження поведінки в динаміці складних економічних систем є предметом навчальних курсів, які включено до програм підготовки студентів економічних спеціальностей багатьох вищих навчальних закладів. Ця обставина спричиняє необхідність визначення дидактичних можливостей комп'ютерних навчальних моделей економічної динаміки з точки зору забезпечення дослідницько-пошукового характеру навчання; розробки методичних прийомів їх створення та використання з метою формування і розвитку якості знань та пізнавальної активності студентів.

Можна виділити основні ознаки створення навчальної економіко-математичної моделі за допомогою комп'ютера, основними серед яких є: наперед сформульована та обмежена множина параметрів (показників) за-

дачі; конкретно окреслене коло питань, що підлягають вивченню; обмежена кількість відомих економічних законів, які мають бути використані для побудови даної моделі; відомі прийоми діяльності із засобами комп'ютерних технологій, обраними для побудови та дослідження даної моделі.

Отже, роботу з розробки конкретної комп'ютерної моделі економічної динаміки доцільно, на нашу думку, проводити за такою дидактичною схемою.

1. *Побудова економіко-математичної моделі*: вивчення теоретичної сутності та особливостей економічного явища, що моделюється; визначення параметрів, які змінюються і залишаються постійними; вибір функцій та рівнянь, що описують дане економічне явище, обґрунтування цього вибору; теоретичний аналіз побудованої математичної моделі з точки зору її адекватності, меж використання та аналіз її реакції на зміну вихідних параметрів.

2. *Вибір програмного забезпечення*, необхідного для побудови моделі та роботи з нею.

3. *Побудова комп'ютерної моделі*. В залежності від обраного програмного забезпечення, моделлю може бути існуюча програма, що дозволяє на основі побудованої математичної моделі відтворювати економічний процес чи явище в динаміці і пристосована для проведення дослідження. Проте найчастіше побудова комп'ютерної моделі на основі вже створеної математичної моделі відбувається в середовищі універсальних або професійних комп'ютерних пакетів, які дозволяють на основі введених вихідних даних та взаємозалежностей змінних відобразити поведінку економічної системи (явища) та мають зручні вбудовані можливості для маніпулювання параметрами системи з метою проведення експериментів.

4. *Робота з готовою комп'ютерною моделлю*, спрямована на дослідження поведінки економічної системи (явища) в різних умовах, одержання економічно значущих висновків щодо динаміки її розвитку, вивчення адекватності комп'ютерної моделі та меж її використання тощо.

Наведена дидактична схема вимагає від викладача розробки відповідного методичного забезпечення: спеціально розроблених завдань, комплексу запитань, які спонукають студента до детального аналізу економічної системи, причин її динамічних змін; теоретично обґрунтованої побудови математичної моделі; втілення її в адекватний комп'ютерний аналог; проведення віртуальних експериментів з досліджуваною економічною системою, явищем.

Запропоновану дидактичну схему було використано при викладанні курсу “Моделювання економічної динаміки” для студентів п'ятого курсу економічного факультету ХНПУ ім. Г.С. Сковороди. Оскільки названий курс пропонується студентам різних спеціалізацій, для побудови більшості моделей було обрано універсальний програмний засіб – табличний процесор Excel з його вбудованими можливостями автоматичних розрахунків, динамічного відтворення табличної інформації у графічному вигляді, вико-

ристання візуальних програмованих об'єктів для оперативного керування параметрами моделі.

Слід зазначити, що розробка комп'ютерних моделей в середовищі Excel (динамічна модель Леонтьєва, дискретна “павутинна” модель попиту і пропозиції, модель розвитку перехідної економіки Міхалевіча, моделі циклічних коливань в економічному розвитку, модель валютної паніки та ін.) дозволила студентам поглибити знання про сутність динамічних економічних систем, механізм роботи економічних законів, формування економічних показників тощо. Проведені з побудованими моделями експерименти, завдяки потужним характеристикам Excel, надали можливість студентам одержати певні дослідницькі навички, вміння постановки і розв'язку задач аналізу, прогнозування, прийняття рішення тощо; дозволили розповсюдити дію деяких моделей на процеси іншої природи. Все це сприяло підвищенню професійних аналітичних та дослідницьких якостей випускників, зростанню їхньої пізнавальної активності.

Таким чином, комп'ютерне моделювання дозволяє значно збагатити дослідження динаміки складних економічних систем; поглибити розуміння сутності економічних закономірностей і явищ; надати навчання творчодослідницького характеру; стимулювати розвиток абстрактно-логічного та аналітичного мислення студентів; формувати їхні навички використання комп'ютерних технологій для організації і проведення віртуальних досліджень.

Література:

1. Клебанов Т.С., Дубровинова Н.А., Полякова О.Ю. и др. Моделирование экономической динамики: Учебное пособие. – Х.: Издательский дом “ИНЖЭК”, 2004. – С.13-17
2. Агапова Т.Е., Берхенс Д., Курран Д. Динамические системы в экономике. – Донецк: ДонГУ, 2000. – 140 с.
3. Горчаков А.В. Математические модели в экономике. – М.: Наука, 1994.
4. Долголаптев И. Работа в Excel 5 на примерах. – 1997. – 256 с.
5. Жак С.В. Математические модели менеджмента и маркетинга. – Ростов на Дону: ЛаПо, 1997. – 157 с.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕГРОВАНОГО КУРСУ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

Ю.С. Матвієнко

м. Полтава, Полтавський державний педагогічний університет
імені В.Г. Короленка
uuri@rambler.ru

Однією з тенденцій розвитку системи шкільної освіти є його профілізація. Очевидно, що навчання учнів математиці в економічних, хіміко-біологічних, гуманітарних, фізико-математичних та інших класах вимагає створення спеціальних умов для забезпечення прикладної математичної підготовки. Це виражається перш за все в тому, що випускників шкіл треба навчити грамотно досліджувати різні реальні ситуації математичними засобами, озброїти їх навичками математичного моделювання. Моделювання є невід'ємним компонентом загальнолюдської культури та методом пізнання навколишнього світу, природи та суспільства.

На нашу думку, доцільним є включення курсу математичного моделювання в навчальні плани загальноосвітніх навчальних закладів відповідного профілю та у курс підготовки вчителів інформатики у вищих педагогічних навчальних закладах. В умовах інформатизації навчання доцільне вивчення методів математичного моделювання в поєднанні з використанням сучасної комп'ютерної техніки, що в повній мірі може бути реалізоване в межах інтегрованого курсу математики та інформатики.

Такий інтегрований курс покликаний продемонструвати широке використання математичного апарату для вивчення процесів та явищ реальної дійсності. Зміст цього предмету складають питання: поняття математичного моделювання та класифікація моделей; методи обчислювальної математики, що використовуються в математичному моделюванні; класичні екстремальні задачі; методи розв'язання транспортних задач; задачі нелінійного програмування; математичне моделювання в біології та екології; математичні моделі у фізиці; математичні моделі в економіці; математичні моделі в гуманітарних науках; комп'ютерна обробка математичних моделей і т.д.

В наш час комп'ютерна промисловість пропонує низку різноманітних засобів моделювання, що дозволяють моделювати складні динамічні системи та проводити з ними експерименти. Найбільш повне дослідження загальносистемних проблем отримується в результаті моделювання об'єктів за допомогою технологій, реалізованих в спеціалізованих обчислювальних пакетах візуального моделювання. На сьогоднішній день існує велика кількість пакетів візуального моделювання. В них користувачу надається можливість описувати систему, що моделюється, переважно у візуальній формі, наприклад, графічно представляючи як структуру системи, так і її поведінку. Сучасні універсальні пакети візуального моделювання мають низку за-

гальних властивостей, все ж їх можна розділити на три основні групи:

1) пакети, що використовують мову блочного моделювання (підсистема Simulink пакету MATLAB [1], пакет EASY5 [2], підсистема SystemBuild пакету *MATRIX*_x [3], VisSim [4]);

2) пакети, що використовують мову фізичного моделювання (Dymola [5], Omola і OmSim [6], Smile [7], Modelica [8]);

3) пакети, орієнтовані на використання схеми гібридного автомату (Shift [9], Model Vision Studium [10]).

На наш погляд, в якості мови моделювання доцільно використовувати Уніфіковану Мову Моделювання (UML – Unified Modeling Language), створену протягом останніх декількох років при підтримці консорціуму Object Managing Group (OMG) [11] спеціалістами фірми Rational Software Corporation, а в якості пакету візуального моделювання – Simulink. Вибір UML в якості мови моделювання обумовлений тим, що процес створення моделі об'єкту часто є справою творчою, особливо коли в якості опису об'єкту присутній опис на власній мові об'єкту та математичний опис його функцій. Отже, при створенні моделі зручно використовувати деяку мову, що дозволить створити візуальну модель об'єкту, описати його структуру, взаємов'язи його складових. UML дозволяє документувати об'єктно-орієнтовані системи, виконувати їх аналіз та проектування, на ній можна змістовно описувати класи, об'єкти та компоненти в різних предметних областях. Основні компоненти UML включають опис семантики (складових одиниць мови) об'єкту, його графічну нотацію та опис додаткових понять, що дозволяють розширити зміст основних понять мови.

Використовуючи UML, можна будувати моделі з базових блоків, таких як класи, інтерфейси, кооперації, компоненти, вузли залежності, узагальнення та асоціації. В процесі розробки система представляється у вигляді об'єднання кількох проекцій, кожна з яких описує певний аспект системи, що розробляється, а разом вони визначають систему в усій її повноті. Ці проекції представляються в UML діаграмами. За допомогою діаграм можна візуалізувати систему з різних точок зору. Оскільки складне ціле не можна зрозуміти, дивлячись на нього лише з одного боку, в UML визначено багато різних діаграм, які дозволяють акцентуватися на різних аспектах системи, що моделюється. Вдалі діаграми полегшують розуміння моделі. Обміркований, продуманий вибір діаграм при моделюванні системи дозволяє задавати правильні питання про неї, допомагає грамотній постановці задачі та прояснює наслідки рішень, що приймаються.

Задачі для практичного розв'язання під час вивчення моделювання в школі потрібно відбирати за такими критеріями:

– кожна задача повинна належати певній предметній області (фізика, екологія та ін.);

– достатня простота (тому що набагато корисніше для учня виконати до кінця нескладне завдання, ніж недовиконати складне).

Передбачається, що учень в якості завдання в рамках вивчення математичного моделювання отримуватиме опис деякої реальної системи звичайною мовою. В процесі виконання завдання учень повинен реалізувати вихідний опис об'єкту спочатку в поняттях UML, а потім, ґрунтуючись на створеній візуальній моделі, в підсистемі Simulink пакету MATLAB.

Можливе також використання одного з існуючих CASE-засобів для роботи з мовою графічного моделювання UML. Серед них слід виділити:

1. *Umbrello UML Modeller*. CASE-засіб для автоматизації етапів аналізу та проектування, а також для генерації коду на мовах C++, Java та PHP. Для мови C++ підтримується відтворення моделі за текстом програми.

2. *Dia*. GPL-аналог комерційної програми для ділової графіки Visio.

3. *Dia2code*. Дана програма є утилітою для перетворення UML діаграм, створених в редакторі Dia, в код мов Ada, C, C++, Java, PHP, Python та SQL.

4. *Dachshund*. Case-засіб для автоматизації етапів аналізу та проектування, а також для генерації кодів та випуску проектної документації.

Далі учень повинен провести обчислювальний експеримент з моделлю та отримані результати інтерпретувати в рамках предметної області системи, що вивчається.

Література:

1. MathWorks, Inc, <http://www.mathworks.com>
2. Boeing, <http://www.boing.com/assocproducts/easy5>
3. Integrated Systems, Inc., <http://www.isi.com>
4. Visual Solution, <http://www.vissim.com>
5. Dynasim, <http://www.dynasim.se>
6. Lund University, <http://www.control.lth.se/~cace/omsim.html>
7. Berlin Technical University, <http://www.first.gnd.de/smile>
8. The Modelica Design Group, <http://www.dynasim.se/modelica>
9. California, <http://www/path/berkeley.edu/shift>
10. www.xjtek.com
11. www.omg.com
12. Фаулер К. Скотт. UML в кратком изложении. Применение стандартного языка объектного моделирования. – М.: Мир, 1999.
13. Бешенков С.А., Ракитина Е.А. Моделирование и формализация: Методическое пособие. – М.: Лаборатория базовых знаний, 2002.
14. Шестаков А.П. Компьютерное математическое моделирование: Лекция 7 // Информатика: Прил. к газ. «Первое сентября». – 2002. – №40.
15. Матвієнко Ю.С. Основні аспекти вивчення математичного моделювання в школі з використанням комп'ютерної техніки // Особистісно орієнтоване навчання математики: сьогодення і перспективи. Матеріали II Всеукраїнської науково-практичної конференції. – Полтава: АСМІ, 2005. – С. 205.

ИЗУЧЕНИЕ СТРУКТУР ДАННЫХ В КУРСЕ ПРОГРАММИРОВАНИЯ КОМПЬЮТЕРНЫХ СПЕЦИАЛЬНОСТЕЙ ВЫСШЕЙ ШКОЛЫ

И.А. Кривель¹, А.Н. Моргун²

¹ г. Черкассы, Факультет рыночных, информационных и инновационных технологий Киевского национального университета технологий и дизайна

² г. Черкассы, Черкасский институт пожарной безопасности имени Героев Чернобыля

a_n_m@rambler.ru

Учитывая современные тенденции и возрастающую роль самостоятельной работы при изучении программирования студентами компьютерных специальностей вузов, необходимо критически подойти к сложившейся методике преподавания. Здесь, в первую очередь, требуют усовершенствования содержание и стиль изложения тех вопросов, которые связаны с освоением методов обработки структур данных.

К сожалению, не всем известно, что в программировании далеко не столь важно умение правильно записывать операторы языка, сколько искусство расставлять их в правильной последовательности. Умение решить задачу есть ни что иное, как оптимальная организация её исходных данных и результатов, а также чёткое представление пути достижения её решения в виде последовательности отдельных, относительно самостоятельных шагов. Иными словами, здесь мы должны повторить ставшую классической фразу “Алгоритмы + структуры данных = программы” [1].

Материалы, посвящённые основами алгоритмизации и программирования обработки структур данных, можно найти в книге [2]. В этой книге делается очередной шаг в программировании: от обычных вычислений, рассмотренных в [3], к обработке простейших упорядоченных совокупностей данных, реализованных в стандартном Паскале. Здесь речь идёт, прежде всего, о массивах различной размерности, строках, множествах и текстовых файлах.

В первую очередь читатель получает основательные знания об особенностях строения перечисленных структур. А затем в книге самое серьёзное внимание уделяется типичным методам и задачам их обработки с использованием хорошо подобранных примеров. При этом широко используются такие эффективные средства, как модульное программирование, динамическая память и рекурсия.

Не обойдены вниманием и общие проблемы, в том числе касающиеся терминологии и классификации структур данных. Авторы книги убеждены в исключительной важности этих вопросов. То, что преподносится читателю, должно систематизировать его знания, а не вводить в заблуждение. В противном случае, мы с вами будем считать обычные массивы “списками

последовательного хранения”. К сожалению, приведённый пример терминологического казуса нередок даже в преподавательской среде.

В основе обучения программированию обработки структур данных лежит всемерное использование аппарата процедур и функций. Это полезно как само по себе, так и в плане дальнейшего перехода к объектному программированию. При этом большое внимание уделяется решению так называемых комбинированных задач, предполагающих применение типичных элементарных алгоритмов при построении более сложных. Благодаря этому, во многом удаётся уйти от такого уродливого явления, как “хаотическое программирование”.

Авторы считают практически бессмысленным использование блок-схем на этапе алгоритмизации процессов решения задач. Громоздкость, необозримость и трудоёмкость построения делают иллюзорной их “наглядность” даже на уровне школьных алгоритмов средней сложности. В современных условиях применения персональных компьютеров процессы алгоритмизации практически неотделимы от программирования. Согласитесь, что программист, с упоением рисуемый блок-схемы, сидя за клавиатурой, выглядит, мягко выражаясь, несерьёзно. Справедливости ради, отметим всё же целесообразность использования блок-схем для повышения наглядности при демонстрации элементарных алгоритмов в младших классах.

Крупнейшим исследователем в области структур данных по праву считается Никлаус Вирт, известный в кругу специалистов как основатель и руководитель творческой школы программирования, которая находится в Цюрихе (Швейцария). Именно с этой школой связывается появление языка Паскаль, а также понимание того, что алгоритмы обработки данных существенным образом зависят от их строения, расположения и порядка, то есть от структуры. Таким образом, становится понятной целесообразность выбора средств языка Паскаль для изучения как основных структур данных, так и методов их обработки.

Изучая основы обработки структур данных, необходимо исходить из следующих положений.

1. Иерархический принцип считается основным при образовании тех или иных структур данных. Это значит, что структуры данных отличаются от данных, которые можно назвать элементарными. В общем случае, структуры данных состояются из компонентов, которые сами могут считаться структурами, но более простыми. На нижнем уровне этой иерархии находятся элементарные (атомарные) данные, не подлежащие дальнейшему расчленению.

2. В определённом смысле ту или иную структуру данных можно отождествлять с некоторой величиной. При этом оказывается возможным говорить о множестве значений величины, имея в виду разнообразие элементарных значений в её составе. Можно говорить также и об объёме отдельного значения величины, подразумевая под этим некий суммарный объём па-

мяти, требуемый для хранения всех элементарных значений в её составе.

3. Основной характеристикой любой величины (структуры данных) является её тип. В общем случае, понятие типа включает множество возможных значений величины, набор допустимых операций с ними, способ представления отдельного значения величины в памяти вычислительной машины, а также его объём. Поскольку иерархический принцип предполагает существование как структурированных, так и элементарных величин, то мы просто обязаны использовать соответствующие понятия структурированных и элементарных (простых) типов. Если все составляющие структурированной величины имеют один и тот же тип, то этот тип называют базовым.

4. Количество возможных значений величины, принадлежащей некоторому типу, называется мощностью типа этой величины. В общем случае, мощность типа T определяется некоторой функцией $Card(T)$. Объём значения типа T определяется функцией $SizeOf(T)$.

5. Различают статические и динамические структуры данных. Основным признаком любой статической структуры данных является неизменность занимаемого ею объёма памяти в процессе исполнения программы. В противном случае структуру данных называют динамической.

Стандартные типы в языке Паскаль, как правило, являются простыми. К ним относятся все порядковые и вещественные типы. К простым относятся также такие нестандартные типы языка Паскаль, как интервальные и перчисляемые.

К основным структурированным типам языка Паскаль относятся тип-массив, строковый тип, тип-множество, а также файловый текстовый тип. При этом массивы, строки и множества мы относим к статическим структурам данных, а текстовые файлы – к динамическим.

Как уже указывалось, ту или иную структуру данных можно отождествлять с некоторой величиной. Рассмотрим этот вопрос несколько подробнее.

С теоретико-информационных позиций формулировка любой задачи представляет собой некоторое исходное сообщение, содержащее информацию, которую мы называем результатом её решения. Пока задача не решена, эта информация является недоступной. Процесс решения задачи состоит в том, что разрабатывается некоторое правило интерпретации исходного сообщения, позволяющее извлечь из него ранее недоступную информацию.

Иными словами, разработанное в процессе решения задачи правило интерпретации, позволяет преобразовать исходное сообщение в некоторое конечное. Конечное сообщение содержит результат решения задачи в явном виде. Таким образом, ранее недоступная информация, содержавшаяся в исходном сообщении, становится доступной в конечном.

Развивая эту мысль, можно утверждать, что алгоритмы представляют собой определённые правила интерпретации, которые следует применять к исходным сообщениям, чтобы получать содержащуюся в них информацию.

При этом исходным сообщением, содержащим недоступную информацию, является совокупность исходных данных алгоритма. В результате исполнения алгоритма получаем некоторое конечное сообщение, представляющее собой совокупность результатов. Таким образом, информация конечного сообщения оказывается вполне доступной.

Учитывая само существование исходных данных и результатов, мы имеем полное право утверждать, что алгоритмы, как правила интерпретации сообщений, имеют вполне определённую специфику. Эта специфика состоит в том, что, как исходные, так и вырабатываемые алгоритмами конечные сообщения представляют собой совокупности некоторых величин.

Будем называть величиной любой объект алгоритма, несущий информацию об определённом значении. Величины в алгоритме выступают основными персонажами процесса исполнения его команд. Этот процесс может сопровождаться преобразованием формы представления величин, а также образованием для них новых значений. Можно утверждать, что исполнение алгоритма представляет собой целенаправленный процесс видоизменения величин.

Рассмотрим пример одной и той же величины, представленной в разных формах. Вещественная константа 1.0, арифметическое выражение $25.0 - 15.0 - 9.0$, вещественная функция $\text{Cos}(0)$ — все они являются формами представления одной и той же величины, несущей информацию об одном и том же числовом значении, равном вещественной единице.

А теперь рассмотрим пример того, каким образом можно осуществить видоизменение формы представления некоторой величины. Командой присваивания $x:=2+3$ целочисленной переменной x присваивается значение арифметического выражения $2+3$. Следовательно, непосредственно после выполнения этой команды величина, несущая информацию о целочисленном значении 5 будет существовать уже не только в форме выражения, но и в форме переменной.

В алгоритмах и программах к величинам относятся следующие объекты: константы, переменные, обращения к функциям и выражения. В языке Паскаль в качестве важнейшей характеристики величины используется её тип. В общем случае тип величины определяет количество и характер отдельных элементарных значений в её составе, а также порядок доступа к ним. При этом различают величины простых и структурированных типов. Величина простого типа представляет собой элементарное (единственное, отдельное и неделимое) значение. Величина структурированного типа представляет собой совокупность значений. В общем случае основные признаки величины структурированного типа таковы: возможность рассматривать величину как единое целое; наличие в составе величины отдельных элементарных значений; упорядоченность отдельных значений в составе величины, то есть наличие структуры.

Основой для построения любой структуры данных является способ

представления элементарных значений в памяти вычислительной машины. Поэтому ближайшей задачей при изучении структур данных является рассмотрение двоичной системы счисления.

Литература:

1. Вирт Н. Алгоритмы + структуры данных = программы. – М.: Мир, 1985.
2. Моргун А.Н., Кривель И.А. Программирование на языке Паскаль. Основы обработки структур данных. – М.: Издательский дом “Вильямс”, 2006.
3. Моргун А.Н. Решение задач средствами языка Turbo Pascal 7.0. – К.: Юниор, 2002.

СУЧАСНІ ТЕНДЕНЦІЇ В ОФОРМЛЕННІ ІНТЕРФЕЙСІВ ПРИКЛАДНИХ ПРОГРАМНИХ ЗАСОБІВ

О.А. Смалько

м. Кам'янець-Подільський, Кам'янець-Подільський державний університет
sm-lena@mail.ru

Навчаючи студентів вищих закладів освіти програмувати прикладні програмні засоби, варто зосередити увагу не лише на аналізі найкращих зразків програмної продукції, загальновідомих правил і принципів проектування та розробки прикладних програм, а й на актуальних питаннях практичності, комфортності дизайну, інтуїтивної зрозумілості інтерфейсів, на правилах “гарного тону” в організації структури і дизайну програмних середовищ, а також на не менш важливих питаннях про необхідність дотримання вимог, пропонованих у міждержавних стандартах країн СНД, у стандартах і специфікаціях, розроблених національною системою стандартизації, узгоджених з міжнародними та європейськими стандартами.

Розробники, приділяючи багато уваги функціональному наповненню програмного продукту, неодмінно повинні враховувати ергономіку сприйняття і логіку розуміння інтерфейсу, адже користувач повинен швидко пристосовуватись до програмного середовища, адекватно інтерпретуючи умовні позначення, піктограми, навігаційні знаки, кнопки керування тощо.

Графічний інтерфейс повинен легко запам'ятовуватись, бути привабливим та добре продуманим. В оформленні бажаний дружній та інструктивний інтерфейс, який за рахунок ретельно продуманих характеристик і грамотної структури може навчати користувачів. Таке розповсюджене поняття, як “інтуїтивний” інтерфейс, краще розуміти у значенні “вгадуваний”. У “вгадуваних” інтерфейсах традиційні, стандартні компоненти мають очікувану поведінку, а нові – логічну та зрозумілу.

Світова практика розробки програмного забезпечення диктує також деякі правила практичності, що побудовані на знаннях про людино-машинну взаємодію і підкріплюються великим практичним досвідом. Зупинимось на них.

Обсяг інформації, яку необхідно запам'ятовувати користувачу, повинен бути якомога меншим. Це зменшення можливе за рахунок застосування візуальних елементів, що визначають властивості, дії і команди.

Концептуальна модель інтерфейсу повинна бути близькою користувачеві. Набір дій, здійснюваних користувачем, повинен мати яскраво виражений, чітко визначений, несуперечливий характер. Засоби керування навігацією повинні виглядати цілісно, їхнє розташування повинно бути передбачуваним у всіх частинах інтерфейсу. Також повинно бути передбачено немодальну взаємодію, прискорений і скорочений клавішний доступ до команд.

Стан програмної оболонки повинен бути цілком спостережуваним. Постійно повинно відбуватись оповіщення користувача про те, що саме виконує програма. В системі повинен передбачуватись продуманий і своєчасний зворотний зв'язок, у разі потреби – підтвердження введених даних, запобігання можливих помилок, допомога у виявленні, розпізнаванні, діагностиці і виправленні помилок, а також допомога при відновленні роботи після їх появи. Необхідно також організувати повноцінну довідкову систему і оформити потрібну документацію.

Доцільно також програмно реалізувати користувацький контроль за системою і можливість легкого виходу з неї, скасування та повтору дій. Програмна оболонка повинна організовуватись за принципом несуперечності, тобто схожі дії в ній повинні реалізовуватись схожим чином. У програмі повинні використовуватись зрозуміла користувачу мова і домовленості, що мають практичний зміст.

Програма повинна мати можливість налаштовуватись користувачем, щоб відповідати вимозі гнучкості й ефективності використання.

Безсумнівно, важлива естетична цілісність програми, причому дизайн по можливості повинен бути мінімалістським, тобто в ньому повинна скорочуватись рідко використовувана інформація.

У програмному інтерфейсі важлива гарна розмітка екрану. З неї починається залучення уваги користувача. Гарний інтерфейс відразу ж привертає погляд користувача до першого поля введення, до початкової послідовності дій або важливої (ключової) інформації. Користувач повинен бути поставлений у такі умови, в яких би він міг швидко і точно визначити, де знаходиться найважливіша інформація і найголовніші частини інтерфейсу. У розробника в розпорядженні є кілька методів, за допомогою яких він може залучати і переключати увагу користувача. Мова йде про кольорове виділення, графіку, анімацію, розмежування, підсвічування, стилі і розміри шрифту, товщину ліній і т.п. Проте, в один контекст взаємодії не слід включати надто багато таких прийомів або розміщувати в ньому занадто велику кількість важливої інформації. Варто пам'ятати також, що у візуальній системі найбільше привертають увагу елементи, що рухаються, на другому місці – виділення за допомогою кольору.

Вибір і групування візуальних елементів інтерфейсу повинен бути осмисленим і спрямованим на просту та логічну взаємодію. Тому природно починати взаємодію з користувачем з лівого верхнього кута екрану, переводячи увагу користувача на правий нижній кут. Утім, можливі й інші рішення, що передбачають більш тривале блукання користувача по контексту взаємодій.

Важливо також продумати найбільш доцільну організацію робочого потоку, пов'язаного з програмним забезпеченням. Не слід наслідувати не завжди виправдані тенденції розбивання процесу на маленькі кроки, кожний з яких представлений в окремому контексті взаємодій. Подібний інтер-

фейс може змушувати користувача надто часто переключатись між контекстами, образами, вікнами, розбиваючи при цьому цілісне завдання на багато малих підзадач, що забирають багато часу для їх вирішення і відволікають увагу від основної задачі. Подібні прийоми є сенс використовувати лише в окремих випадках, наприклад, при організації роботи майстрів.

У користувацькому інтерфейсі для найбільш ефективної підтримки взаємодії інструментальних засобів візуальні елементи необхідно розбивати на семантичні групи, що допомагають вирішувати підзадачі. Тому компоненти, що сприймаються разом (або зазвичай використовуються разом), краще розташовувати недалеко одне від одного в користувацькому інтерфейсі. Це допоможе в створенні ефективного, компактного, гнучкого, функціонального інтерфейсу. У цьому розумінні дуже популярними є “плаваючі” інструментальні панелі, на яких розміщують інструменти, що легко піддаються настроюванню.

Інформацію, інструменти та функції потрібно так розподіляти між контекстами взаємодії, щоб вони не виглядали розкиданими по декількох екранах або діалогових вікнах. У протилежному випадку робота користувача буде неефективною, при цьому також може продукуватись велика кількість помилок.

Досить ефективно дозволяє організувати інформацію і заощаджувати екранний простір синхронна індикація – набір пов’язаних між собою контекстів взаємодій, влаштованих таким чином, що дії, виконані в одному контексті, миттєво впливають на інші контексти. Синхронізовані контексти взаємодій дозволяють користувачам переглядати одну й ту саму інформацію в різних формах, так, як їм зручно. Розподіл інформації із синхронізованих контекстів взаємодій зменшує навантаження на короткочасну пам’ять користувача.

У відповідності зі структурним принципом практичність підвищується, коли схожі або взаємозалежні елементи подаються в інтерфейсі схожим чином, а несхожі та незалежні відповідним чином відокремлюються один від одного. Гарна організація візуальних елементів дозволяє користувачеві розуміти зміст всіх контекстів взаємодій.

Організація візуальних елементів у вікнах здійснюється за допомогою набору графічних прийомів, що поєднують чи роз’єднують об’єкти за їхніми видимими особливостями. Важливо також продумано розташовувати елементи, забезпечуючи сусідство лише взаємозалежних об’єктів або демонструючи взаємозв’язок віддалено розміщених об’єктів за допомогою, наприклад, кольору, звичайних ліній чи візуально утворюваних блоків.

Для підвищення практичності інтерфейсу та забезпечення зорового комфорту також корисно залишати пропуски між видимими елементами, значками, написами і межами (краями) командних кнопок, панелей, діалогових вікон, не групуючи їх біля країв вікон чи рамок.

Простір між об’єктами може мати білий, сірий і навіть чорний колір,

залежно від оформлення інтерфейсу, тому потрібно враховувати не лише його розмір, але і колір. Адже, як правило, графічні користувацькі інтерфейси виглядають більш привабливо і навіть легше сприймаються, коли білий колір хоча б приблизно збалансований і рівномірно розподілений по контексту взаємодії. Досвідчені розробники інтерфейсів вважають, що ідеальною є рівна кількість білого кольору у верхній і нижній половині контексту взаємодії, а також ліворуч і праворуч. Тим же розумінням вони керуються, вибираючи місце розташування керуючих елементів та інших видимих об'єктів: ліворуч і праворуч, зверху і знизу їхня кількість повинна бути приблизно однаковою. Метою при цьому є не досягнення бездоганної симетрії, а створення відчуття збалансованості.

Коли необхідно в контексті взаємодії розмістити багато об'єктів або елементів керування, як правило, організовують діалогові вікна з закладками. Однак при цьому варто ефективно використовувати екранний простір, щоб одні закладки не виявлялись практично порожніми, а інші – перевантаженими керуючими елементами. Ефективний розподіл елементів по сторінках діалогу – задача непроста і потребує перегляду принципів розподілу різних категорій інформації. При цьому важлива візуальна організація, що спирається на семантику і параметри задачі, естетичне задоволення при цьому є все-таки менш значимим фактором [1].

Але інтерфейс обов'язково повинен бути охайним. Усі видимі елементи вікна повинні бути створені на професійному рівні, вирівняні з усіх боків, повинні гарно виглядати. Основні засоби навігації і стандартні командні кнопки повинні бути однакового, стандартного розміру, хоч іноді окремі елементи екранних інтерфейсів можуть відрізнитись як за розміром, так і за формою. Проте всередині одного контексту взаємодії не варто використовувати компоненти надто різних типорозмірів, адже необхідно прагнути, щоб інтерфейс виглядав акуратно і професійно.

Графічний користувацький інтерфейс немислимий без значків. Значки – це графічні символи, зображувані на інструментах та інших видимих елементах. Існують значки, які безпосередньо зображують потрібні об'єкти, а також такі, що являють певну аналогію з потрібним інструментом, об'єктом, командою чи дією. Інколи при зображенні значків, за допомогою яких мають виконуватись певні операції, використовують метафори, омоніми і такий прийом, як синекдоха (коли частина об'єкта представляє цілий об'єкт або навпаки). Однак усе це не повинно бути надто заумним, далеким від реальності чи походити на візуальні каламбури. “Натягнуті”, неприродні імітації можуть зробити інтерфейс незрозумілим, а спрощені моделі можуть ускладнити користування програмою.

Проте важливо пам'ятати, що якщо на придумування одного значка йде більше декількох хвилин, то такий значок вже не буде вважатись інтуїтивно зрозумілим. Якщо на розгадування призначення якого-небудь елемента в користувача йде більш однієї-двох секунд, напевно чи такий елемент можна

назвати “очевидним” або “легко вгадуваним”. Незрозумілі значки роблять систему більш важкою в запам’ятовуванні, що призводить до уповільнення процесу опанування роботи з нею. При проектуванні значків та інструментів слід дотримуватись простого, послідовного стилю з помірним використанням кольорів, при якому значки легко розрізняються і розпізнаються [1].

Скрізь, де лише можливо, при створенні значків розробник повинен дотримуватись визначених домовленостей, маючи на увазі найочевидніші асоціації і прийняті стандарти. Універсальні знаки завжди повинні мати подібність з тим, що вони представляють. Варто піклуватись про те, щоб створені значки не суперечили значкам, що добре знайомі користувачам.

Звичайно, простіше всього за допомогою значків представляти об’єкти або конкретні іменники: вони просто зображуються, іноді в спрощеній або абстрактній формі. Дії зображувати складніше, однак їх іноді вдається замінити іменниками. Найскладніше передати за допомогою значків абстрактні дієслова. Тут розробнику доводиться мати справу зі знаковою, ієрогліфічною системою, семіотикою. Однак не слід думати, що з кожним елементом, дією, командою повинен бути зв’язаний унікальний інструмент або значок.

Серед знаків, за допомогою яких, як правило, прискорюється адаптація користувача до інтерфейсу, можна назвати гліфи – маленькі значки на командних кнопках. В принципі, гліфи на кнопках загромождають користувацький інтерфейс, однак їхнє вибіркове використання цілком припустиме і може допомогти користувачеві швидше зрозуміти, що натискати, навіть якщо кнопки розташовані десь у незвичному місці. Але не всім користувачам гліфи на командних кнопках подобаються, тому їхня наявність або відсутність, як і наявність або відсутність графічних значків на інструментальних панелях, повинні регулюватись користувачем.

Зрозуміло, що при наявності великої кількості інструментальних засобів, пропонуваніх програмою, збільшується і кількість інструментів, які потрібно використовувати. Ці інструменти, як правило, розміщують на інструментальних панелях. Доцільна організація інструментів завжди повинна базуватись на їхньому значенні і застосуванні. Набори інструментів часто розбивають на категорії, і якщо не доцільно розносити їх на різні панелі, то необхідно функціонально їх розділяти, наприклад за допомогою вертикальних ліній чи обмежувачих рамок.

Інструментальні панелі можуть мати як горизонтальне, так і вертикальне розташування. Однак, слід зазначити, що в деяких випадках вертикальне розташування дуже ускладнює виконання операцій, особливо тих, що проводяться попарно.

При розробці інтерфейсів програмних засобів також обов’язково мають враховуватись такі немаловажні психологічні аспекти комп’ютерного дизайну, як питання ілюзій зображень, в тому числі анімованих, контрасту кольорів, розмірів, форм, “підпорядкування” одних об’єктів іншим тощо.

Важливі також не лише форма, колір та інтуїтивна зрозумілість

об'єктів, знаків, піктограм, але й загальна кількість рівноцінних елементів на екрані. Наприклад, якщо число однакових кнопок навігаційної панелі не перевищує 6–7, користувач здатний швидко “зафіксувати” їх у пам'яті і звести користування ними до автоматизму.

Рядок меню у вікні програми (як правило, він передбачений у тому чи іншому вигляді у будь-якій прикладній програмі) повинен містити основні категорії або набори задач верхнього рівня, щоб користувач з максимальною ймовірністю міг з першої спроби вгадати, де він зможе знайти потрібні функції і настроювання. При конструюванні рядка меню слід пам'ятати, що як недостатність, так і надлишок меню говорить про проблеми зі структурою. Недостатність, наприклад, може означати, що багато функцій глибоко “заховані” у вкладені структури меню, а це, в свою чергу, може ускладнювати задачу пошуку того, що потрібно користувачу для вирішення його задач. Занадто великого числа категорій команд, довгих меню і надто глибокої вкладеності структури меню також слід уникати. Як правило, добре продумані рядки меню містять 3–12 назв, але для більшості програм достатньо 6–9 назв. У вкладеності меню два-три рівні повинні вважатися максимумом.

Що стосується кількості елементів у кожному з меню, то воно також повинно бути розумним, проте слід мати на увазі, що більш вдалим варіантом є більша кількість пунктів і менший ступінь вкладеності, але не навпаки. Саме така політика сприяє швидкому розумінню і швидкій навігації. Вертикальні розміри вікна накладають жорсткі обмеження, у які повинен вписуватись розробник, уникаючи при цьому рішень, за яких використовуються багато стовпців. Якщо меню організовано добре, то на його верхньому рівні рідко зустрінеться більше 15–20 елементів. Вкладені меню повинні бути меншими, ніж меню верхніх рівнів. Тут не повинно бути більше 7–9 пунктів [1].

Зовнішньо повинні розрізнятись різні категорії об'єктів, що знаходяться в меню. Так, посилання на дії, що виконуються безпосередньо при натисканні на них, повинні відрізнятись від посилань, що відкривають діалогові вікна і наступні вкладені рівні меню.

При проектуванні графічних користувацьких інтерфейсів слід передбачати можливість виконання всіх операцій за допомогою однієї лише клавіатури. Один зі способів надання клавіатурного доступу – це призначення так званих “прискорюючих” (або “гарячих”) клавіш, що спрацьовують при одночасному натисканні, як правило, клавіші Alt і підкресленої літери чи цифри, про яку можна довідатись у назвах меню і пунктах цих меню. “Гарячі” клавіші слід добирати дуже ретельно, по можливості, варто використовувати комбінації, що вже є традиційними або зарезервовані в багатьох розповсюджених програмах. Бажано, щоб “прискорюючі” комбінації клавіш не були просто випадковим набором літер, а легко запам'ятовувались і забезпечували спрощену роботу з клавіатурою.

Розробник повинен продумати як повну, так і скорочену версію меню, панелей (чи наборів) інструментів, команд. Скорочені версії зазначених об'єктів інтерфейсу, зазвичай, з'являються за замовчуванням при встановленні програми і повинні містити в собі найважливіші, ключові елементи. Коли користувач не використовує окремі команди, кнопки чи панелі інструментів протягом тривалого часу, доцільно запрограмувати можливість їхнього приховування (із можливістю подальшого розгортання до повного вигляду в разі потреби).

Корисними вважаються також контекстні меню, які з'являються при натисканні правої клавіші миші, сукупності клавіш Shift+F10 або спеціальної клавіші Application (Додаток) на клавіатурах з підтримкою Windows. Вони надають доступ до багатьох функцій програми, не займаючи при цьому екранного простору доти, поки їх не викликають. На практиці контекстні меню застосовуються не всіма користувачами, оскільки такі меню вимагають швидкої переорієнтації і переключення уваги, а це не для всіх людей є легкою справою. Але створювати їх все ж доцільно.

Правила організації контекстних меню нагадують правила організації звичайних випадаючих меню, з тією лише різницею, що у контекстному меню повинні розміщуватись операції, які відносяться до виділеного об'єкта. У контекстному меню не повинно бути більше 7–8 пунктів, а порядок їхнього розташування повинен відповідати передбачуваній імовірності звертання до відповідних функцій. При цьому досить спірною вважається доцільність дублювання команди, яка може викликатись за допомогою подвійного натискання клавіші миші, у першому пункті контекстного меню [1].

Корисною вважається тенденція використання технології розгортання меню в обидва боки, а також програмна реалізація “липких” меж, що дозволяє не випускати покажчик курсору з фокусу елемента меню чи кнопки навіть тоді, коли він відходить на деяку відстань.

Що стосується компонентів користувацького інтерфейсу, за допомогою яких користувачі здійснюють вибір певних параметрів, то їх розробляти ще складніше. Базовий набір елементів керування досить простий і, як відомо, представлений декількома основними компонентами: перемикачами (селективними кнопками), прапорцями, лічильниками, фіксованими і випадаючими списками.

При використанні в інтерфейсі елементів керування необхідно дотримуватись певних рекомендацій. Так, перемикачі найкраще використовувати, коли потрібно вибирати одне значення з відносно невеликої множини. Їхня перевага полягає в тому, що видимими є всі елементи, тому користувачеві легко охопити поглядом відразу всю множину значень. Однак якщо список великий, то перемикачі будуть займати надто багато місця на екрані. При мінливих розмірах множини значень перемикачі використовуються рідко, тому що в цьому випадку слід або якимось намагатись змінювати екранний

простір, що відводиться під них, або якимось видозмінювати недоступні значення.

Традиційні інструменти для вибору будь-якої кількості значень з множини – прапорці – найбільш ефективно працюють, коли передбачається робота з помірною кількістю значень. Однак інколи їх використовують і з досить великими наборами даних. Коли значень справді багато, їх краще поділити на логічні групи, елементи в яких об'єднані або на основі семантики, або на основі функціональності. Як і перемикачі, прапорці рідко використовуються, коли множина значень велика або її розмір може варіюватися.

Для вибору одного або декількох елементів з групи також можуть використовуватися кнопки, які виступають в ролі вимикачів. При натисканні на таку кнопку вона стає натиснутою, при повторному натисканні вона вимикається. Щоб пояснити користувачу, що кнопки “поводяться” як група вимикачів, розробники, за звичай, обводять їх рамками чи відділяють від інших стилем, кольором, формою, лінією тощо, підкреслюючи їх взаємозв'язок і те, що вони працюють в групі.

Лічильники найчастіше використовуються з числовими значеннями порівняно невеликого діапазону, хоча можуть застосовуватися і з іншими типами даних. Якщо значення не є числовими, то їх варто відсортувати за якимось природним, очевидним принципом, інакше користувач ніколи не знайде те, що йому потрібно. Крім того, у цьому випадку особливо важливо, щоб множина значень була невеликою. Лічильники зручні своєю компактністю і простотою у використанні. Головним їхнім недоліком є те, що неможливо переглянути всі значення відразу, хоча вони повинні вгадуватися за тим, як відсортовані.

Фіксовані списки мають обмеження за розміром видимої ділянки, але не за числом елементів, що входять до них. Це означає, що у фіксованих списках може міститись більше значень, ніж у перемикачах, і при цьому вони будуть займати менше місця на екрані, але досягти цього можна лише шляхом встановлення смуги прокручування. Через це знижується видимість, а необхідність у прокручуванні списку знижує продуктивність.

Випадаючі списки націлені на економію екранного простору. Зрозуміло, за це доводиться розплачуватись швидкістю і зручністю. Вони займають навіть менше місця, ніж фіксовані списки, оскільки насправді користувач бачить не список, а лише поточне його значення доти, поки не відкриє його за допомогою спеціальної кнопки розкриття. Випадаючі списки можна удосконалювати, забезпечивши, наприклад, автоматичну появу списку при влученні у фокус кнопки розкриття покажчика миші.

Існують також розширені фіксовані списки. Вони, на відміну від звичайних, дозволяють вибрати більше одного значення зі списку. Список може повністю не поміститись у видиму ділянку, може містити в собі змінювану кількість елементів. Для позначення виділених елементів встановлюю-

ються спеціальні позначки, що можуть виглядати по-різному. На практиці такі списки зустрічаються рідко, оскільки для виділення кожного значення списку його необхідно заново відкривати, а це не дуже зручно для користувачів. Також не дуже зручним є те, що для перегляду всіх виділених елементів списку його необхідно розкривати.

Іноді програмісти використовують комбіновані елементи керування, що сполучають у собі якись два елементи, наприклад, текстове поле введення з випаданим списком із фіксованою кількістю значень, так, що користувач може або вибрати один із заданих пунктів, або ввести значення в текстове поле. Таке рішення може служити для двох різних цілей, одна з яких полягає в наданні користувачеві можливості введення нових значень, а інша – у швидкому виборі з уже наявних. В руках умілого користувача, що звик до того ж працювати з клавіатурою, комбіновані елементи керування виявляються особливо ефективним інструментом, оскільки досить лише почати введення шуканого значення, як можливі варіанти, що починаються з уведених літер, самі з'являються в полі введення. Зрозуміло, цю можливість попередньо слід запрограмувати.

Керуючі елементи вибору іноді можуть міститись в меню. Такі пункти меню повинні “включатись” і “виключатись”. Для цього застосовуються традиційні позначення: маленька крапка (у випадку перемикача) і галочка (у випадку прапорця), наявність чи відсутність яких говорять про встановлений стан об'єкта меню [1].

Звичайно, всі текстові написи у вікні повинні подаватись в єдиному, стандартизованому вигляді. В цілому, інтерфейс не повинен бути перевантаженим словами. Мовні засоби слід застосовувати точно і в потрібних місцях. При виборі манери написання тексту слід пам'ятати, що читання інформації, написаної великими літерами може уповільнити процес сприйняття її читачем, курсивний текст може здаватись світлішим за оточуючий і марніти на його фоні, крім того, за найбільш популярних роздільних здатностей сучасних моніторів складні курсивні шрифти можуть здаватись дещо рваними у порівнянні з прямими того ж розміру і тієї ж гарнітури. Важливо також при веденні діалогу з користувачами уникати дрібних шрифтів. Крім цього, потрібно пам'ятати, що будь-який шрифт – як будь-яка форма і будь-який колір – має свою тональність, настрій [2].

По можливості, в програмі слід реалізувати функцію автоматичного масштабування вікон повідомлень і діалогових вікон, які адаптуватимуться під мінливі розміри шрифтів.

Не ставлячи за мету видовищність інтерфейсу, можна, однак, використовувати певну кількість кольорів для оформлення елементів інтерфейсу та достовірного відображення об'єктів, що містяться на кнопках, панелях тощо. За потреби можна також використовувати високоякісні графічні зображення, виконану з високим ступенем вірогідності анімацію, в тому числі і тривимірну, коментарі, підказки, звуки подій (звичайно, із реалізованою

можливістю в разі потреби все це приховувати чи відключати).

При виборі колірної рішення програмного інтерфейсу слід пам'ятати, що в 93% людей певні кольори і їхні сполучення викликають ті самі емоції, а також те, що не всі люди здатні бачити всю повноту фарб світу, адже приблизно одна дванадцята частина всіх чоловіків і трохи менша частка жінок страждає тією чи іншою формою дальтонізму, колірною сліпотою.

Психологічний аспект людського кольоросприйняття і колірних асоціацій, сполучуваності кольорів уже досить непогано розроблений. Також існує багато корисних рекомендації щодо добору колірної гами інтерфейсів. Не слід, однак, зловживати при створенні інтерфейсів різнобарвними палітрами з великими ділянками яскравих, інтенсивних кольорів. Деякі сполучення кольорів, такі як зелений на червоному або червоний на зеленому, як відомо, особливо негативно впливають на людину.

Незважаючи на те, що інформацію і різні деталі можна виділяти за допомогою кольору, колірне кодування саме по собі не варто використовувати для виділення якихось справді суттєвих для користувача компонентів. Слід також уникати сполучення тексту, графіки й фону з подібною насиченістю кольорів чи з однаковою яскравістю, уникати фонів, перевантажених графічними елементами. Варто дотримуватись відповідності колірної палітри, оптимальності сполучення кольорів і їх яскравості, а також підтримувати високий контраст всього екранного зображення [2].

Якщо в інтерфейсі передбачено використання анімаційних елементів, динамічних зображень, то бажано, щоб вони не були ненав'язливими, а мали певну інформативну функцію. Користувачу доцільно надавати можливість забирати з екрану рухливі компоненти інтерфейсів в разі потреби, адже вони, як відомо, привертають до себе значну увагу і можуть спричинити гальмування процесу розв'язування поставлених задач.

При використанні в створюваних програмах звуків, слід надати повний контроль над звуками користувачам, причому такий, що не буде зведений до таких примітивних інструментів, як регулятор гучності в операційній системі чи кнопка на колонках.

Звуки необхідно співвідносити із оточенням, у якому вони будуть звучати. Смысл усіх звуків має розуміти навіть недосвідчений користувач. Важливо для різних цілей застосовувати різні звуки. Наприклад, легкий щиглик або інший ненав'язливий, але "підбадьорюючий" звук може служити підтвердженням введення. Приємний, але вже більш виразний звук (наприклад, дзенькіт дзвіночка) може говорити про успішне завершення операції або введення даних. Невелика помилка може супроводжуватись трохи менш приємним, проте привабливим увагу звуком, наприклад коротким мінорним акордом або просто гудком.

Сучасною тенденцією в оформленні прикладних програмних продуктів може стати ідея "мульти-дизайності", про яку вже почали говорити вітчизняні фахівці [2]. Це реалізована в програмі можливість зміни дизайну безпо-

середньо під час роботи з нею. Мова йде не лише про зміну колірної гами інтерфейсу, а про можливість вибору зовсім іншого, нового дизайну, що повністю змінює вигляд робочого вікна. Це може частково вирішити проблему вибору того дизайну, який найбільше подобається користувачу, найкраще підходить до обстановки, відповідає настрою і т.ін.

Література:

1. Константайн Л., Локвуд Л. Разработка программного обеспечения / Перевод с англ. В.Шрага. — СПб.: Питер, 2004. — С. 60-236.
2. Кравцов Д.Г. Аналіз і реалізація модуля для обробки множинних дизайнів для сайтів та електронних підручників // Науковий часопис НПУ імені М.П. Драгоманова. Серія №2. Комп'ютерно-орієнтовані системи навчання: Зб. наук. праць / Редкол. — К.: НПУ ім. М.П. Драгоманова. — №2 (9). — 2005. — С. 287-294.

МІЖПРЕДМЕТНІ ЗВ'ЯЗКИ ТА ФОРМУВАННЯ ЕЛЕМЕНТАРНИХ УМІНЬ РОБОТИ У СЕРЕДОВИЩІ AUTOCAD

Ю.О. Дорошенко, В.О. Очеретний
м. Київ, Інститут педагогіки АПН України
labinf@edu-ua.net

Освоїти комп'ютерну графіку (КГ) можливо лише шляхом виконання певних практичних завдань на комп'ютері із використанням відповідних програмних засобів растрової і векторної графіки. Звичайно, за умови попереднього вивчення необхідних теоретичних основ виконання таких завдань на методологічному, змістовому та операційному рівнях.

Для початкової реалізації такого навчання пропонується лабораторна і практична роботи зі спеціалізованого комп'ютерного практикуму. Зазначені роботи реалізують методику поступового опанування роботи у середовищі графічних редакторів як багатофункціональних інструментальних засобів з опорою на розв'язання системи завдань, які змістово забезпечують фундаментальність такого навчання та реалізують міжпредметні та внутріпредметні зв'язки. Базовим інструментальним програмним засобом обрано САПР AutoCAD як одну із найбільш поширених та багатофункціональних програм. Опанування цієї програми дозволить без великих зусиль освоїти роботу з іншими графічними редакторами.

Лабораторна робота №1

Тема: Графічне моделювання тривимірного простору. Команди побудови графічних примітивів, редагування фрагментів зображення, режими об'єктної прив'язки та вимірювання відстаней у AutoCAD. Комп'ютерний тренажер “Епюр точки”.

Мета:

Навчальна:

- актуалізувати та систематизувати знання з основ нарисної геометрії щодо координатно-площинного моделювання простору (епюр Монжа) та задання у цій моделі геометричного об'єкта “точка”, запису визначника точки та з'ясування її положення (октант) у тривимірному просторі;
- надати початкові відомості щодо роботи у середовищі AutoCAD;
- сформулювати уявлення про пошарову структуру комп'ютерної моделі векторного графічного зображення;
- сформулювати елементарні навички побудови графічних примітивів та редагування простих лінійних зображень у AutoCAD;
- засвоїти команди AutoCAD: LINE, DONUT, DIST, COPY, MOVE, DTEXT;
- опрацювати застосування режимів об'єктної прив'язки AutoCAD: ENDpoint, CENTER, INTERsec;
- опанувати побудову прямих ліній у фіксованому (орто-) режимі;
- навчитися переглядати і редагувати властивості графічних об'єктів.

Виховна:

- виховати акуратність, чіткість, точність виконання дій за комп'ютером шляхом реалізації необхідних графічних побудов;
- прививати технологічну культуру роботи у середовищі векторного графічного редактора.

Розвивальна:

- активізувати та розвинути теоретико-практичне мислення під час розв'язання змістово і сутнісно відомої задачі у нових умовах;
- розвинути просторову уяву та алгоритмічне мислення.

Практично-корисна:

- опанувати прості прийоми роботи та навчитися розв'язувати нескладні графічні завдання у середовищі AutoCAD.

Завдання. На епюрі Монжа (за варіантом):

- побудувати відсутню проекцію точки за двома заданими;
- записати визначники (три) проекцій точки;
- записати визначник точки;
- відносно заданої декартової системи координат вказати октант (1–8) тривимірного простору, в якому знаходиться точка.

Інструктивне пояснення. Робота виконується у середовищі AutoCAD за варіантами. Кожен варіант представлений пошаровим кресленням, записаним у файлі. Кожне креслення містить по три точки, епюри яких рознесені за окремими шарами.

Усі побудови виконуються інструментальними засобами AutoCAD.

Стислі відомості про використовувані команди та методичні рекомендації щодо виконання заданої роботи

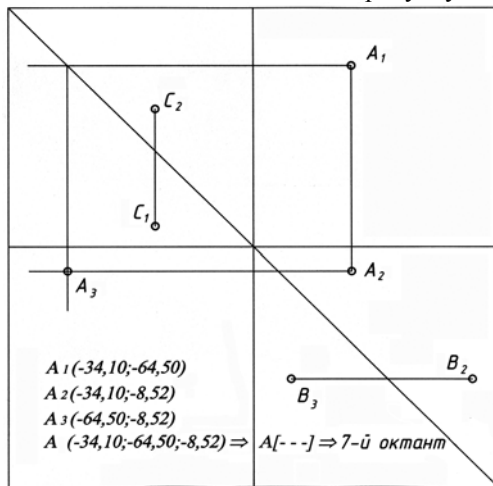
Хто поставив собі за мету опанувати КГ та КГІТ, мають, перш за все, навчитися моделювати геометричний простір із розташованими у ньому об'єктами; записувати визначники геометричних фігур та тіл; формулювати, формалізувати і розв'язувати позиційні і метричні задачі. При цьому, реалізується перехід від навчання класичної нарисної геометрії до комп'ютерної нарисної геометрії, тобто, відбувається інтеграція сучасних комп'ютерних технологій до усталеної дисципліни.

У зв'язку з означеним, на початку навчального процесу з КГ та КГІТ пропонується застосовувати комп'ютерний тренажер, призначений для моделювання точки на епюрі Монжа як першооснови опрацювання моделей тривимірного геометричного простору. Цей тренажер створений і функціонує у середовищі САПР AutoCAD.

Під час виконання цього завдання, наочувані, окрім закріплення фундаментальних знань з основ нарисної геометрії, ознайомлюються з правилами роботи у середовищі САПР AutoCAD та формують початкові уміння і навички виконання простих графічних побудов із вказуванням певних режимів їх реалізації та власне роботи програми. Зокрема, вчать проводити строго вертикальні чи горизонтальні лінії, користуватися режимами

об'єктної прив'язки, маніпулювати шарами креслення та налаштовувати їхні параметри (властивості), визначати відстані між двома точками, друкувати та редагувати текст, переглядати і змінювати властивості графічних об'єктів тощо.

Приклад виконаного завдання наведено на рисунку.



Перевірка правильності виконаного завдання реалізується за допомогою спеціальної програми. За цією програмою відтворюються проекції точки, заданої за знайденим визначником. У разі правильного результату програмно відтворені проекції збігаються із заданими у завданні.

Практична робота №2

Тема: Побудова креслення “Прокладка” у середовищі AutoCAD за протоколом.

Мета:

Навчальна:

- актуалізувати та систематизувати знання з креслення;
- надати початкові відомості щодо побудови креслень у середовищі AutoCAD;
- сформулювати уявлення про пошарову структуру комп'ютерного креслення;
- сформулювати уявлення про загальну декартову, локальну (місцеву) декартову та локальну полярну системи координат креслення і навчити явно задавати положення точок у цих системах координат;
- засвоїти команди AutoCAD: LINE, POINT, CIRCLE, TRIM, PEDIT, DIM(hor), DIM(ver), DIM(diam);
- опрацювати застосування режимів об'єктної прив'язки AutoCAD: ENDpoint, CENTER, INTERsec, TANGent;
- закріпити вміння побудови прямих ліній у фіксованому (орто-) режимі;
- сформулювати навички побудови графічних примітивів та редагування простих лінійних зображень у AutoCAD шляхом відрізання певної частини ко-

нтуру;

- навчити об'єднувати лінійні примітиви у цілісну полілінію, редагувати її та змінювати властивості;
- сформувані уявлення про нанесення розмірів на комп'ютерному кресленні і навчити наносити лінійні (вертикальні і горизонтальні) розміри, розміри діаметрів та редагувати їх відповідно до вимог ЄСКД;
- навчити переглядати і редагувати властивості графічних об'єктів.

Виховна:

- виховувати акуратність, чіткість, точність виконання вказаних (у протоколі) дій за комп'ютером шляхом введення певних команд і опцій та реалізації необхідних графічних побудов;
- прививати технологічну культуру роботи у середовищі векторного графічного редактора;
- виховувати технологічну дисципліну виконання графічних побудов.

Розвивальна:

- активізувати та розвинути теоретико-практичне мислення у процесі виконання роботи за заданим сценарієм (протоколом);
- розвинути алгоритмічне мислення та інформатично-технологічну культуру.

Практично-корисна:

- опанувати прості прийоми роботи та навчити будувати нескладні креслення плоских деталей у середовищі AutoCAD.

Послідовність виконання роботи.

1. Завантажити файл шаблону креслення vprava.dwg.
2. Зберегти креслення під новим ім'ям у власній папці.
3. Побудувати креслення за наведеним нижче протоколом.
4. Зберегти побудоване креслення у власній папці.
5. Представити креслення викладачеві для перевірки.
6. Відтворити креслення на плоттері.

Інструктивне пояснення. Робота виконується у середовищі AutoCAD за комп'ютерним протоколом. Робота студента полягає у точному введенні необхідних команд, опцій, режимів роботи і даних за протоколом та відслідковуванні реакції програми – виконанням нею заданих графічних побудов та генерацією певних текстових повідомлень.

Стислі відомості про використовувані команди та методичні рекомендації щодо виконання заданої роботи

Одним із можливих методичних прийомів, спрямованих на швидке опанування навчуваними побудови креслень у AutoCAD без тривалого теоретичного вивчення необхідних команд, є виконання певних дій за заданим сценарієм – протоколом. Зазначений протокол створюється програмою AutoCAD синхронно з виконанням графічних побудов за керуючою інформацією, яку вводить оператор. Тобто, учням надається заздалегідь приготовлений протокол, який забезпечує досягнення бажаного результату.

Учневі слід розібратися у наданому протоколі, з'ясувати, які дії він має виконати, якою має бути реакція програми та як працюють використані для побудов команди, опції, режими і які дані слід вводити.

У протоколі та текстова інформація, яку має вводити учень з клавіатури, виділена півжирним, околиці використовуваних для побудов точок зображення вказано їх номерами у прямокутній рамці. Реакція програми у вигляді текстових повідомлень показана звичайним шрифтом.

Команда LINE призначена для побудови прямої лінії за двома крайніми точками. Команда POINT призначена для відтворення точки за двома її координатами. За командою CIRCLE будується коло із заданими центром (точка) та радіусом (діаметром). Команда TRIM призначена для відрізання (знищення) частини лінійного об'єкта (контуру). За командою PEDIT виконується редагування складеної лінії. Нанесення розмірів відбувається за командою-режимом DIM: hor – горизонтальні, ver – вертикальні, diam – нанесення діаметру кола.

У процесі виконання побудов використовуються режими об'єктивної прив'язки OSNAP:

- ENDPoinT – кінцева точка;
- INTersection – точка перетину двох ліній;
- CENter – центр кола;
- TANgent – дотик до лінії.

Усі необхідні для виконання розглядуваного завдання параметри робочого середовища та пошарова структура креслення записані у файлі креслення (шаблоні) vrgava.dwg, який слід завантажити на початку роботи та (з метою недопущення внесення будь-яких змін до шаблону) зберегти у власній папці під оригінальним ім'ям.

Креслення, яке слід побудувати, та всі точки, посилання на які є у протоколі, показано на рисунку.

Протокол побудови креслення "Прокладка"

***** Активация шару "Osi" *****

Command: **line**

Specify first point: **50,175**

Specify next point or [Undo]: **@305,0**

Command: **point**

Current point modes: PDMODE=0 PDSIZE=0.0000

Specify a point: **end** **1**

of

Command: **point**

Current point modes: PDMODE=0 PDSIZE=0.0000

Specify a point: **@80,0**

Command: **line**

Specify first point: **@0,80**
Specify next point or [Undo]: **@0,-160**
Specify next point or [Undo]:

Command: **point**
Current point modes: PDMODE=0 PDSIZE=0.0000
Specify a point: **end**
of

Command: **point**
Current point modes: PDMODE=0 PDSIZE=0.0000
Specify a point: **@-45,0**

Command: **line**
Specify first point: **@0,45**
Specify next point or [Undo]: **@0,-90**
Specify next point or [Undo]:

***** Активация шару "Kontur" *****

Command: **circle**
Specify center point for circle or [3P/2P/Ttr (tan tan radius)]: **int**
of
Specify radius of circle or [Diameter]: **75**

Command: **circle**
Specify center point for circle or [3P/2P/Ttr (tan tan radius)]: **int**
of
Specify radius of circle or [Diameter] <75.0000>: **d**
Specify diameter of circle <150.0000>: **80**

Command: **line**
Specify first point: **tan**
to
Specify next point or [Undo]: **tan**
to
Specify next point or [Undo]:

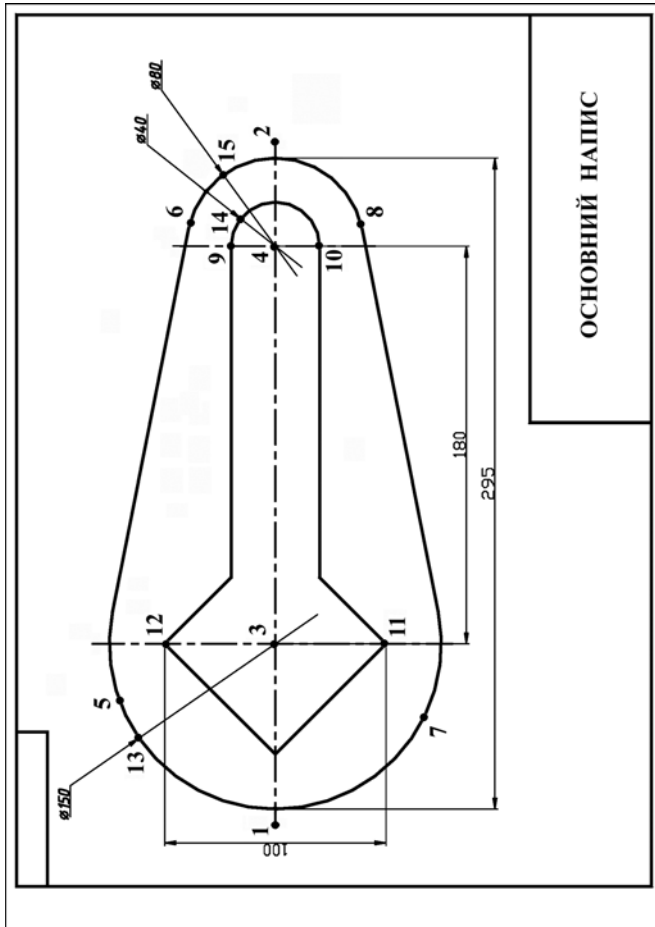
Command: **line**
Specify first point: **tan**
to
Specify next point or [Undo]: **tan**
to

Specify next point or [Undo]:

Command: **trim**

Current settings: Projection=UCS, Edge=None

Select cutting edges ...



Select objects: 1 found

Select objects: 1 found, 2 total

Select objects:

Select object to trim or shift-select to extend or [Project/Edge/Undo]:

Select object to trim or shift-select to extend or [Project/Edge/Undo]:

Select object to trim or shift-select to extend or [Project/Edge/Undo]:

Command: **circle**

Specify center point for circle or [3P/2P/Ttr (tan tan radius)]: **cen**

Specify radius of circle or [Diameter] <40.0000>: **d**

Specify diameter of circle <80.0000>: **40**

Command: **point**

Current point modes: PDMODE=0 PDSIZE=0.0000

Specify a point: **cen**

of

Command: **line**

Specify first point: **@0,50**

Specify next point or [Undo]: **@-50,-50**

Specify next point or [Undo]: **@50,-50**

Specify next point or [Close/Undo]: **@50,50**

Specify next point or [Close/Undo]: **@-50,50**

Specify next point or [Close/Undo]:

Command: **line**

Specify first point: **int**

of

Specify next point or [Undo]: <Ortho on>

Specify next point or [Undo]:

Command: **line**

Specify first point: **int**

of

Specify next point or [Undo]:

Specify next point or [Undo]:

Command: **trim**

Current settings: Projection=UCS, Edge=None

Select cutting edges ...

Select objects: 1 found

Select objects: 1 found, 2 total

Select objects:

Select object to trim or shift-select to extend or [Project/Edge/Undo]:

Select object to trim or shift-select to extend or [Project/Edge/Undo]:

Select object to trim or shift-select to extend or [Project/Edge/Undo]:

Select object to trim or shift-select to extend or [Project/Edge/Undo]: *Cancel*

Command: **trim**

Current settings: Projection=UCS, Edge=None

Select cutting edges ...

Select objects: 1 found

Select objects: 1 found, 2 total

Select objects:

Select object to trim or shift-select to extend or [Project/Edge/Undo]:

Select object to trim or shift-select to extend or [Project/Edge/Undo]:

Select object to trim or shift-select to extend or [Project/Edge/Undo]: *Cancel*

Command: **pedit**

Select polyline or [Multiple]:

Object selected is not a polyline

Do you want to turn it into one? <Y> **y**

Enter an option [Close/Join/Width/Edit vertex/Fit/Spline/Decurve/Ltype gen/Undo]: **j**

Select objects: 1 found

Select objects: 1 found, 2 total

Select objects: 1 found, 3 total

Select objects:

3 segments added to polyline

Enter an option [Open/Join/Width/Edit vertex/Fit/Spline/Decurve/Ltype gen/Undo]: **w**

Specify new width for all segments: **0.8**

Enter an option [Open/Join/Width/Edit vertex/Fit/Spline/Decurve/Ltype gen/Undo]:

Command: **pedit**

Select polyline or [Multiple]:

Object selected is not a polyline

Do you want to turn it into one? <Y> **y**

Enter an option [Close/Join/Width/Edit vertex/Fit/Spline/Decurve/Ltype gen/Undo]: **j**

Select objects: 1 found

Select objects: 1 found, 2 total

Select objects: 1 found, 3 total

Select objects: 1 found, 4 total

Select objects: 1 found, 5 total

Select objects: 1 found, 6 total

Select objects:

6 segments added to polyline

Enter an option [Open/Join/Width/Edit vertex/Fit/Spline/Decurve/Ltype gen/Undo]: **w**

Specify new width for all segments: **0.8**

Enter an option [Open/Join/Width/Edit vertex/Fit/Spline/Decurve/Ltype gen/Undo]:

***** Активация шару "Dim" *****

Command: **dim**

Dim: **hor**

Specify first extension line origin or <select object>: **cen** 3
of

Specify second extension line origin: **cen** 6
of

Specify dimension line location or [Mtext/Text/Angle]:
Enter dimension text <180>:

Dim: **hor**

Specify first extension line origin or <select object>: **int** 1
of

Specify second extension line origin: **int** 2
of

Specify dimension line location or [Mtext/Text/Angle]:
Enter dimension text <295>:

Dim: **ver**

Specify first extension line origin or <select object>: **int** 11
of

Specify second extension line origin: **int** 12
of

Specify dimension line location or [Mtext/Text/Angle]:
Enter dimension text <100>:

Dim: **diam**

Select arc or circle: 13
Enter dimension text <150>:
Specify dimension line location or [Mtext/Text/Angle]:

Dim: **diam**

Select arc or circle: 14
Enter dimension text <40>:
Specify dimension line location or [Mtext/Text/Angle]:

Dim: **diam**

Select arc or circle: 15
Enter dimension text <80>:
Specify dimension line location or [Mtext/Text/Angle]:

ИЗУЧЕНИЕ ОСНОВ РАБОТЫ С ГРАФИЧЕСКИМИ РЕДАКТОРАМИ В КУРСЕ «КОМПЬЮТЕРНАЯ ГРАФИКА»

Ю.А. Супрунова

г. Кривой Рог, Криворожский институт Кременчугского университета
экономики, информационных технологий и управления
j@alba.dp.ua

В настоящее время компьютерная графика широко используется при подготовке специалистов различного профиля. Для студентов инженерных специальностей целью обучения, как правило, является освоение конкретной графической программы и выполнение в ее среде заданий профессиональной направленности.

Главной целью инженерного образования является подготовка специалистов, обладающих высоким уровнем профессиональной квалификации, компетентностью в избранном деле и комплексом личностных качеств, актуальных в современных условиях информатизации профессиональной деятельности, представляющих социальную значимость и ценностную потребность для вступающего в трудовую жизнь молодого человека. Чем выше уровень развития профессионально и лично важных качеств у студентов, тем эффективней и качественней их профессиональная подготовка. Компьютерная графика и анимация вызывает у студентов особый интерес, так как дает им возможность реализации собственных замыслов.

Одной из тенденций преподавания компьютерных дисциплин на современном этапе является отказ от технократического подхода в пользу гуманистического. Сущность гуманистического подхода к образовательному процессу состоит в актуализации творческого потенциала, заложенного в личности человека. Процесс обучения должен быть лично ориентированным, так как, развивая личность обучаемого, мы закладываем основу для творческого приложения полученных знаний.

Развитию процессов творческого образно-графического мышления студентов способствуют такие методы, которые активизируют профессиональное становление будущих инженеров благодаря стимулированию их интереса к компьютерной графике. Этому существенно способствует участие в студенческих олимпиадах и конкурсах по графическим информационным технологиям и системам.

Учебный процесс по курсу «Компьютерная графика» строится следующим образом:

1. Ознакомление с теоретической частью курса.
2. Выполнение набора упражнений, представленных в цикле лабораторных работ.
3. Выполнение индивидуального контрольного задания: студенту предлагается исходное сканированное цветное изображение, которое необходи-

мо воспроизвести с максимальной точностью в среде CorelDRAW.

4. Выполнение творческой индивидуальной работы, которая состоит в создании собственного оригинального графического изображения и подробного описания процесса его создания. При этом в качестве главных критериев оценки работы выступает как эстетические критерии, так и сложность технических аспектов реализации проекта.

Такая последовательность изучения материала позволяет студентам получить более прочные знания в области компьютерной графики и прививает практические навыки при работе с векторными и растровыми изображениями в настольных издательских системах.

Вашему вниманию предлагается цикл лабораторных работ, реализованный в виде электронного учебника. Лабораторные работы ориентированы на рассмотрение основных приемов работы в среде графического редактора Corel DRAW. Цикл практических упражнений подобран таким образом, чтобы студенты получили основные навыки создания и обработки растровых и векторных изображений.

В ходе выполнения лабораторных работ студент знакомится с различными инструментами редактора, с трансформациями и эффектами, аспектами работы с трехмерной векторной графикой, а также с обработкой растровых изображений. Особое внимание уделено использованию различных утилит, таких как CorelTRACE (преобразование растровых изображений в векторные) и CorelCAPTURE (формирование растровой копии некоторой области изображения, видимой на экране), входящих в состав пакета CorelDRAW.

Помимо знакомства с программным продуктом, выполнение представленных лабораторных работ позволяет развить творческое воображение студента и познакомить его с основными правилами дизайна, что не может быть лишним в подготовке специалистов любых направлений. Трюки и эффекты, рассмотренные в лабораторных работах актуальны не только для CorelDRAW, но и для многих других графических редакторов.

Представленные лабораторные работы могут быть также использованы и в ходе изучения курса «Информатика и вычислительная техника». Чтобы заинтересовать студента-первокурсника компьютерной техникой, научить его основам работы с интерфейсом любого программного продукта, необходимо сделать занятия максимально творческими и интересными. Для этих целей как нельзя лучше подходит графический редактор CorelDRAW, который, одновременно являясь профессиональным пакетом, тем не менее, обладает достаточно простым и интуитивным интерфейсом, доступным для начинающего пользователя. Использование компьютерной графики на всех этапах преподавания информатики позволяет повысить мотивацию студентов и способствует развитию их творческого потенциала.

Обобщая опыт преподавания дисциплины, хотелось бы отметить следующее. Современная компьютеризированная графическая подготовка в

техническом вузе – это фундаментальная сфера знаний, умений и специфических личностных качеств, без которых не может состояться современный инженер.

Литература:

1. Гурский Ю., Гурская И., Жвалевский А. CorelDRAW 12. Трюки и эффекты. – СПб.: Питер, 2004. – 464 с.
2. Бурлаков М.В. CorelDRAW 12. – СПб.: БХВ-Петербург, 2004. – 688 с.

ДО ПИТАННЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ОЗНАЙОМЧОГО КУРСУ РОБОТИ З ТЕКСТОВИМИ ПРОЦЕСОРАМИ

Р.В. Ліхачов

м. Кривий Ріг, Криворізький металургійний факультет
Національної металургійної академії України

Створіть систему, якою зможуть користуватися навіть дурні, і нею користуватимуться лише вони.

Один з законів Мерфі (Принцип Шоу)

Маємо те, що маємо...

Протягом останніх років ми все частіше стикаємось з великою кількістю книжок, що намагаються допомогти нам у вивченні того чи іншого програмного продукту самостійно. Загальна кількість їх вже давно перевищила ті межі, коли можна сказати, що відсутність або обмаль літератури може негативно вплинути на процес навчання. Проте якість цих видань, звісно ж не у поліграфічному сенсі, не узгоджується з поняттям *навчання*.

Якість друкованого матеріалу не критикував лише ледачий, та кожного разу, коли бачиш на черговій розкладці новеньку, щойно віддруковану книгу, з *політ-коректною* назвою “MS WORD для тих, хто морально розкладається”, починаєш мимоволі скаженіти. Бентежить те, що деякі викладачі спираються в своїй роботі саме на такого роду видання, вважаючи їх чи не найкращим засобом до самовдосконалення студентів і власне себе. Користі від такого навчання стає дедалі менше і жах втратити повноцінну мислячу молодь вже підкрався до нас занадто близько.

Головною причиною нашого невдоволення є поверховість матеріалу, що його викладають автори. Більшість подібних видань містить лише короткі відомості з питання верстки чи мистецтва оформлення документів з точки зору стандартів, про розмаїття шрифтів можна лише здогадуватись. Проте значна кількість сторінок присвячується покроковим алгоритмам (мабуть для читачів з дитячих дошкільних закладів) зміни кольору символів чи їх розміру та оформлення тощо (найчастіше вся ця краса зустрічається одночасно і в одному слові).

Такі покрокові інструкції, що надиктовані викладачем, який не володіє матеріалом в достатній мірі, перетворюють студента під час лекції на зомбі, на дешеву друкарську машинку, чи, будемо сучасніше, на засоби розпізнання та введення тексту.

Розуміння такий підхід не додає анітрішечки, до того ж унеможливорює сприйняття матеріалу в цілому, варто лише студенту відволіктися та втратити бодай найменшу частину конструкції.

Підгрунтя проблем

Приклад для всіх, хто замислюється: довжина окружності визначається формулою $2\pi R$. Деякі викладачі спрощують її до πD , і саме в цьому місті їм варто було б зупинитись.

Придивимось до спрощеної формули уважніше. Проста заміна зробила її краще, коротше та більш приємною для внесення у комплект захлавлених книжечок чи шпаргалок, проте не додала розуміння.

Початкова формула показує зв'язок радіанної міри з лінійною, перетворюючи кут на довжину, вона розповідає нам, як можна, взявши максимально можливий кут у 2π (360°), за допомогою радіусу отримати жадану довжину, унаочнюючи зв'язок *КУТ – ДОВЖИНА*, пов'язані чарівним π .

Якщо студент, що розуміє це, не вагаючись, зможе легко віднайти довжину не лише всієї окружності, а й її частки, перетворивши кут на те, що ми вимірюємо за допомогою лінійки, ми досягли мети. Все просто: хочемо мати чверть довжини, то ж маємо помножити необхідний кут на радіус $R\pi/2$.

Переносячи це на текстові процесори, отримуємо два положення.

По-перше: ні покроковим алгоритмам, ні диктантам, ні зомбуванню.

По-друге: більше базового матеріалу, більше спроб пояснити основи корисних дій, а не їх наслідки.

Приклад з життя: початок пари, студенти мляво заходять до аудиторії, ось-ось почнеться феєрична трагедія під назвою **досидимо, браття**.

Запитую: “Хто знає MS WORD?” (звісно ж жартую), проте кількість піднятих миттєво рук натякає, що жартують наді мною. Фраза з задніх рядів “Краще розкажіть, чи ми гратимемо у COUNTER-STRIKE?” “А як же ж...” – відповідаю завзято, от лише перевіримо чи дійсно всі такі знавці як кажуть?

Запитую, чи стикались коли-небудь з проблемою автозбереження, що її так нав'язливо нам пропонує вище згаданий редактор (нагадаю – це коли неможливо врешті-решт зберегти документ з причин відсутності місця на жорсткому диску, хоча його там в достатку)? Відповідають, що проблему мали, проте вирішення не знайшли, *переклацавши* все можливе, зрештою втратили документ і передруковували всю ніч.

Те, що вирівняти текст по-центру можна правильно, а не за допомогою пробілу, знають лише одиниці, і таких прикладів – безліч.

Кажу їм, де ж глибина поглядів, де самопізнання системи з якою працюєш певний час? Мовчать.

Коротенько про головне

Якого підходу варто дотримуватись у викладенні курсу з *текстових процесорів*.

Останнє визначення є базовим – **ТЕКСТОВІ ПРОЦЕСОРИ**.

Починаю завжди з того, що комп'ютер, та встановлене на ньому ПЗ – це лише інструмент у руках справжнього майстра, а не станок з автоматичним керуванням.

Сторінка, що ми її бачимо в центрі редактору, виглядає однаково, будь вона хоч формату А0, а хоч і А5, тому першою чергою в створенні майбутнього шедедру дивимось у параметри цієї самої сторінки.

Виглядає занадто просто, скажете ви? Ні, не просто, а *фундаментально!* І це закон, для початківців та професіоналів.

Далі йдуть розповіді про те, що комп'ютер – це не друкарська машинка і натискання ENTER'у для формування нового рядка неприпустиме, бо він використовується лише як створювач *абзацу*.

Про те, що можна побачити “те чого нема”, не для того, звісно, щоб довести, що його не існує, а задля можливості ліпше контролювати процес створення документу, та наявність помилок, знають всі, а користуються одиниці – ЗАВАЖАЄ, кажуть. Прозоро натякаю, що не заважає, поступово підводжу до думки, що це варто робити, хоча б інколи, і ось вже з задніх рядів: “...а й дійсно зручно...”, заскрипіли авторучки...

Про те, що все це можливо робити не лише за допомогою миші, спочатку не вірять. *Відпускають* її, як до лав Радянської армії, зі сльозами та іншим антуражем. Кажу ж вам – не жалкуйте, любі мої! Намагаюсь пояснити це так, щоб самі кинули її геть, як погану звичку – зі схліпуванням, та все ж полишають цей продукт нової епохи, цей GIRL-FRIEND всіх молодих гравців, та любителів малювання у PAINT.

Навчаю працювати не відриваючи рук від клавіатури, та спершу головне: НАБІР та ФОРМАТУВАННЯ тексту, це дуже і дуже різні операції, та намагались виконувати їх одночасно призводить до проблем.

Спочатку одне, а вже потім все інше!

Про багатство гарнітур, та їх використання для надання документові належного вигляду, можу розповідати годинами. Слухають і розуміють, перевірено вже багато разів.

Те, що не потрібно перемикати “оті циферки”, аби краще бачити набране у тексті, не вірять, а як бачать, що є можливість масштабування, починають розуміти, що кегль – це інструмент для формування сприйняття тексту, а не для розглядання “горобців” на пів-сторінки і нерозуміння, чому не вміщуються слова у строки.

Напівжирний, курсив та підкреслений – браття на вік.

Розповідаю, чому в англійській версії вони позначаються **В I U**, це допомагає запам'ятати гарячі клавіші, які допомагають використовувати ці можливості форматування.

Все це лише початок, та ті, хто зрозумів суть, підтримують такий формат матеріалу з ознайомлення слухачів з текстовими процесорами.

МЕТОДОЛОГІЧНІ ПІДХОДИ ДО ЗАСТОСУВАННЯ КОМП'ЮТЕРА У НАВЧАЛЬНІЙ ДІЯЛЬНОСТІ СТУДЕНТІВ ІНЖЕНЕРНИХ СПЕЦІАЛЬНОСТЕЙ

І.Є. Фільо

м. Рівне, Національний університет водного господарства
та природокористування

Впровадження сучасних інформаційних технологій в систему вищої освіти вимагає від науково-педагогічних працівників організації і проведення досліджень з питань комп'ютеризації навчального процесу в контексті світових та європейських вимог, для успішного влиття України в Болонський процес.

У становленні і розвитку вузівської комп'ютеризації провідну роль відіграють дослідження Ю.К. Бабанського, В.П. Беспалько, Т.В. Габая, П.Я. Гальперіна, Б.С. Гершунського, А.М. Довгялло, В.Г. Домрачева, І.Я. Лернера, В.Я. Ляудіс, І.В. Марусевої, Ю.І. Машбиця, В.П. Мізінцева, Н.І. Монахова, Л.Ф. Плеухової, І.В. Роберт, А.Я. Савельєва, Ю. Ситникова, Н.Ф. Талізної, Л.В. Трет'якової, В.Ф. Шолоховича, Д.Б. Ельконіна та інших.

Проблеми використання комп'ютерів при вивченні технічних дисциплін досліджували В. Мартинкус, Г. Козлакова, П. Куш, Ю. Таран, Н. Акименко, К. Вузливі, І. Матюшко, М. Бугаєць, В. Биков, А. Васюра, Р. Селезнява. Вплив інформаційних технологій на розвиток інженерної освіти розглядали В. Юрин, М. Носков, Л. Артюхіна, Н. Жбанова, М. Благовещенська, В. Мануйлов, І. Федоров, С. Герасимов, Н. Ємел'янова.

Аналіз наукової літератури [1–3] дозволив виділити деякі проблеми комп'ютеризації інженерної освіти, а саме:

- відсутні або недостатньо висвітлені підходи до створення й упровадження нового покоління інформаційно-програмного забезпечення, зокрема для вищої технічної школи;
- відсутні психолого-педагогічні дослідження щодо ефективності застосування сучасних інформаційних технологій навчання при підготовці студентів інженерних спеціальностей за різними циклами дисциплін і різними ступенями освіти;
- не визначений доцільний рівень забезпечення освітніх закладів дидактичними засобами, комп'ютерною технікою і навчальними комп'ютерними програмами залежно від спеціальності, за якою готується майбутній інженер;
- не визначені підходи до застосування комп'ютера у навчальній діяльності для формування комп'ютерної грамотності та інформаційної культури студентів інженерних спеціальностей.

Потенціал сучасних інформаційних технологій навчання може бути по-

вно реалізований за умов достатньо високої комп'ютерної грамотності та інформаційної культури працівників системи вищої освіти. Необхідність формування такої культури в умовах сучасності вимагає від викладача вмінь: неперервно здійснювати власну самоосвіту та самоудосконалювати професійну майстерність через опанування нових інформаційних технологій; уявляти можливості комп'ютера як засобу навчання; будувати таку методичну систему, що в умовах комп'ютерних технологій навчання буде ефективна на кожному етапі навчального процесу, в кожний момент учбової і навчальної діяльності; добирати, розробляти і вибирати найкращі та найдоцільніші для навчання програмні продукти; вчити студентів ефективно і творчо освоювати і осмислювати здобуту інформацію; готувати студента до повноцінної життєдіяльності в умовах інформатизації суспільства та відповідних знань: новітніх інформаційних технологій та їх функціонування; мережі Інтернет; застосування комп'ютерів в усіх галузях людської діяльності.

Зрозуміло, що інформаційна культура студента, як суб'єкта навчання, формується під впливом тих динамічних змін, що відбуваються в сучасній національній системі вищої освіти. Широке використання обчислювальної техніки у всіх сферах діяльності сучасного інженера пред'являє до його професійної кваліфікації низку додаткових вимог, що полягають в оволодінні новими інформаційними технологіями інженерної праці. Це вимагає від студентів – майбутніх інженерів знань, вмінь і навиків: активно вибудувати свій навчальний процес, вибираючи основну траєкторію в освітньому середовищі; формувати свою інформаційну культуру для ефективного використання інформаційних технологій в майбутній професійній діяльності; розуміти, що для розв'язування не всіх навчальних завдань потрібен комп'ютер і вміти використовувати його лише там, де це дає стійкий рівень засвоєння знань; знаходити, обробляти і зберігати інформацію, використовуючи сучасні інформаційні технології, комп'ютерні комунікаційні системи; вільно орієнтуватися у світовому інформаційному просторі.

Проблему комп'ютерної грамотності деякі автори розглядають вузько, як «оволодіння вміннями розв'язувати задачі на комп'ютері». З цим можна не погодитись навіть в тому, що комп'ютерна грамотність має допомагати студенту – майбутньому інженеру швидко орієнтуватись в інформаційному суспільстві; пристосовувати нові інформаційні технології до своєї галузі діяльності; самостійно і творчо приймати рішення; самоудосконалювати та саморозвивати свої можливості, здібності, уміння. Зрозуміло, що перед вищою технічною школою, дослідниками в напрямку методології використання ЕОМ у навчанні спеціалістів інженерного профілю стоїть ряд питань. Які знання з інформаційних технологій для інженера слід вважати достатніми, враховуючи, що він має бути кваліфікованим користувачем? Якою має бути система освіти інженера, що не є фахівцем у галузі інформаційних технологій та програмування? Що повинен знати і вміти такий інженер-спеціаліст? Вирішення цих питань дає можливість дослідити підходи до застосування

комп'ютера у навчальній діяльності студентів інженерних спеціальностей.



Рис. 1. Формування інформаційної культури та комп'ютерної грамотності студентів інженерних спеціальностей, нефакхівців з інформаційних технологій

Системний та модульний підхід до формування знань, умінь і навичок у процесі набуття комп'ютерної грамотності та інформаційної культури студентів інженерних спеціальностей, нефахівців з інформаційних технологій, можна прослідкувати на рис. 1.

В умовах розвитку інформатизації освіти, комп'ютерна підготовка інженера в технічному університеті повинна проводитись на основі системного та модульного підходу. Формування комп'ютерної грамотності та інформаційної культури студента – майбутнього інженера починається з розширення та поглиблення знань з інформаційних технологій і представляє собою три основних послідовних і неперервних етапи: базова, спеціальна і науково-спеціальна підготовка з інформаційних технологій та додатково може включати до- і післявузівську підготовку. Довузівська підготовка дозволяє формувати у майбутнього студента технічного університету інтерес до професії, професійну інформаційну картину світу. На першому основному етапі (1-2 курс) необхідно сформувати у студента чіткі та стійкі знання, вміння, навички з користування ЕОМ та прикладного програмного забезпечення загального призначення. Для цього розглядаються питання теоретичної інформатики, комп'ютерної технології обробки інформації. Таким чином, майбутній інженер – неспеціаліст в області інформаційних технологій одержує базові знання насамперед користувача. Від нього вимагаються загальні навички роботи з комп'ютером, знання типової структури ЕОМ та уявлення про периферійні пристрої, вміння вільно користуватися операційною системою та програмами-оболонками, вміння користуватися прикладним програмним забезпеченням загального призначення: обробляти числову, текстову і графічну інформацію, працювати з базами даних, працювати в комп'ютерних мережах, розв'язувати задачі з допомогою сучасних систем програмування.

На другому етапі (3-4 (5) курс) необхідно сформувати у студента вміння використовувати базові знання з інформаційних технологій при роботі з прикладними пакетами науково-технічних розрахунків та програм спеціального призначення. Від нього вимагаються вміння з допомогою комп'ютера виконувати різні чисельні, аналітичні математичні та інженерні розрахунки, розв'язувати оптимізаційні задачі, перевіряти статистичні гіпотези, вміти працювати з професійно-орієнтованими пакетами прикладних програм, системами підтримки комп'ютерного експерименту, користуватися засобами автоматизації проектних, конструкторських робіт та технологічних процесів. На етапі спеціально-наукової підготовки інженера (5-6 курс), потрібно формувати у інженера-спеціаліста готовність до професійної діяльності, проведення наукових досліджень зі спеціальності. Підготовка спеціаліста завершується на додатковому етапі післявузівської підготовки за рахунок підвищення кваліфікації та перекваліфікації, формування здатності до постійного удосконалення комп'ютерної грамотності та інформаційної культури.

Резюмуючи вищенаведене, можна стверджувати, що якісна підготовка сучасного спеціаліста-інженера в умовах інформатизації суспільства залежить від неперервного і послідовного освоєння сучасних інформаційних технологій. Разом з тим, засоби обчислювальної техніки повинні підсилювати інженерну підготовку в конкретній предметній області та формувати стійку мотивацію і пізнавальний інтерес до навчання.

Література:

1. Плеухова Л.Ф., Ситников Ю.К. Познавательная деятельность студентов в условиях компьютерного обучения // Педагогика. – 1999. – №7.
2. Козлакова Г.О. Теоретичні і методичні основи застосування інформаційних технологій у вищій технічній освіті: Монографія. – К.: ІЗМН, 1997.
3. Инженер XXI століття: особистість і професіонал в світі гуманізації та гуманітаризації вищої технічної освіти. / За ред. М.Є. Добрускіна: Монографія. – Х.: Рубікон, 1999.

Розділ II

Професійна підготовка вчителя інформатики

НА ШЛЯХУ ІНФОРМАТИЗАЦІЇ ОСВІТИ

Т.П. Кобильник

м. Київ, Національний педагогічний університет імені М.П. Драгоманова
taras2408@mail.ru

Сучасний час у суспільстві характеризується надзвичайно швидким зростанням темпів науково-технічного прогресу, безперервним збільшенням обсягів інформації і зміною змісту знань, умінь і навичок, якими повинні володіти спеціалісти у різних галузях, наслідком чого є підвищення вимог до якості їх підготовки.

Одним з перспективних шляхів підвищення процесу якості навчання є його автоматизація, тобто використання інформаційних технологій навчання. Інформаційні технології – це сукупність методів і засобів реалізації і забезпечення інформаційних процесів в різних галузях людської діяльності з використанням засобів інформатизації (насамперед персональних комп'ютерів).

Швидкі темпи зростання інформаційних ресурсів призвели до протиріччя між створенням і використанням інформації. Як зазначає А.І. Арнольд [1], почали відбуватися парадоксальні випадки: інколи легше винайти щось знову, ніж віднайти креслення і опис раніше винайденого. Наслідком інформаційного надлишку стає дефіцит потрібної інформації. Великий обсяг інформації пришвидшив процес інформатизації сучасного суспільства. Як вважають А.В. Копилов [2], А.І. Ракітов [3], інформатизація – це процес, у якому соціальні, технологічні, економічні, політичні і культурні механізми не просто пов'язані між собою, а є єдиним цілим. Головне в її змісті – якісні перетворення (на основі найновіших технічних та інших досягнень) всього інформаційного середовища життя суспільства з метою оптимізації результатів будь-якої соціально важливої діяльності.

Потрібно зазначити складність і багатозначність терміну “інформатизація”. Щоб зрозуміти значення даного терміну, звернемо увагу на його загальність і значення термінів з подібним закінченням: колективізація, індустріалізація, автоматизація, комп'ютеризація тощо. Як правило, терміни з подібним закінченням часто означають складні соціально-економічні, науково-технічні і суспільно-політичні процеси активізації конкретних сфер людської діяльності, викликані потребами суспільства на певному етапі його розвитку і пред'являє до працівників даної галузі особливі вимоги.

Проведений аналіз терміну “інформатизація” показує, що інформатизація – це процес підвищення ефективності застосування інформації у суспільстві з допомогою інформаційних технологій. Враховуючи багатоаспектність і всеохоплюваність даного явища, можна дати ряд визначень в залежності від конкретної предметної області інформатизації.

Сьогодні надзвичайно велике значення для суспільства має інформати-

зація освіти. Це, насамперед, пов'язано з тим, що наше майбутнє головним чином залежить від долі освіти.

Інформатизація освіти являє собою систему методів, процесів і програмно-технічних засобів, об'єднаних з метою збору, зберігання, розповсюдження і використання інформації. Мета інформатизації освіти полягає в глобальній інтенсифікації інтелектуальної діяльності за рахунок використання інформаційних технологій.

Серед багатьох факторів, які визначають взаємозв'язки між системою освіти та інформаційними технологіями, найбільш важливими, на думку багатьох дослідників, є наступні: усвідомлення фундаментальної ролі інформації, збільшення обсягів інформації, розвиток інформаційної техніки та технології, становлення інформаційного суспільства. Розглянемо кожен з цих факторів детальніше.

Усвідомлення фундаментальної ролі інформації у житті людства, можливість вирішення глобальних проблем на основі використання інформації та знань, виникнення негативних наслідків у випадку нехтування інформацією при прийнятті рішень – все це повинно знайти відображення у роботі всіх ланок системи освіти.

У зв'язку з цим стає очевидною необхідність створення нових педагогічних технологій, спрямованих на формування інформаційного світогляду особистості, яка живе і працює в сучасному, якісно новому інформаційному середовищі, усвідомлює суть інформаційних перетворень, які відбуваються, здатною ефективно використовувати інформаційні технології. Окремим завданням при цьому є навчання раціональним прийомам роботи з інформаційними ресурсами, включаючи технологію і алгоритми пошуку, відбору, аналізу інформації.

Результатом швидкого збільшення обсягів інформації став “інформаційний вибух”, одним з наслідків якого є інформаційна криза, суть якої полягає в тому, що при надлишку інформації у суспільстві в цілому окрема особистість має великі труднощі при отриманні доступу до необхідної їй інформації і вимушена подолати ряд перепон, обумовлених законами розсіювання і старіння інформації, термінологічними, мовними та іншими інформаційними бар'єрами. Подолання цих бар'єрів неможливе без знань про властивості інформації та інформаційних ресурсів, закономірності поширення і способів доступу до них. Тому вже на етапі навчання майбутній спеціаліст повинен отримати інформацію про інформацію.

Протягом усієї історії людства процес постійного зростання ролі інформації і збільшення її обсягів, необхідних для забезпечення життєдіяльності суспільства, супроводжувався розвитком і вдосконаленням інформаційної техніки і технології. Однією з найбільш яскравих ілюстрацій стрімкого впровадження в життя інформаційних технологій є Internet. Internet характеризується небувалими в історії комунікаційних технологій темпами росту. При цьому особливу увагу користувачів (учнів, студентів) слід звернути на

роботу з інформаційно-пошуковими системами Internet. Під час першого знайомства з ними у користувачів може скластися хибне уявлення про те, що в Internet є будь-яка інформація зі всіх питань. Під час подальшої роботи користувач поступово розчарується в Internet, вважаючи, що там є будь-яка інформація, але не та, яка йому потрібна. Часто подібне розчарування виникає у зв'язку з тим, що користувач не знає, як правильно працювати з інформаційно-пошуковими системами і як правильно формулювати пошукові запити. Більшість користувачів проводять пошук, складаючи прості запити з ключових слів, зовсім не використовуючи логічних операторів і спеціальних слів, які дозволяють формувати складні запити.

Інша причина розчарування – мовний бар'єр. Більшість користувачів не знає англійської мови на достатньому рівні, у зв'язку з чим англійські інформаційні ресурси ними не використовуються, а україномовні разом з російськомовними складають тільки декілька відсотків від світових ресурсів Internet. Тому в процесі навчання користувачів з інформаційно-пошуковими системами необхідно пропонувати шукати інформацію з тієї проблематики, яка найбільш відображена в україномовній та російськомовній частині Internet.

Майбутні спеціалісти також повинні отримувати уявлення про інформаційні ресурси Internet з проблем їхньої майбутньої професії; вони повинні знати найбільш популярні Web-сайти з питань освіти і науки; вміти виконувати пошук в електронних каталогах і базах даних інформаційних центрів, які займаються проблемами їхньої професійної діяльності. Особливу увагу слід звернути на те, що в Internet є багато неякісної (недостовірної) інформації, у зв'язку з чим користувач повинен вміти проводити відбір у знайдений інформації.

Багато дослідників позв'язують долю суспільства з розвитком інформаційних ресурсів. При цьому називаються дві формальні ознаки вступу в інформаційне суспільство: по-перше, більше половини трудових ресурсів зайняті у створенні та обробці інформації, по-друге, побудована система глобальної комунікації, до якої відкритий доступ для всіх завдяки персональним комп'ютерам.

В інформаційному суспільстві важлива роль належить комп'ютеризованому навчанню – педагогічній системі, основними складовими якої є педагог, комп'ютерні навчаючі програми і звичайні засоби навчання [4].

Згідно об'єктивно діючого закону, інформація з часом старіє. У зв'язку з цим надзвичайно важливо сформулювати у майбутніх спеціалістів уявлення про діалогічність, варіативність, відкритість знань, виробити потребу у неперервній освіті. Необхідно навчити їх творчо використовувати отриману інформацію при прийнятті компетентних управлінських рішень з врахуванням економічних, екологічних, моральних та естетичних аспектів у процесі своєї професійної діяльності. Розв'язання цих та інших проблем можливе в рамках формування інформаційної культури спеціаліста.

В Україні прийнята Концепція інформатизації загальноосвітніх навчальних закладів, комп'ютеризації сільських шкіл [5], де зазначено, що головною метою інформатизації загальноосвітніх навчальних закладів є підготовка підростаючого покоління до повноцінної плідної життєдіяльності в інформатизованому суспільстві, підвищення якості, доступності та ефективності освіти. Реалізація головної мети інформатизації загальноосвітніх навчальних закладів передбачає досягнення таких підцілей:

– формування інформаційної культури учнів, яка стає сьогодні невід'ємною складовою загальної культури кожної людини і суспільства в цілому;

– рівнева і профільна диференціація навчально-виховного процесу з метою якомога повнішого розвитку нахилів і здібностей дітей, задоволення їх запитів і потреб, розкриття їхнього творчого потенціалу; удосконалення управління освітою; підвищення ефективності наукових досліджень.

Процес інформатизації шкільної освіти спрямований на розв'язання фундаментальних проблем освіти, таких як протиріччя між збільшенням обсягу змісту навчання і обмеженою кількістю навчального часу; зменшення частки об'єму шкільних знань учнів відносно знань, отриманих поза школою; необхідність врахування психофізіологічного розвитку учнів та їх індивідуальних можливостей до навчання.

Інформатизація освіти створює передумови для широкого впровадження в практику психолого-педагогічних розробок, які забезпечують перехід від механічного засвоєння знань до оволодіння вміннями самостійно набувати нові знання; дозволяє підвищити рівень науковості шкільного експерименту, наблизивши його методи і організаційні форми до експериментально-дослідницьких методів наук, що вивчаються; забезпечує залучення до сучасних методів роботи з інформацією, інтелектуалізацію навчальної діяльності.

У системі освіти вже досить значну роль відіграють інформаційні технології. Метою їх застосування є підвищення інтелектуальних можливостей людини в інформаційному суспільстві, а також інтенсифікація і підвищення якості навчання на всіх ступенях системи освіти.

Інформаційна технологія навчання – це цілісна навчальна система, яка являє собою інтеграцію технічного, дидактичного, інформаційно-освітнього середовища, що забезпечує виконання спільних дій суб'єктів навчальної діяльності в умовах інформатизації освіти, спрямованих на досягнення результатів навчання. До них відносять усі технології, які використовують спеціальні технічні інформаційні засоби (персональний комп'ютер, аудіо, відео, кіно). Вони дають можливість створювати, зберігати, обробляти, передавати інформацію і забезпечувати ефективні способи її подання суб'єкту навчального процесу.

Комп'ютер як елемент інформаційних технологій дозволяє забезпечити доступ практично до будь-якої інформації за лічені хвилини, секунди, що

може принципово змінити підхід до проблеми шкільної освіти і самоосвіти, до вибору методів та організаційних форм навчання.

Процес впровадження інформаційних технологій в навчання достатньо складний і потребує фундаментального осмислення. Застосовуючи комп'ютер, необхідно слідкувати, щоб учень (студент) не перетворився на автомат, який уміє думати тільки за запропонованим йому алгоритмом. Для подолання цієї проблеми необхідно разом з інформаційними технологіями навчання застосовувати й традиційні. Навчаючі і контролюючі програми повинні надавати користувачу можливість побудови свого власного алгоритму дій. Завдяки цьому учень (студент) починає систематизувати і застосовувати свої знання в реальних умовах.

Впровадження інформаційних технологій навчання вимагає вирішення перш за все психолого-педагогічних, матеріально-технічних, науково-методичних, організаційних проблем. До їх вирішення повинні долучитися Міністерство освіти і науки України, вищі навчальні заклади з кафедрами інформатики та методику викладання певних предметів, загальноосвітні установи. Проте використанням комп'ютерної техніки на уроках не потрібно зловживати, оскільки це може призвести до негативних наслідків навчання. Тільки вчитель повинен вирішувати, коли доцільно застосовувати інформаційні технології, а коли використовувати традиційні засоби і методи навчання, а це, в свою чергу, вимагає від нього відповідної підготовки.

Література:

1. Арнольдов А.И. Информация – общечеловеческая ценность // Проблемы информатизации культуры: Сб. статей. – Вып. 3. – М.: 1996. – С. 13–24.
2. Копылов В.А. Еще раз о термине “информатизация” // НТИ, Сер. 1. – 1994. – № 8. – С. 4–7.
3. Ракитов А.И. Философия компьютерной революции. – М.: Политиздат, 1991. – 288 с.
4. Захарова И.Г. Информационные технологии в образовании. – М.: Академия, 2003. – 212 с.
5. Концепція програми інформатизації загальноосвітніх навчальних закладів, комп'ютеризації сільських шкіл // Комп'ютер в школі та сім'ї. – 2000. – №3. – С. 3–10.

ОПТИМАЛЬНИЙ ВИБІР МЕТОДІВ НАВЧАННЯ З МЕТОЮ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОЦЕСУ НАВЧАННЯ НА УРОКАХ ІНФОРМАТИКИ

Л.Л. Сушенцева, О.Ю. Штиркова
м. Кривий Ріг, Криворізький технічний університет

Постановка проблеми. З кожним роком все більше і більше підприємств, фірм, організацій переходять до комп'ютеризації виробничих процесів. Тому в наш час користуватися комп'ютером необхідно вміти кожній людині, і вища освіта не може залишитися осторонь від цієї проблеми. Разом з тим прикро усвідомлювати, що значна частина випускників вищих навчальних закладів I та II рівнів акредитації мають недостатні вміння і навички роботи з комп'ютером. Однією з причин, на наш погляд, є недостатня методична підготовка викладачів інформатики.

Як показує практика, викладачі інформатики досить компетентні з фахових дисциплін, але недостатньо володіють методикою викладання. Наші спостереження свідчать, що викладачі інформатики часто одноманітно проводять заняття, використовуючи пасивні методи навчання (розповідь, робота з літературою, тощо), не надають належної уваги мотивації пізнавальної діяльності та актуалізації опорних знань учнів. На нашу думку, досягнути більш повного та глибокого осмислення і засвоєння учнями знань, вмінь, навичок та професійної майстерності при роботі з комп'ютером можливо лише за умови доцільного вибору методів навчання. Означена проблема цікавила і цікавить як науковців, так і педагогів-практиків (В.С. Безрукава, Н.П. Волкова, М.А. Жиделев, Н.Г. Ничкало, А.В. Петровський, М.М. Фіцула, В.В. Ягупов та багато інших). Так, В.В.Ягупов вважає, що «методи навчання безпосередньо формують ... характер взаємин викладачів і учнів, суттєво впливають на формування суб'єктних відносин між ними» [4, 318].

Метою даної статті є розкриття оптимального вибору методів навчання з метою підвищення ефективності навчання на уроках інформатики.

Аналіз джерел та публікацій. Загально відомо, що метод - це спосіб або система прийомів для досягнення певної мети, шлях до мети. Багато науковців інтерпретують поняття «методи навчання» по-різному. Як зауважує Н.П. Волкова, метод навчання є способом керівництва пізнавальною діяльністю учнів, що має виконувати три функції: навчаючу, виховну і розвиваючу [2]. Отже, методи навчання – це способи взаємодії між викладачем і учнями по досягненню мети навчання.

Вивчення психолого-педагогічної літератури показує, що єдиної класифікації методів навчання немає. Наш науковий інтерес викликали дослідження Н.Г. Ничкало та Ю.К. Бабанського. Так, Ю.К. Бабанський поділяє методи навчання на методи організації і здійснення навчально-пізнавальної діяльності, методи стимулювання і мотивації учіння та методи контролю і

самоконтролю у навчанні [1].

Оскільки методи організації і здійснення виконують освітню функцію Н.Г. Ничкало поділяє їх на перспективні методи та методи, виділені за логічною ознакою.

Дотримуючись означеної класифікації, науковець відносить до перспективних методів: – словесні (розповідь, пояснення, бесіда та інші), наочні (демонстрація, показ операцій або процесів, спостереження учнів тощо), практичні (вправи, лабораторні роботи, розв'язання технічних завдань).

Методи виділені за логічною ознакою Н.Г. Ничкало поділяє на репродуктивні (запам'ятовування учнями навчального матеріалу) та проблемно-пошукові (передбачають створення викладачем проблемної ситуації, активне їх обдумування учнями та знаходження виходу або рішення ситуації) [3].

Педагогу-практику важко вибрати з усього різноманіття методів та їх класифікацій найбільш оптимальні. На нашу думку, знання педагогом критеріїв вибору методів навчання буде сприяти розв'язанню цієї проблеми. До таких критеріїв слід відносити мету навчання, рівень мотивації навчання, об'єм та рівень складності навчального матеріалу, рівень підготовленості учнів, вікові та індивідуальні особливості учнів, матеріально-технічні умови навчання, тип та структуру заняття, особистісні якості та рівень підготовленості викладача.

Залежно від поставленої мети уроку слід обирати метод навчання. Якщо метою уроку є надання певних відомостей із теми, то можна використовувати словесні методи, якщо метою є формування вмінь та навичок – бажано використовувати практичні та проблемно-пошукові методи.

Відомо, що деякі з викладачів дуже часто використовують метод самостійної роботи учнів, роботу з книгою. В наш час не всі учні забезпечені необхідними підручниками. Тому, викладачу слід розробляти певні теоретичні відомості у вигляді методичний рекомендацій або порад. Оскільки, мова йде про викладача саме інформатики, то можна розробляти представлення теоретичного матеріалу як дидактичного засобу навчання – у вигляді простого тексту у текстовому редакторі, у вигляді презентації тощо. Тим самим, учні будуть постійно працювати за комп'ютером, що дасть змогу вдосконалювати їх вміння й навички при роботі з ним. Якщо кабінет інформатики має відповідне матеріально-технічне оснащення, викладач може покроково виводити інформацію на екран і одночасно пояснювати та наводити приклади.

Слід враховувати індивідуальні та вікові особливості учнів. Для високого рівня підготовленості учнів слід обирати такі методи навчання, як проблемні, евристичні та дослідницькі. Для менш сильних – розповідь, бесіда, інструктажі. Оскільки, кожен учень – це перш за все особистість, яка має певний тип і швидкість мислення, сприйняття. Тому, ми пропонуємо на уроках інформатики на етапі закріплення знань або актуалізації опорних знань використовувати картки-завдання, які включають три рівні складнос-

ті. Перший рівень – «легкий» – може бути у вигляді тестів (приблизно 10 питань) або запитань (наприклад: дайте визначення поняттю ...). Другий – «середній» – практичне завдання (виконати певні дії на комп'ютері), які б включали необхідні знання, вміння та навички. Третій – «високий» – у вигляді проблемної ситуації чи завдання творчого характеру.

Викладач також повинен враховувати об'єм та рівень складності навчального матеріалу. Якщо вивчаються більш легкі теми та розділи (наприклад, текстовий редактор, табличний процесор), то можна обирати словесні, практичні або проблемно-пошукові методи. Спочатку можна розповісти, потім дати конкретне завдання, згодом поставити проблему і запропонувати учням знайти рішення. Якщо вивчаються більш складні або ж великі за обсягом теми та розділи (наприклад, робота з базами даних, логічні мови програмування), слід використовувати методи стимулювання і мотивації учіння, методи контролю і самоконтролю у навчанні, обов'язково наочні, практичні, репродуктивні та проблемно-пошукові методи навчання.

Вивчення передового досвіду і власна педагогічна практика свідчать, що результативність навчання у значній мірі залежить від інтересу учня по предмету вивчення, від його мотивів. Тому, на етапі мотивації пізнавальної діяльності слід орієнтувати учнів на практичний сенс теми, а також сприяти підвищенню оцінки особистості в очах оточуючих. Як відомо, більш вдалою є здійснення мотивації у процесі розповіді, спрямованої на майбутню професійну діяльність учнів. Викладення нового матеріалу бажано подавати з використанням комплексу методів. Тобто, викладач пояснюючи новий матеріал повинен весь час пов'язувати його з практикою, постійно наводячи приклади (бажано із життя). Далі рекомендовано дати практичну або лабораторну роботу. Завдання лабораторної роботи мають бути розміщені у порядку від простого до складного. На нашу думку, саме поступовість і ускладнення завдань дають змогу більш ефективно формувати в учнів професійні вміння та навички. При захисті своїх робіт учням бажано запропонувати відповісти на контрольні запитання до лабораторної роботи, що складені відповідно до теоретичного матеріалу та поставити проблемне питання або завдання.

Залежно від типу уроку можна використовувати різні методи навчання. Так, при традиційних уроках (урок засвоєння нових знань, урок формування вмінь та навичок, урок застосування знань, вмінь та навичок, урок узагальнення та систематизації знань, комбінований урок) можна використовувати різні методи навчання. На нетрадиційних уроках (урок-змагання, урок-подорож, урок-дослідження, уроки-ділові або рольові ігри) рекомендуємо використовувати методи логічного спрямування, а саме проблемно-пошукові.

У своїй роботі для досягнення поставленої мети викладач інформатики повинен поєднувати як традиційні, так і нетрадиційні типи уроків. Ми рекомендуємо на початку вивчення тем проводити стандартні уроки засвоєння

нових знань, формування вмінь та навичок або ж просто урок-лекцію. А уроки практичного спрямування, де учні застосовували б свої знання, бажано проводити нетрадиційно. Як показує педагогічна практика, більш ефективно учням запам'ятовуються ті уроки, на яких вони логічно мислять і одночасно грають. При вивченні лексичного матеріалу було б корисно, щоб він був підкріплений наочністю (картки, малюнки, слайди, репродукції і т.д.), а потім використаний у практичних вправах.

Продовжуючи думку про комплексне використання методів навчання на уроці, рекомендуємо алгоритмізувати дії викладача інформатики при оптимізації вибору методів навчання:

- прийняти рішення про вибір методів самостійного вивчення теми учнями чи вивчення її під керівництвом викладача;
- прийняти рішення про вибір репродуктивних чи пошукових методів;
- прийняти рішення про вибір індуктивних чи дедуктивних методів навчання;
- прийняти рішення про вибір поєднання словесних, наочних чи практичних методів навчання;
- прийняти рішення про вибір методів стимулювання навчальної діяльності;
- прийняти рішення про вибір методів контролю та самоконтролю;
- продумати запасні варіанти поєднання методів у випадку відхилень і невідповідностей учнів за результатами виконання домашнього завдання та повторення пройденого (труднощі у виконанні домашнього завдання, питання, що виникають щодо раніше вивченого, висока втомлюваність учнів тощо).

Висновок. Отже, для оптимізації вибору методів навчання на уроках інформатики з метою підвищення ефективності процесу навчання слід враховувати ряд критеріїв, а саме: мету навчання, рівень мотивації навчання, об'єм та складність навчального матеріалу, рівень підготовленості учнів, вікові та індивідуальні особливості учнів, матеріально-технічні умови навчання, тип і структуру заняття та особистісні якості й рівень підготовленості викладача. Здійснювати вибір методів навчання необхідно за певним алгоритмом дій.

Література:

1. Бабанский Ю.К. Оптимизация процесса обучения. – М.: Педагогика, 1977.
2. Волкова Н.П. Педагогіка: Посібник для студентів вищих навчальних закладів. – К.: Академія, 2001. – 576 с.
3. Педагогічна книга майстра виробничого навчання: Навч.-методичний посібник / За ред. Н.Г. Ничкало. – К.: Вища шк., 1994. – 383 с.
4. Ягупов В.В. Педагогіка: навчальний посібник. – К.: Либідь, 2002. – 560 с.

УПРАВЛІННЯ ПІДГОТОВКОЮ МАЙБУТНЬОГО ВИКЛАДАЧА СПЕЦІАЛЬНИХ ДИСЦИПЛІН

Т.В. Волкова

м. Бердянськ, Бердянський державний педагогічний університет
імені П.Д. Осипенко

Організовуючи управління підготовкою майбутнього викладача спеціальних дисциплін професійно-технічного навчального закладу, виходимо з досвіду використання у дидактиці технологій навчання (В.П. Беспалько, В.В. Гузеєв, В.М. Монахов, О.В. Сергєєв, С.О. Смирнов та інші). Враховуючи необхідність забезпечення систематичності й фундаментальності знань, розвитку особистості майбутнього викладача відповідно до пріоритетних завдань підготовки, а також те, що цей процес відбувається в період інформатизації освіти і те, що учасником його є майбутній викладач, який має достатню як педагогічну, так і комп'ютерно-інформаційну підготовку, а також інтегруюче значення інформаційного середовища, у якому постійно відбувається весь процес цієї підготовки, слід визнати, що найбільш продуктивним є управління на основі сучасних інформаційних технологій, які дозволяють найбільшою мірою забезпечити мобільність і варіативність індивідуальної та групової навчально-пізнавальної діяльності майбутнього викладача (В.Ю. Биков, Н.В. Апатова, Ю.О. Дорошенко, Н.В. Морзе, Д.Ш. Матрос, І.В. Роберт).

Суть нашого підходу полягає в максимальному керівництві процесом інтеграції педагогічної та комп'ютерно-інформаційної підготовки. Це, у свою чергу, вимагає дотримання основних принципів педагогічної технології, зокрема проектування, відтворення, цілеутворення, цілісності, і передбачає алгоритмічність процесу відбору форм, методів і засобів відповідно до обраного змісту, які найбільше відповідають меті та рівню готовності майбутнього викладача до професійної діяльності. Враховуючи керованість процесу інтеграції, виділяємо як головну ланку організацію гнучкої діяльності майбутнього викладача на основі теорії планомірного, поетапного формування розумових дій і понять.

Особливістю професійної підготовки майбутнього викладача є використання модульного навчання, яке передбачає вивчення п'яти модулів.

Основними етапами методики професійного навчання у межах кожного з них є виклад теоретичного матеріалу укрупненим блоком, розв'язання типових професійно спрямованих завдань (інваріантний блок змісту), особистісно зорієнтоване навчання (варіативний блок змісту), підсумкове узагальнення тематичного модуля та контроль-корекція.

Самостійна робота майбутнього викладача є метою і умовою успішного професійного навчання і до майбутньої педагогічної практики та виконання курсового проектування. Навички самостійної діяльності формуються за-

вдяки переходу від повного управління пізнавальною діяльністю з боку викладача до дозованої допомоги і самоуправління за допомогою засобів інформаційно-комунікаційних технологій, а потім – до творчого самостійного розв’язання професійно значущих завдань за допомогою методів і засобів інформатики.

Методика передбачає вхідний контроль при вивченні кожного тематичного модуля і відповідну корекцію змісту, поточний контроль при вивченні елементів інваріантної та варіативної частин (результати роботи семінару-практикуму і завдань самостійної роботи, яка виконується кожним майбутнім викладачем), підсумковий контроль у формі захисту проекту при завершенні вивчення модуля (результати роботи у вигляді електронного портфеля студента, демонстрація їх за допомогою розробленої мультимедійної презентації); контроль міцності знань, умінь і навичок при виконанні професійно спрямованої навчально-пізнавальної та науково-дослідницької роботи. На рис. 1 представлено основні форми організації професійної підготовки майбутнього викладача.

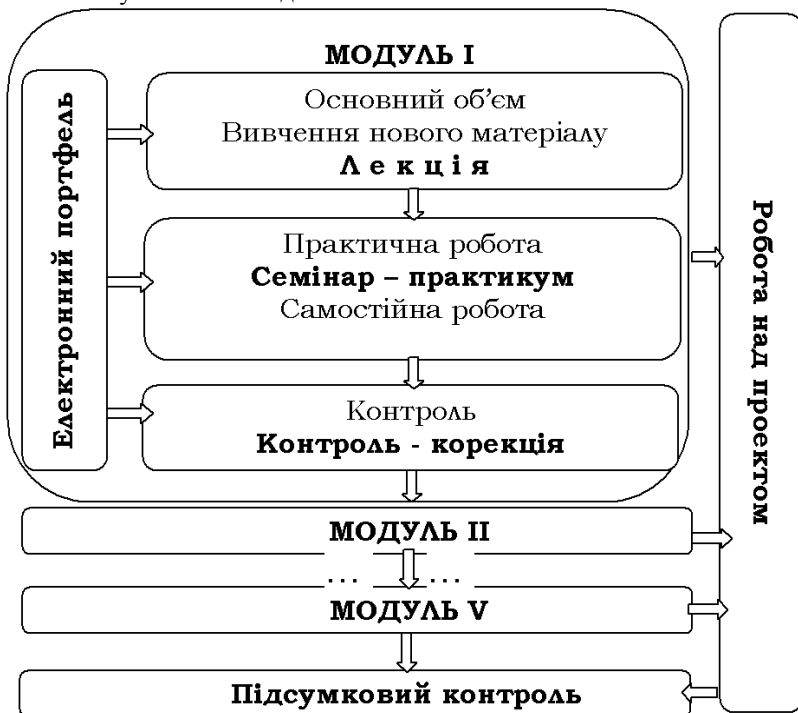


Рис. 1

Лекція є першою і дуже важливою ланкою у ланцюжку формування знання-розуміння й умінь. На лекції закладаються концептуальні теорети-

ко-методологічні основи для подальшого самостійного опанування знань. Ця форма навчання використовується насамперед для формування мотивації, орієнтуру на досягнення мети навчання кожного модуля. Оскільки концентрація необхідних психічних характеристик майбутнього викладача на об'єктах пізнавальної діяльності можлива лише за наявності усвідомленої мети, розуміння місця предмета вивчення у професійній галузі, лекція мусить орієнтувати майбутнього викладача і у цьому напрямку, надаючи можливість на базі фундаментальних знань розв'язувати професійні завдання.

Наступний етап передбачає розв'язання типових професійних завдань, які є основою наступної викладацької діяльності. На кожному з етапів засвоєння вона виконують різні функції. Так, на етапі виконання матеріалізованих дій завдання є метою створення початкової мотивації діяльності, на етапі ознайомлення з орієнтовною основою дії сприяють активному розкриттю цієї діяльності, на етапі виконання діяльності – формують необхідні характеристики дії (аналіз та синтез, узагальнення, класифікація, порівняння, абстрагування, аналогія). Система завдань сконструйована таким чином, щоб у процесі її розв'язання у майбутнього викладача сформувались компоненти професійної компетентності.

У ролі таких завдань може виступати комплекс методичних завдань, які майбутній викладач розв'язує на практичних заняттях (інваріантна частина) та під час самостійної роботи (варіативна частина). Систематичне і цілеспрямоване застосування методичних завдань у навчальній діяльності майбутнього викладача забезпечує формування у нього узагальнених методичних умінь, а також розвиток індивідуальних професійних та особистісних якостей (інтерес до професійної діяльності, творчу активність, ініціативу та самостійність методів і засобів навчання). Методичні завдання є особливим видом професійно-педагогічних завдань і за своїми функціями, метою, структурою змісту, характеристикам і результатами розв'язання є навчальними. Методичним є завдання, результатом розв'язання якої є отримання методичного факту у вигляді різних методичних розробок з конкретного предмета. Формою представлення результатів розв'язку такого виду завдань є представлені майбутнім викладачем логічні або знакові конструкції, що задають певний план дій у вигляді методичних прийомів, методичних принципів, правил навчальної діяльності, методів навчання, методик тощо.

У методичному завданні, як і в будь-якому іншому, можна виділити такі структурні елементи: предметну галузь, яка визначається конкретною технічною дисципліною; вимогу завдання, що задається умовою і метою його розв'язання; операційну основу (сукупність дій – операцій, які слід виконати над умовою задачі, щоб задовольнити її вимогу – Г.О. Балл, Н.Ф. Гализіна, Д.Б. Ельконін, В.І. Загв'язинський, Ю.І. Машбиць). Зміст методичного завдання відображає ту чи іншу сторону професійної діяльності майбутнього викладача. Для розв'язання методичного завдання необхідно встановити міжпредметні зв'язки, *інтегрувати* знання кількох наук. Залеж-

но від змісту та вимоги завдання можуть формувати різні за ступенем узагальненості методичні уміння. Таким чином, результатом розв'язання методичних завдань у процесі навчання майбутнього викладача є засвоєння методичних знань у діяльності, що характеризує більш високий рівень узагальнення з педагогіки, психології, інформаційних та комп'ютерних технологій; послідовне зростання об'єму методичних умінь, що формуються, та оволодіння прийомами самооцінки результатів власної діяльності; оцінка набутого досвіду методичної діяльності та вироблення ставлення до обраної спеціальності; залучення до методичної творчості.

Отже, методичне завдання є формою організації навчальної діяльності, спрямованої на розвиток узагальнених професійних умінь і формування професійної компетентності майбутнього викладача спеціальних дисциплін. Застосування методичних завдань дозволяє досягти високого рівня професійної компетентності при реалізації принципів методичної підготовки, системного підходу до неї, застосування відповідних технологій. У системі професійної підготовки майбутнього викладача актуального значення набуває проблема самостійного перенесення методичних умінь без систематичного консультування з боку викладача. Сконструйований та спланований процес формування системи методичних знань та умінь має включати прийоми і методи, які ведуть до самонавчання. У табл. 1 наведено зміст методичних завдань при вивченні майбутнім викладачем інформаційно-комунікаційного модуля.

Таблиця 1

Завдання для практичної та самостійної роботи майбутнього викладача

Практична робота	Самостійна робота
<i>Тема змістового елемента 1.1 Концептуальні положення професійного навчання робітників комп'ютеризованого виробництва.</i>	
1.1П. Методологічна основа методики професійного навчання. Аналіз професій групи адміністрування баз даних.	<p><i>1.1.1С. Обґрунтувати доцільність уведення нової спеціальності, потребу її на ринку праці.</i></p> <p>1. Проаналізувати узагальнені види діяльності та компетенції кожної сфери діяльності фахівця у галузі автоматизації.</p> <p>2. Проаналізувати найменування професій, спеціальностей, кваліфікацій у галузі автоматизації.</p>
	<p><i>1.1.2С. Розробити типові завдання діяльності оператора автоматизованого робочого місця.</i></p> <p>1. Для професії «Оператор» згідно з Довідником кваліфікаційних професій виписати зміст розділів «Завдання та обов'язки», «Має знати», «Кваліфікаційні вимоги», «Орієнтовний перелік питань для атестації з професії».</p> <p>2. Розглянути схему діяльності оператора (рис. 2), проаналізувати психофізіологічні основи цієї дія-</p>

Практична робота	Самостійна робота						
	<p>льності, визначити функції та засоби діяльності (табл. 2).</p> <p>Рис 2. Схема діяльності оператора у системі “Людина-машина”.</p> <p style="text-align: right;">Таблиця 2</p> <p>Типові завдання та способи діяльності оператора</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Функції діяльності</th> <th style="width: 50%;">Засоби діяльності</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Функції діяльності	Засоби діяльності				
Функції діяльності	Засоби діяльності						

Тема змістового елемента 1.2. Методика створення електронного портфеля майбутнього викладача.

<p>1.2.П. Електронна технологія портфеля майбутнього викладача Структура портфеля майбутнього викладача. Правила оформлення та правила презентації.</p>	<p>1.2.1С. Дослідити інформаційних процесів у біологічних, соціальних і технічних системах 1. Виділити та проаналізувати інформаційні процеси у біологічних, соціальних та технічних системах. 2. Оцінити інформацію з позицій її властивостей, визначити необхідну для розв’язання задачі. 3. Визначити класи завдань, які можна розв’язувати за допомогою існуючої інформації.</p>
---	--

Тема змістового елемента 1.3. Основи системного підходу.

<p>1.3.П. Система. Компоненти системи. Управління системою як інформаційний процес.</p>	<p>1.3.1С. Визначити загальні особливості управління. 1. Проаналізувати мету управління при роботі з апаратними та програмними засобами комп’ютера. 2. Скласти алгоритми розв’язання певного класу задач та до кожного алгоритму – блок-схему.</p>
---	--

Тема змістового елемента 1.4. Наукові основи інформаційних технологій.

<p>1.4П. Аналіз інформаційних завдань управління, що реалізують інформаційний процес. Типи завдань за призначенням</p>	<p>1.4.1С. Дослідити за допомогою інформаційних моделей структуру і поведінку об’єкта відповідно до поставленого завдання 1. Вивчити структуру об’єкта. 2. Формалізувати інформацію певного виду. 3. Структурувати дані та знання при розв’язуванні</p>
--	---

Практична робота	Самостійна робота
<p>1) регламентні інформаційно-довідкового призначення</p> <p>2) економічного аналізу</p> <p>3) поглибленого економічного аналізу</p> <p>4) оптимізаційні економічні задачі</p>	<p>завдань.</p> <p>3. Скласти ділові папери за певною формою. <i>Типи завдань</i></p> <p>1. Створення і заповнення бази даних у СУБД.</p> <p>2. Редагування бази даних.</p> <p>3. Формування запитів і вибір за запитом. <i>Типи завдань</i></p> <p>1. Створення і відправка електронної кореспонденції.</p> <p>2. Пошук і відбір інформації в мережі <i>Типи завдань</i></p> <p>1. Набір тексту, збереження документа.</p> <p>2. Оформлення тексту списками і колонками.</p> <p>3. Робота з редактором формул.</p> <p>4. Шаблони документів.</p> <p>5. Створення і заповнення таблиць.</p> <p>6. Використання графічних об'єктів. <i>Типи завдань</i></p> <p>1. Створення і заповнення табличних даних.</p> <p>2. Умовна функція та логічні вирази.</p> <p>3. Практичне використання табличного процесора для розв'язання економічних завдань.</p>
<p><i>Тема змістового елемента 1.5. Створення гіпертекстових інформаційних моделей.</i></p>	
<p>1.5П. Створення гіпертекстових інформаційних моделей</p>	<p><i>1.5.1С. Побудувати гіпертекстову модель певного класу об'єктів.</i></p> <p>1. Виділити клас із запропонованих (інформація, знання, інформаційна культура, інформаційно-комунікаційна компетентність, інформаційна безпека, інформаційний продукт, інформаційний процес, автоматизоване робоче місце, електронний офіс, технологія «Клієнт-сервер»).</p> <p>2. Побудувати гіпертекстову модель виділеного класу об'єктів. Визначити смислові зв'язки між об'єктами.</p>
<p><i>Тема змістового елемента 1.6. Інформаційні та комунікаційні технології навчання. Програма Інтел «Навчання для майбутнього».</i></p>	
<p>1.6П. Міжпредметні проектні роботи для майбутнього робітника професійно-технічного навчального закладу.</p>	<p><i>1.6.1С. Технологія презентацій.</i></p> <p>1. Розробити сценарій пробного уроку зі спецтехнології для майбутнього робітника автоматизованого виробництва.</p> <p>2. Реалізувати методичну розробку за допомогою</p>

Практична робота	Самостійна робота
	технології презентацій. 1.6.2С. <i>Ознайомитися з технологією дистанційної освіти.</i> 1. Визначити перспективні напрямки використання дистанційних форм навчання. 2. З'ясувати проблеми дистанційної освіти (організаційні, правові, методичні).

Наскрізною формою організації професійного навчання є метод проєктів, який завершує вивчення кожного тематичного модуля і є формою підсумкового контролю на даному етапі.

Теоретичний модуль розподілено на три змістові модулі. Навчання з кожного змістового модуля включає слухання і конспектування лекцій, семінарські заняття, письмову контрольну роботу, систему самостійних завдань. Кожний змістовий модуль оцінюється у 33 бали. За ведення конспектів лекцій, за систематичність у роботі можна отримати від 0 до 5 балів. Оцінка роботи на семінарських заняттях залежно від активності коливається від 0 до 2 балів. Відповідь на контрольній роботі (модульний контроль) може оцінюватися: «не задовільно» – 0 балів, «задовільно» – 1 бал, «добре» – 2 бали, «відмінно» – 3 бали. Кожне завдання самостійної роботи оцінюється 3 балами, оскільки таких завдань у кожному модулі 6, то сума балів становить 18. Додатково майбутній викладач може отримати ще 5 балів за творче виконання розробок і захист електронного портфеля.

Під ефективністю функціонування запропонованої системи управління професійним навчанням майбутнього викладача розуміється комплексна характеристика результатів функціонування з урахуванням відповідності цих результатів соціальному замовленню суспільства, концепції та завданням розвитку вищої освіти, тобто інтегральна характеристика всіх її аспектів для досягнення головної мети професійного навчання – формування професійної компетентності майбутнього викладача спеціальних дисциплін.

Література:

1. Балл Г.А. Теория учебных задач: Психолого-педагогический аспект. – М.: Педагогика, 1990. – 184 с.
2. Биков В.Ю. Дистанційні технології навчання в сучасній освіті // Проблеми сучасного підручника: Зб. наук. праць. – К.: Педагогічна думка, 2004. – Вип. 5. – С. 15–22.
3. Дорошенко Ю.О. Основні напрямки розвитку інформаційно-навчального середовища // Збірник наукових праць. Педагогічні науки. Випуск 36. – Херсон: Вид-во ХДУ, 2004. – С. 18–28.
4. Монахов В.М. Педагогическое проектирование – современный инструментариий дидактических исследований // Школьные технологии. – 2001. – №5. – С. 75-99.

ПОЯСНЮВАЛЬНО-ІЛЮСТРАТИВНИЙ МЕТОД В КОНЦЕПЦІЇ ЕЛЕКТРОННОГО НАВЧАННЯ ІНФОРМАТИКИ

І.М. Лукаш

м. Чернігів, Чернігівський державний педагогічний університет
імені Т.Г. Шевченка
lukash@cg.ukrtel.net

На сучасному етапі розвитку освіти концепція електронного навчання покладається в основу національних планів навчальних технологій багатьох країн світу. Однією з причин такої значної уваги до застосування інформаційних технологій у системі освіти є критичне становище людства як біологічного виду в еволюції: проміжок часу інформаційного оновлення суспільства став меншим за час його біологічного оновлення (зміна поколінь). Вважається, що в розвинутих країнах людина в активний період свого життя повинна буде змінювати до п'яти галузей професійної діяльності, а 80% знань, що будуть необхідні сьогоднішнім випускникам, ще нікому не відомі [3]. Пошук ролі і місця інформаційних технологій у середовищі початкової, середньої, вищої та неперервної освіти потребує перегляду вже традиційних, укорінених методів навчання.

Пояснювально-ілюстративний або інформаційно-рецептивний метод виділяється І.Я. Лернером як репродуктивний метод навчання [5], у процесі застосування якого вчитель повідомляє матеріал, а учні його сприймають. У класифікації методів за дидактичними задачами навчання [2] наводиться метод повідомлення готового знання шляхом інформуючого викладення. Задачею вчителя при застосуванні цього методу є викладення змісту знань та дій шляхом усного повідомлення, а задачею учня є розуміння змісту пояснень, описів об'єктів та дій з ними. Якість засвоєння нового матеріалу залежить від попередньо засвоєних знань та пізнавальних умінь: виділення, порівняння характеристик об'єктів та дій; здійснення узагальнень, класифікації, підведення під поняття тощо. Зазначені дії виконуються репродуктивно, оскільки вони актуалізуються не учнем, а вчителем. Учні самому не доводиться встановлювати зв'язки між елементами знання. Вчитель повинен чітко та послідовно викладати матеріал, підкреслюючи всі необхідні для розуміння компоненти змісту.

Незважаючи на переваги творчо-пошукового, репродуктивного навчання не втратило свого значення та актуальності. Вважається, що до структури уроків вищого рівня, коли творче пізнання переважає над продуктивним, рекомендується переходити тільки через опрацювання системи попередніх уроків репродуктивного рівня [4], на яких учні повинні ознайомитись з теоретичними основами предмету та прийомами виконання інтелектуальних дій, необхідних для розв'язування задач. В [1] зазначається, що репродуктивний підхід до створення схеми орієнтовної основи дії, яка містить не тільки

зразки дії, але й усі вказівки щодо правильного виконання дії з новим матеріалом, призводить до швидкого та безпомилкового протікання навчання. Учень набуває певного уміння аналізувати матеріал з позиції майбутньої дії. Остання виявляє стійкість при зміні умов та, хоча й обмежено, але переноситься на нові завдання.

Об'єктивний процес інформатизації суспільства та концепція електронного навчання орієнтують освіту на розвиток дистанційної освіти, за допомогою якої існує можливість вирішити ряд проблем: навчання дітей, які не можуть навчатися традиційним шляхом; отримання додаткової освітньої інформації, яка швидко змінюється; збільшення кількості учасників навчання, які оперативно взаємодіють один з одним, тощо [6]. За одним із підходів реалізації дистанційного навчання учень розглядається як одержувач деякого інформаційного змісту для засвоєння, що передбачає застосування пояснювально-ілюстративного методу, наприклад, при розміщенні навчального матеріалу на Web-сторінці.

Не менш важливим для удосконалення процесу навчання є проектний метод, який належить до продуктивних методів навчання і здійснюється як в рамках дистанційного навчання у вигляді телекомунікаційних проєктів, так і у процесі традиційного навчання. У зв'язку з тим, що реалізація проєкту передбачає практичне поєднання різноманітного навчального матеріалу, однією з умов організації проєктної діяльності учнів є наявність певних описів як теоретичних знань, так і набору програмних засобів, якими необхідно буде оволодіти для розв'язування задачі-проєкту. Під створення таких описів можна розуміти пояснювально-ілюстративний метод навчання.

Підвищення ефективності навчання шляхом його індивідуалізації також вимагає від вчителя створення на основі пояснювально-ілюстративного методу певних посібників, зокрема, електронних, за допомогою яких можна здійснювати як випереджуюче навчання для сильніших учнів, так і навчання в зручному темпі для інших учнів, надаючи їм можливість ознайомлюватись з необхідним матеріалом достатню кількість разів і достатню кількість часу.

Не викликає сумнівів доцільність подання навчального матеріалу саме в електронному вигляді для реалізації пояснювально-ілюстративного методу навчання як при дистанційному, так і при традиційному навчанні, як при проблемному, так і при репродуктивному навчанні. Доцільно пам'ятати про те, що 80% інформації про оточуючий світ людина отримує через органи зору. Багатослівний звуковий або текстовий (на моніторі) коментар викликає швидко втому та ускладнює сприйняття динамічних процесів (анімацій) [7]. Тому, дотримуючись електронної концепції навчання, пояснювально-ілюстративний метод навчання доцільніше реалізовувати через відеопоказ об'єктів вивчення та орієнтовних основ дій над цими об'єктами.

Існує ряд програмних засобів для запису подій, що відбуваються на екрані монітора, у вигляді AVI- та SWF-файлів та програм для конвертації

файлів одних відеоформатів в інші. В залежності від формату відеоролика його можна розмістити на Web-сторінці, додати до презентації, створеної у Microsoft PowerPoint чи у Macromedia Flash тощо, реалізуючи тим самим пояснювально-ілюстративний метод в різних формах самостійного навчання. Готуючись до уроку інформатики, вчитель має можливість легко записати відеоролики розв'язування ключових задач, які потім зможе продемонструвати учням на уроці та надати для додаткового самостійного перегляду тим, хто не встиг “схопити зльоту” новий матеріал або був відсутній.

Програма CamStudio (eHelp Corporation) використовується для створення відео файлів, а також для запису звукового супроводу фільму. В комплекті з програмою створення відеофільму надається програма CamStudio Player для його перегляду. Панель інструментів програми CamStudio (рис. 1) містить такі засоби: 1 – включення запису фільму; 2 – пауза, призупинення запису; 3 – закінчення запису; 4 – подання форми вікна програми у зменшеному або повному вигляді; 5 – коментар до фільму, тобто задання тексту або малюнку, що залишається поверх вікна фільму.

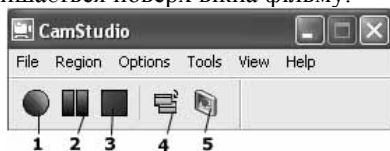


Рис. 1

Перш ніж здійснювати запис фільму, доцільно виконати певні налагодження програми, наприклад, через команду меню Region\ Full Screen встановити режим повноекранного запису. В пункті меню Options\ Video Options можна задати швидкість зміни кадрів. Засібу Options\ Keyboard Shortcuts дозволяє задати комбінації клавіш швидкого доступу. Маючи запущену і згорнуту програму CamStudio, можна здійснювати запис фільму за допомогою певних функціональних клавіш, не показуючи на екрані панель програми, а тим самим не записуючи її до фільму. Команда Options\ Cursor Options дозволяє задати режим наявності курсору в кадрах фільму та його вигляд. Для додання коментарів до фільму доцільно скористатися інструментом 5 на рис. 1. У вікні, що з'явиться, на закладці Shapes через команду New Shape контекстного меню створити новий коментар, задавши розмір прямокутної області, текстове повідомлення або завантаживши малюнок. Зроблені коментарі можна зберігати в файлі для подальшого використання.

Програма HyperCam (Hyperionics Technology) також призначена для створення відео .avi файлів. Інтерфейс програми (рис. 2) складається з вікна, що має кілька закладок.

Закладка Screen Area використовується для задання області екрана, у якій буде здійснюватися запис. На закладці Hot Keys задаються комбінації клавіш для здійснення ключових команд запису. Закладка AVI File містить інформацію (яку при необхідності можна змінювати) про робочу папку, ку-

ди буде зберігатися записаний фільм; про швидкість запису і відновлення кадрів; про включення режиму запису звука. На закладці Sound можна задавати певні характеристики запису звука. Закладка Other Options містить опції запису курсору та коментарів до фільму (Screen Notes). Цікавим є механізм Starburst to the movie when mouse is clicked, який дозволяє показувати графічну фігуру, наприклад, червоного кольору, коли була натиснута ліва кнопка миші, і синього – коли була натиснута права кнопка миші, що надає значні зручності для демонстрації орієнтовної основи дії, в якій використовуються як ліва, так і права кнопки миші.

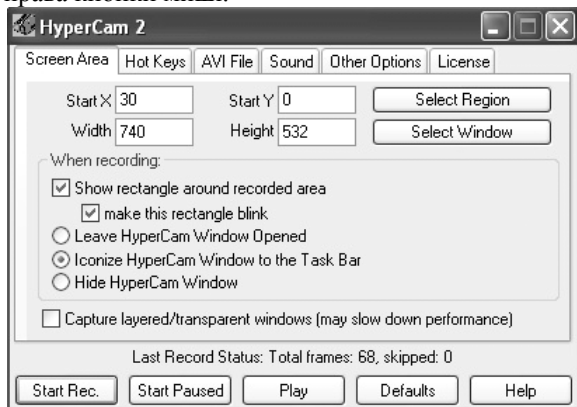


Рис. 2.

Програма Flash Cam Corporate Edition2 (рис. 3, 4) призначена для створення Flash ролика у форматі .swf. Програма як Flash камера в певні моменти часу здійснює за вимогою користувача (натиснення клавіші миші) захват зображення на екрані, формуючи набір ключових фреймів (кадрів), які можна редагувати та домальовувати як звичайні графічні зображення.

Програма запам'ятовує координати розташування курсору миші, сама прораховує і будує траєкторію його руху до нової позиції, зафіксованої у наступному фреймі. Якщо дворазово натиснути лівою кнопкою миші на будь-якому із фреймів, то завантажиться вікно редагування ролика (рис. 4), у якому можна візуально побачити ту лінію, яку програма визначила для пересування курсору.

Панель інструментів редактора також має ряд властивостей для налагодження ролика, зокрема, інструмент, позначений на рис. 4 зірочкою, надає можливість задання швидкості зміни фреймів.

Для перегляду роликів в програмі передбачено завантаження їх або до програми Flash Player, або до Internet Explorer у вигляді Web-сторінки. Вибір способу перегляду ролика здійснюється через пункт меню Options. Ця програма може бути використана для створення навчальних Flash презентацій. Для конвертації Flash роликів до формату .avi існує програма Magic

Swf2Avi.

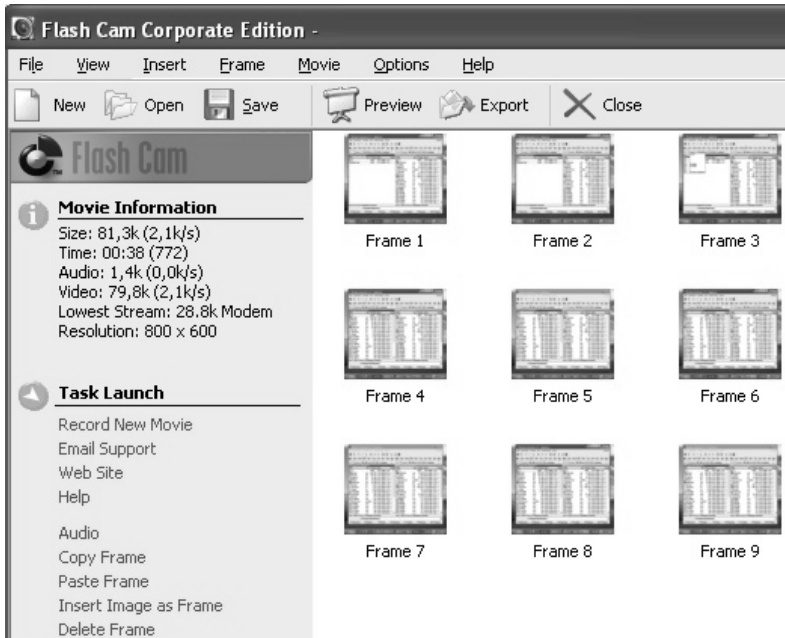


Рис. 3.

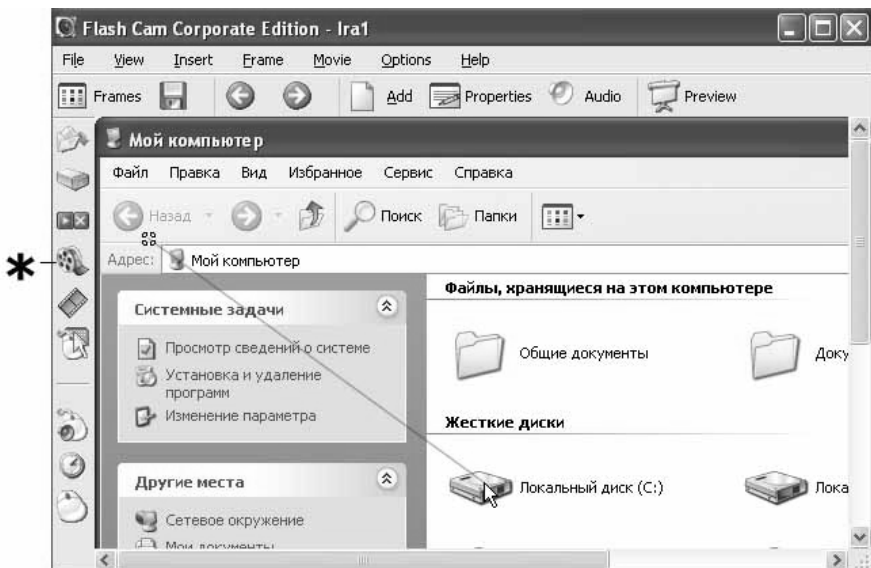


Рис. 4.

При використанні розглянутих програмних засобів вчитель має нескладну можливість вдосконалення реалізації пояснювально-ілюстративного методу навчання згідно із концепцією електронного навчання. Доцільним є застосування зазначених програм для виконання лабораторних робіт із реалізації пояснювально-ілюстративного методу навчання в рамках курсу методики навчання інформатики для педагогічних навчальних закладів.

Література:

1. Гальперин П.Я., Талызина Н.Ф. Современное состояние теории поэтапного формирования умственных действий. – Вестник МГУ. Серия 14. Психология. – М., 1979. – С. 54-63.
2. Ильясов И.И., Галатенко Н.А. Проектирование курса обучения по учебной дисциплине. – М.: Логос, 1992. – 207 с.
3. Колин К.К. Стратегические приоритеты образования // Открытое образование. – 2001. – №5. – С.6-12.
4. Коротяев Б.И. Методы учебно-познавательной деятельности учащихся: 13.730. Автор...докт. пед. наук. – М., 1971. – 37 с.
5. Лернер И.Я. Дидактические основы процесса обучения. – М.: Педагогика, 1981.
6. Морзе Н.В. Методика навчання інформатики: У 4 ч. – К.: Навчальна книга, 2003. – Ч.1: Загальна методика навчання інформатики. – 254 с.
7. Стародубцев В.А. Компьютерные и мультимедийные технологии в естественнонаучном образовании. – Томск: Дельтаплан, 2002. – 224 с.

ЗДОБУТКИ ТА ПРОБЛЕМИ ВПРОВАДЖЕННЯ ПРОГРАМИ «INTEL® НАВЧАННЯ ДЛЯ МАЙБУТНЬОГО»

С.В. Кондратенко

м. Кривий Ріг, Криворізький державний педагогічний університет
ksv_ipm@mail.ru

Відповідно до наказу Міністерства освіти і науки України від 22.04.2005 р. №248, у жовтні 2005 року Криворізький державний педагогічний університет приєднався до експерименту по впровадженню Програми «Intel® Навчання для майбутнього» у практику роботи вищої школи з метою підготовки педагогічних кадрів на замовлення сучасного інформаційного суспільства. Під керівництвом досвідчених тренерів Intel-центру м. Київ – Лариси Шевчук та Алли Забарної – 12 викладачів університету пройшли інтенсивний тренінг за програмою та отримали сертифікати.

Не уповільнюючи темпів роботи, набутих під час тренінгу, викладачами були намічені перші кроки ознайомлення студентів з інноваційними підходами у викладенні шкільних предметів з використанням інформаційно-комунікаційних технологій (ІКТ). Викладачами-методистами з української мови та літератури (З.П. Бакум), німецької мови (І.В. Мельничуком) та географії (Н.Б. Жигуною) миттєво були внесені зміни у навчальні програми з курсів методики навчання конкретних предметів для магістрів 5-го і 6-го років навчання.

Метою нашої статті є представлення результатів впровадження Програми на фізико-математичному факультеті тренерами-методистами кафедри інформатики та прикладної математики.

Н.В. Морзе, науковий керівник, старший тренер та внутрішній експерт Програми, пропонує її впровадження у навчальний процес вищих педагогічних навчальних закладів за однією з таких моделей [2]:

– педпрактика – тренінг (48 годин аудиторних + 20 годин самостійної роботи);

– базовий курс «Нові інформаційні технології» – тренінг (54 години аудиторних);

– факультативний курс «Інноваційні педагогічні технології в освіті» – тренінг;

– обчислювальна практика на фізико-математичному факультеті (тренінгова модель);

– курс «Методика навчання (конкретного навчального предмета)» (48 аудиторних годин);

– базовий класичний курс «Нові інформаційні технології»;

– факультативний курс «Інноваційні педагогічні технології в освіті».

Було заплановано навчання студентів-фізиків 4 курсу під час обчислювального практикуму. Вибір фізиків слухачами курсу не був випадковим,

адже через кілька місяців на них чекала педагогічна практика – реальна можливість реалізації авторських навчальних проєктів.

На проведення обчислювального практикуму за навчальним планом виділяється 72 години аудиторного часу на 2 навчальні тижні (включаючи суботу) – 36 лабораторних занять. Таким чином, на засвоєння кожного з дванадцяти модулів було виділено три лабораторні заняття.

Для проведення тренінгу була опрацьована та адаптована програма курсу «Інноваційні педагогічні технології в освіті» [2].

Планувалося залучити до тренінгу всіх студентів потоку у кількості 45 осіб. Дотримуючись рекомендацій ефективного впровадження Програми, всі вони були запрошені на захист Портфолію викладачів. Але через обмеженість комп'ютерних аудиторій, придатних для навчання за Програмою, і обмеженою кількістю підготовлених тренерів-методистів Програми, стало можливим навчання лише їх частини – 19 студентів. Проте, учасниками тренінгу стали ті студенти, які виявили бажання перевірити власні творчі здібності у роботі над розробкою навчального проєкту. До речі, альтернативною тренінгу з інноваційної методики було вивчення алгоритмів комп'ютерної графіки.

Студентам пропонувалося підготувати навчальний проєкт або з фізики, або з астрономії, або з інформатики.

Статистика остаточного вибору тематики проєктів представлена у таблиці:

Предметна галузь навчального проєкту	Кількість проєктів
Фізика	10
Астрономія	7
Інформатика	1
Математика	1
Разом:	19

Перш, ніж здійснити вибір теми проєкту, студенти отримали завдання ознайомитися з діючими шкільними програмами, встановити, вивчення яких тем припадатиме на час проходження практики.

Поставлена задача не всім виявилася під силу, тому значна кількість навчальних проєктів відповідали тематиці курсової роботи з астрономії, або стосувалися тих розділів шкільного курсу фізики, які на той момент вивчалися за програмами вузівських дисциплін з фізики.

Дотримуючись вимог, висунутих до Портфолію, кожен студент намагався представити відмінний проєкт. Проте обмеженість у часі, відсутність педагогічного досвіду і особливо творчого натхнення як раз і стали об'єктивними критеріями для оцінювання результатів, які виявились не однаково якісними.

На нашу думку, до найбільш вдалих робіт студентів слід віднести навчальні проєкти саме з фізики на теми «Зі швидкістю вітру» (5-7 кл.), «Екс-

тремальний серфінг – верхи на радіохвилі» (10-11 кл.), «Кристалоселекція» (10-11 кл.), «Мертва петля» (8-9 кл.) та «Зіграйте з нами в схованки» (8-9 кл.).

На жаль, Програмою не передбачено ознайомлення тренерів з недоліками проектів одразу після перевірки документів, надісланих до Intel-центру (м. Київ). Тому, наскільки висунуті припущення щодо «вдалості» навчальних проектів відповідають дійсності – судити складно, адже вирішальною залишається суб'єктивна оцінка місцевих тренерів, а не оцінка спеціалістів центру.

З упевненістю можемо зазначити, що обрана тренінгова модель реалізації програми у ВНЗ – «Обчислювальна практика на фізико-математичному факультеті» – цілком виправдала себе. Але, підводячи підсумки проведеного практикуму-тренінгу за Програмою, можна відзначити як позитивні моменти, так і негативні. І першим позитивом є стовідсоткова успішність – всі слухачі тренінгу підготували і презентували Портфолію з 14 документів, отримали сертифікати і залік.

До негативних слід віднести такі моменти:

– через невизначеність бази проходження педагогічної практики – школи і класу – студенти не можуть підготувати проект для його безпосереднього впровадження під час проходження практики;

– розробка якісних проектів, готових до реалізації в школі під час педагогічної практики на 4 курсі, вимагає участі викладачів-методистів при його розробці на тренінгу студентів.

З метою удосконалення підготовлених навчальних проектів, студентам, які пройшли курс, ми пропонуємо доопрацьовувати проект шляхом написання курсової роботи з методики викладання відповідної дисципліни.

За станом на 18 січня 2006 року з червня 2005 р. за програмою «Intel® Навчання для майбутнього» навчено 309 викладачів та 434 студенти ВНЗ I-IV рівнів акредитації [1].

Література:

1. Михайліченко М.В., Морзе Н.В. МОН та Майкрософт: інформаційна співпраця продовжується. – Матеріали сайта: <http://www.mon.gov.ua>
2. Морзе Н.В. Особливості навчання майбутніх вчителів ефективному використанню інформаційно-комунікаційних технологій у навчальному процесі // Збірник наукових праць Уманського державного педагогічного університету ім. П. Тичини. – К.: Міленіум, 2005.
3. Морзе Н.В. Програма курсу «Інноваційні педагогічні технології в освіті» для вищих навчальних закладів системи педагогічної освіти.
4. Нанаєва Т. Програма «Intel® Навчання для майбутнього» // Комп'ютер у школі та сім'ї. – 2004. – №2. – С. 15.

«ХТО ВОЛОДІЄ ІНФОРМАЦІЄЮ, ТОЙ ВОЛОДІЄ СВІТОМ»

О.М. Туравініна

м. Кривий Ріг, Криворізький державний педагогічний університет

Опинившись серед учасників тренінгу «Intel. Навчання для майбутнього» у Криворізькому державному педагогічному університеті, моїм очікуванням про пізнання чогось нового в методичній стороні своєї діяльності не було меж. І ці очікування справдилися. Не знаю чому, але в питанні підбору теми до проекту було складно. Може тому, що сам предмет «Шкільний курс інформатики» направлений на ознайомлення та впровадження в життя нових інформаційних технологій, а тренінг проводиться саме з цією метою. Врешті-решт темою проекту, орієнтованого на учнів 11 класу, було обрано «Інформація. Види інформації та її властивості». Відповідно до обраної теми ключовим питанням проекту став вислів «Хто володіє інформацією, той володіє світом».

Весь проект представлений у вигляді портфоліо – впорядкованої електронної збірки матеріалів, що являє собою сукупність методичних та дидактичних матеріалів, допоміжних документів, учнівських робіт, прикладів, розроблених вчителем для учнів тощо. Зауважу, що створення прикладів від імені учнів виявилось непростим завданням для багатьох учасників тренінгу. Кожний відчув, як складно передбачити хід дитячих думок щодо дослідження.

За планом проекту для наочного ознайомлення з темою дослідження та завданням вчителем створена презентація, в якій перед учнями схематично ставиться ціла низка питань (рис. 1).

Дайте відповіді на такі питання:

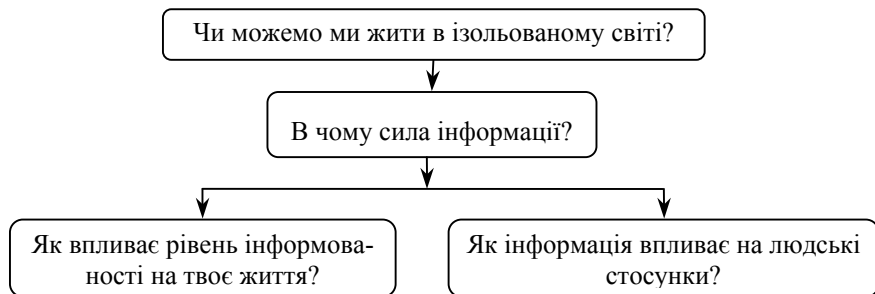


Рис. 1.

Учням пропонується експериментально перевірити сутність процесу отримання інформації, дослідити її властивості, довести ефективність використання різних засобів обробки інформації, дати характеристику інформаційним процесам. Для систематизації всіх опрацьованих джерел (друкова-

них та електронних) пропонується створити базу даних, для чого необхідно виконати наступні дії:

- обгрунтувати вибір СУБД;
- спроєктувати структуру БД;
- проаналізувати типи даних;
- побудувати запити, необхідні для автоматизації пошуку.

За задумкою викладача, результати досліджень (на будь-якому етапі) учні можуть представити у вигляді презентації, публікації або Web-сайту.

Наприкінці презентації вчитель знайомить учнів з критеріями оцінювання кожного учнівського прикладу, надає перелік літератури та електронних адрес. За бажанням, якщо дослідна робота вийде далеко за межі рекомендованих джерел, перелік може бути значно розширеним.

Проект відповідає державним освітнім стандартам та навчальним програмам, виправдовує навчальні цілі та очікувані результати навчання. Це підтверджується змістом плану проекту, що створюється вчителем на початку всієї роботи, і який, разом з планом реалізації проекту є обов'язковим документом методичного забезпечення.

Важливе значення серед методичного забезпечення відводиться створенню засобами Microsoft Excel таблиці оцінювання учасників проекту (рис. 2), яка демонструє яким чином формується загальна оцінка.

Оцінювання рівня досягнень учнів з використання комп'ютерних технологій					
Прізвище та ім'я учня	Бали за презентацію	Бали за публікацію	Бали за Web-сайт	Загальна сума балів	Оцінка учня
Маханько Оксана	12	12	10	34	11
Шостак Ірина	10	11	10	31	10
Лузін Костя	11	10	11	32	11
Дмитренко Євген	10	10	10	30	10
Коломієць Микола	6	3	8	17	6
<i>Максимум</i>	12	12	12	36	

Рис. 2.

На основі таблиці з метою наочного подання результатів оцінювання створюється діаграма (рис. 3).

Проект ще не завершено. Попереду багато роботи з його вдосконалення. Можливо, дослідницька робота поставить нові запитання, відповіді на які учням буде цікаво знайти самостійно.

Неможливо не погодитися з розробниками цього тренінгу: «Створення Портфоліо навчального проекту можливе для Всіх вчителів Будь-Якого навчального предмета». І це перевірено на практиці.

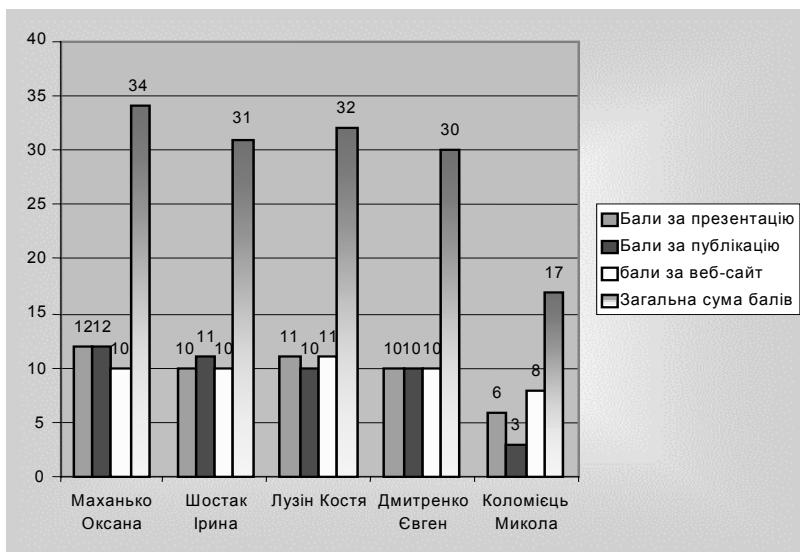


Рис. 3.

Література:

1. Intel® навчання для майбутнього. – К.: Видавнича група BHV, 2004. – 416 с.
2. Глинський Я.М. Практикум з інформатики: Навч. посіб. – 8-е оновл. вид. – Львів: Деол, СПД Глинський, 2005. – 296 с.
3. Глушаков С.В., Сурядний А.С., Хачиров Т.С. Домашний ПК. – Харьков: Фолио, 2004.
4. Морзе Н.В. Методика навчання інформатики: Методичний посібник: у 4 частинах / За ред. М.І. Жалдака. – К.: Навчальна книга, 2003. – Ч. 3: Методика навчання основних послуг глобальної мережі Internet. – 196 с.
5. Морзе Н.В., Дементієвська Н.П. Телекомунікаційні проекти: стан та перспективи // Комп'ютер у школі та сім'ї. – 1999. – №4. – С. 21-24.
6. Нанаєва Т. Програма "Intel. Навчання для майбутнього" // Комп'ютер у школі та сім'ї. – 2004. – №2. – С. 15-16.

ІНТЕРАКТИВНІ ТЕХНОЛОГІЇ НАВЧАННЯ НА УРОКАХ З ОСНОВ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ

А.П. Забарна

м. Сміла, Смілянський природничо-математичний ліцей
liceum@ck.ukrtel.net, zam@smela.com.ua

Сучасні зміни в освіті зробили актуальним звернення вчителів і вчених-педагогів до проблем застосування активних та інтерактивних методів навчання. Інтерактивні технології навчання створюють необхідні умови як для становлення і розвитку компетентностей учнів, так і для розвитку й виховання особистості активних громадян із відповідною системою цінностей. Інтерактивність у навчанні можна пояснити як здатність до взаємодії, навчання у режимі бесіди, діалогу, дії. Отже, у дослівному розумінні інтерактивним може бути названий такий метод навчання, при якому той, хто навчається, є активним учасником навчального процесу, тобто здійснює щось: говорить, управляє, моделює, пише, малює тощо. Він виступає не лише слухачем, спостерігачем, а бере активну участь у тому, що відбувається [1].

Застосуванню інтерактивних технологій в навчально-виховному процесі присвячено значну кількість публікацій, наприклад [1; 2; 4; 5; 6; 7; 8; 9]. В деяких з них [7; 9] досить детально описується організація та проведення інтерактивного уроку. Проте, як правило, публікації стосуються навчання предметів гуманітарного циклу або містять загальний опис даної технології.

В даній статті описується структура інтерактивного уроку та наводиться приклад уроку з розробленого автором циклу інтерактивних уроків з програмування (всього 15 уроків).

Сьогодні визначені такі основні ознаки інтерактивної моделі навчання [7]:

- це навчання змінює взаємодію педагога і учнів: активність педагога поступає місцем активності учнів, а завдання учителя полягає у створенні умов для їх ініціативи;

- це навчання через досвід, яке повинне супроводжуватися набуттям учасниками конкретного досвіду, його осмислення, рефлексію, застосування на практиці;

- це діалогове навчання, під час якого здійснюється активна взаємодія вчителя й учнів. Інтеракція виключає домінування будь-якого учасника навчання, будь-якої думки, точки зору над іншими.

Сутність інтерактивного навчання полягає у тому, що це – співнавчання, взаємонавчання (колективне, групове, навчання у співпраці), в якому учень і вчитель є рівноправними, рівнозначними суб'єктами освітнього процесу та розуміють, що саме вони роблять, аналізують та усвідомлюють, що вони знають, вміють і яку діяльність здійснюють [1].

Як відомо, основною формою організації навчальної діяльності є со-

годні класно-урочна система. Проте, застосування інтерактивних технологій висуває певні вимоги до структури уроків. Як правило, структура таких занять складається з п'яти етапів [9]:

- 1) *мотивація учнів до навчання* (5% часу занять);
- 2) *оголошення, представлення теми та очікуваних навчальних результатів* (5% часу занять);
- 3) *надання необхідних відомостей* (10-15% часу занять);
- 4) *виконання інтерактивної вправи* – центральна частина заняття (50-60% часу на уроці);
- 5) *підбиття підсумків, оцінювання результатів роботи учнів* (до 20% часу на уроці).

Етап мотивації має на меті сфокусувати увагу учнів на проблемі й викликати інтерес до обговорюваної теми. Учень має бути налаштованим на ефективний процес пізнання, бути в ньому зацікавленим, усвідомлювати, що він зараз робитиме і навіщо. З цією метою можуть бути використані прийоми, що створюють проблемні ситуації, викликають подив, інтерес до змісту знань. Це може бути і коротка розповідь учителя, і бесіда, і демонстрація наочностей і нескладна інтерактивна технологія (“мозковий штурм”, “мікрофон”).

Другий етап – *оголошення теми та очікуваних результатів*. Мета цього етапу – забезпечити розуміння учнями змісту їхньої діяльності, тобто того, чого вони повинні досягти на уроці і чого від них чекає вчитель. Обов'язкове формулювання очікуваних результатів інтерактивного уроку для сприяння успішності навчання учнів повинне відповідати таким вимогам:

- висвітлювати результати **діяльності на уроці учнів, а не вчителя**, і повинно формулюватися, починаючи зі слів: “Після цього уроку учні зможуть...”;
- чітко відображати рівень навчальних досягнень учнів, який очікується після уроку. Тому в очікуваних результатах уроку повинні бути названі: обсяг і рівень засвоєння матеріалу учнями, що буде забезпечений на уроці; обсяг і рівень навичок і вмій, який буде досягнуто після уроку; розвиток (формування) емоційно-ціннісної сфери учня, яка забезпечує формування переконань, характеру, впливу на поведінку тощо;
- щоб було зрозуміло, як можна виміряти такі результати, коли вони будуть досягнуті;
- бути короткими, зрозумілими і для учнів, і для самого вчителя, і для батьків учнів і для інших вчителів.

Отже, очікувані результати мають бути сформульовані за допомогою відповідних дієслів, наприклад для набуття знань: пояснювати, визначати, характеризувати, порівнювати, відрізняти тощо; для формування уміння і навички: дискутувати, аргументувати думку, дати власну оцінку, проаналізувати тощо; для вироблення ставлення: сформулювати та висловлювати влас-

не ставлення до ..., пояснювати своє ставлення до

Наступний етап – *надання необхідних відомостей*. Мета цього етапу – надати учням достатній обсяг відомостей для того, щоб на їх основі вони могли виконувати практичні завдання. Це може бути міні-лекція, читання роздаткового матеріалу, знайомство з необхідним відомостями за допомогою технічних засобів навчання або наочностей.

Виконання інтерактивних(ної) вправ(и) – найважливіший етап заняття. Його метою є засвоєння навчального матеріалу, досягнення очікуваних результатів уроку. Обов'язковою є така послідовність і регламент організації виконання інтерактивної вправи:

- 1) інструктування (повідомлення про мету вправи, правила, послідовність дій, кількість часу);
- 2) об'єднання в групи і (або) розподіл ролей;
- 3) виконання завдання, при якому вчитель виступає як організатор, помічник, ведучий дискусії, надає учасникам максимум можливостей для самостійної роботи і навчання у співпраці один з одним;
- 4) презентація результатів виконання вправи;
- 5) усвідомлення отриманих результатів, що досягається шляхом їх спеціального колективного обговорення.

Існує досить велика кількість різновидів інтерактивних вправ. В літературі [7; 8; 9] найчастіше зустрічається опис таких: “два-чотири-всі разом”, “акваріум”, “уявний мікрофон”, “мозковий штурм”, “Броунівський рух”, “ажурна пилка”, “метод ПРЕС”, “дискусія”, “дебати”, “рольові та ситуаційні ігри”, “дерево рішень”, “посиденьки” та багато інших. Проаналізувавши вище названі та інші інтерактивні вправи ми дійшли висновку, що на уроках інформатики найбільш придатними до застосування інтерактивними вправами можуть бути “ажурна пилка”, “мозковий штурм”, “посиденьки”, “Броунівський рух”, “рольова гра”, “робота в малих групах”.

П'ятий етап – *підбиття підсумків, рефлексія уроку*. Саме тут прояснюється зміст проробленого, з'ясовується ступінь виконання очікуваних результатів уроку, встановлюється зв'язок між тим, що вже відомо, і тим, що знадобиться учням у майбутньому. Зазвичай на підсумковому етапі учням пропонується відповісти на питання, які допоможуть відновити послідовність виконаних дій, проводиться вивчення відтвореної послідовності дій з точки зору її ефективності, продуктивності, відповідності поставленим завданням тощо. Очевидно, що головним призначенням підведення підсумків є повернення до очікуваних результатів навчання і можливість переконатися, що учні їх досягли.

Наведемо приклад інтерактивного уроку з інформатики.

Тема уроку: Створення циклічних програм на знаходження суми, добутку та визначення кількості елементів з певними властивостями.

1. Мотивація навчальної діяльності учнів.

Вчитель пригадує разом з учнями, що на попередніх уроках вони по-

знайомились з трьома операторами циклу, з форматами їх запису, алгоритмами та особливостями їх роботи. Вчитель пропонує учням скласти алгоритм знаходження середньої температури за місяць та підрахунку кількості теплих ($t^{\circ}\text{C} > 0$) та холодних ($t^{\circ}\text{C} < 0$) днів. Учні приходять до висновку, що основним завданням в цьому алгоритмі є знаходження суми всіх середньодобових температур та організація підрахунку кількості відповідних днів. Вчитель повідомляє, що при розв'язуванні різноманітних обчислювальних задач часто виникає потреба підраховувати кількість якихось подій або якихось даних, що задовольняють певним умовам, накопичувати суму або добуток. Для розв'язання таких задач також використовуються оператори циклу.

II. Оголошення теми та очікуваних навчальних результатів. Після цього уроку учні зможуть:

2. пояснювати алгоритм знаходження суми та добутку скінченої кількості чисел, знаходження кількості елементів з певними властивостями;

3. наводити особливості знаходження суми та добутку скінченої кількості чисел, підрахунку кількості елементів з певними властивостями;

4. складати алгоритми на знаходження суми, добутку скінченої кількості чисел, знаходження кількості елементів з певними властивостями та реалізовувати ці алгоритми на одній з мов програмування.

5. набути навичок роботи в групі.

III. Надання необхідних відомостей.

Вчитель пояснює, що для обчислення суми, добутку або кількості під ці величини виділяється спеціальна змінна відповідного типу. Далі, поєднуючи оператор циклу з оператором присвоєння, можна організувати накопичення суми, добутку або кількості. В кожному з цих випадків накопичення відбувається за такою схемою: спеціальна змінна, де буде відбуватися накопичення – це один кошик, в який весь час докладають дані.

IV. Інтерактивні вправи.

1. Вчитель пропонує учням виконати інтерактивну вправу за методом “Ажурної пилки”. Для цього вчитель пропонує учням об'єднатися в групи. А саме: учні отримують картку одного з кольорів (синього, червоного або жовтого) з номером на ній (від 1 до 5). В результаті формуються три “домашні” групи по п'ять учнів у кожній. Групи отримують картки із завданнями. Кожній “домашній” групі пропонується познайомитись із змістом карток, розглянути наведені блок-схему та програму, відповісти на питання, обговорити висновки, наведені в картах, записати в зошит програму та висновки.

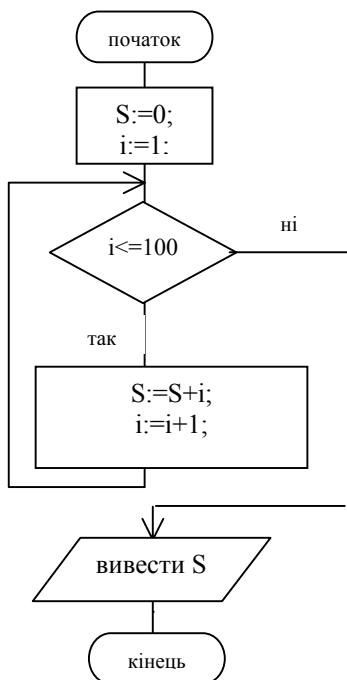
Зміст карток:

С и н і

Обчислення суми.

Завдання. Скласти програму для обчислення суми цілих чисел від 1 до 100.

Для накопичення суми введемо спеціальну змінну, наприклад S. Під



час обчислення суми початкове значення змінної S , де накопичується сума, має **дорівнювати нулеві**. Для цього використовують команду присвоєння $S:=0$. Так як треба перебрати натуральні числа від 1 до 100 з кроком 1, то доцільно скористатися оператором циклу з параметром.

Познайомтесь з блок-схемою алгоритму накопичення суми та розгляньте програму, яка реалізує даний алгоритм (файл prog1.pas):

```

var s, i:integer;
begin
  s:=0;
  for i:=1 to 100 do s:=s+i;
  writeln('s=',s);
end.

```

Змінна i отримуватиме значення: 1, 2, 3, ..., 100, а змінна S – такі: 0, 1, 3, 6, 10, 15, ..., 5050.

Як треба змінити програму, щоб вона обчислювала суму чисел від 200 до 300?

Висновки: для накопичення суми треба:

- ввести змінну для обчислення суми;
- надати цій змінній початкового значення 0;
- організувати цикл, в тілі якого записати оператор $\text{сума}:=\text{сума}+\text{доданок}$;

Ч е р в о н і

Обчислення добутку.

Задача. Обчислити добуток перших n яти елементів послідовності, значення яких знаходяться за формулою $a_k=2+|\sin 3k|$, $k=1, 2, \dots, 5$.

Для накопичення добутку введемо спеціальну змінну, наприклад D . Під час обчислення добутку початкове значення змінної, де накопичується добуток, має дорівнювати одиниці. Отже, $D:=1$. Так як треба перебрати натуральні числа (індекси елементів) від 1 до 5 з кроком 1, то доцільно скористатися оператором циклу з параметром.

Познайомтесь з блок-схемою алгоритму накопичення добутку та розгляньте програму, яка реалізує даний алгоритм (файл prog2.pas):

```

var d, k:integer;
begin
  d:=1;
  for k:=1 to 5 do d:=d*(2+abs(sin(3*k)));
  writeln('d=',d:7:1);
end.

```

Змінна k отримуватиме значення: 1, 2, 3, ..., 5.

Як треба змінити програму, щоб вона обчислювала добуток перших десяти елементів?

Висновки: для накопичення добутку треба:

- ввести змінну для обчислення добутку;
- надати цій змінній початкового значення 1;
- організувати цикл, в тілі якого записати оператор добуток:=добуток*множник;

Ж о в т і

Обчислення кількості елементів з певними властивостями.

Задача. Скільки елементів послідовності $a_i = 1 - \cos i$, де $i = 1, 2, \dots, 20$, задовольняють умову $0 < a_i < 1$?

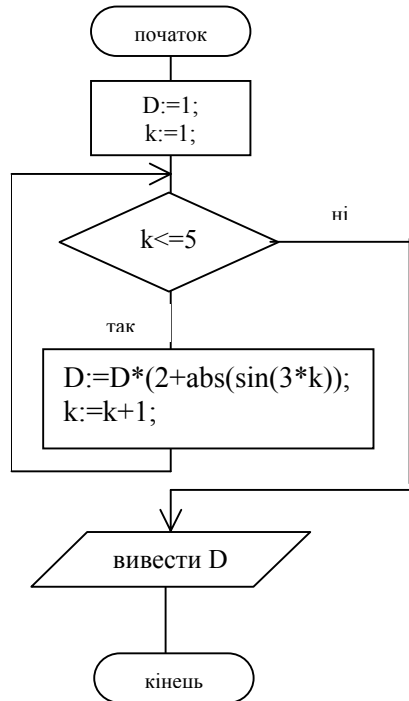
Для обчислення кількості елементів, що задовольняють вказану умову, введемо спеціальну змінну, наприклад k . Під час обчислення кількості початкове значення змінної, де буде накопичуватись кількість, має дорівнювати нулеві ($k := 0$). Так як треба перебрати натуральні числа (індекси елементів) від 1 до 20 з кроком 1, то доцільно скористатися оператором циклу з параметром, в тілі якого має бути команда $k := k + 1$;

Познайомтесь з блок-схемою алгоритму обчислення кількості та розгляньте програму, яка реалізує даний алгоритм (файл prog3.pas):

```
var a:real;
    i,k:integer;
begin
k:=0;
for i:=1 to 20 do begin
a:=1-cos(i);
if (a>0)and(a<1) then k:=k+1;
end;
writeln('k=',k);
end.
```

Як треба змінити програму, щоб за її допомогою можна було б знайти кількість елементів, більших за 1?

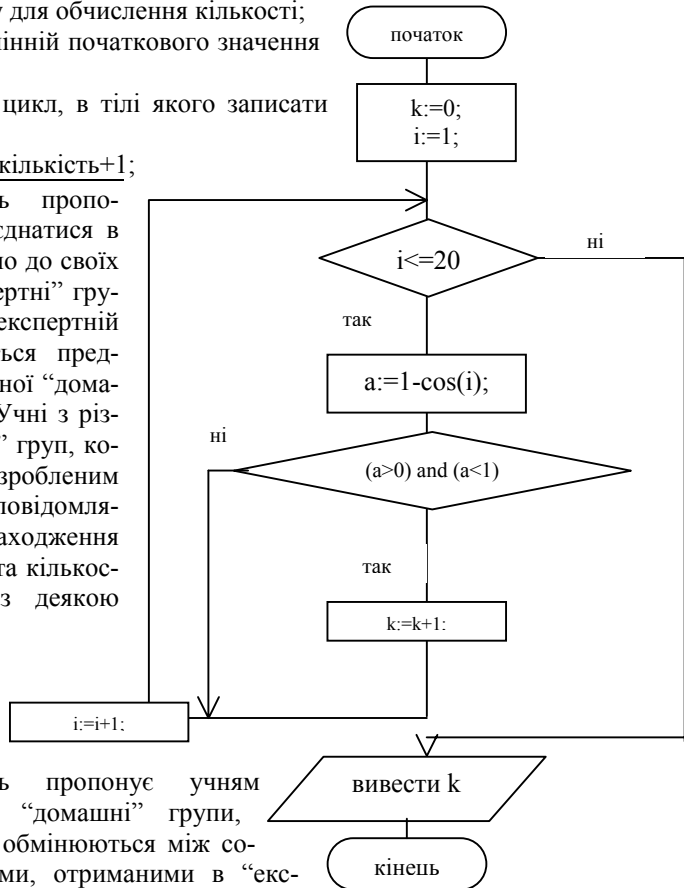
Висновки: для накопичення кількості елементів з певними властивостями треба:



- ввести змінну для обчислення кількості;
- надати цій змінній початкового значення 0;
- організувати цикл, в тілі якого записати оператор

$\text{кількість} := \text{кількість} + 1;$

2. Вчитель пропонує учням об'єднатися в групи відповідно до своїх номерів ("експертні" групи). У кожній експертній групі опиняються представники з кожної "домашньої" групи. Учні з різних "домашніх" груп, користуючись зробленим конспектом, повідомляють правила знаходження суми, добутку та кількості елементів з деякою властивістю.



3. Вчитель пропонує учням об'єднатися в "домашні" групи, учасники яких обмінюються між собою відомостями, отриманими в "експертних" групах, порівнюються правила обчислення суми, добутку та кількості.

4. "Домашні" групи отримують завдання на картках. Зміст карток:

Скласти алгоритми розв'язання наступних задач. Записати ці алгоритми мовою програмування. Порівняти отримані програми. Перевірити, чи відповідають ці програми висновкам, наведеним у картках з попередньої вправи.

Задача 1. Підрахувати кількість теплих днів за один місяць, за умови, що середньодобова температура вводиться з клавіатури.

Задача 2. Знайти середню температуру за місяць, за умови, що середньодобова температура вводиться з клавіатури.

Задача 3. Знайти середнє геометричне n'яти чисел, введених з клавіатури.

Клас об'єднується в загальне коло для підведення підсумків вправи. Представники кожної "домашньої" групи повідомляють свій варіант однієї з

програм до задач 1-3.

V. Практична робота на ПЕОМ.

Вчитель пропонує учням виконати програми до задач 1-3 на комп'ютері.

VI. Підбиття підсумків, оцінювання результатів уроку.

Учням пропонується відповісти на запитання:

1. Які оператори мови програмування Паскаль використовуються для знаходження суми, добутку та кількості елементів з певною властивістю?
2. Яким повинно бути початкове значення змінної, в якій передбачається накопичувати суму?
3. Яким повинно бути початкове значення змінної, в якій передбачається накопичувати добуток?
4. Яким повинно бути початкове значення змінної, в якій передбачається накопичувати кількість?
5. Яка загальна схема знаходження суми (добутку, кількості)?
6. Які помилки можуть виникнути при знаходженні суми, добутку та кількості?
7. Як ви гадаєте, чи виконали ми ті завдання, що поставили на початку уроку?
8. Які труднощі виникали у вас протягом уроку?
9. Що ви порадите однокласникам для подолання труднощів, що виникали?

VII. Повідомлення домашнього завдання.

§20, вправи 1-8 підручника [2].

Накопичений в Україні та за кордоном досвід переконливо засвідчує, що інтерактивні методи сприяють інтенсифікації навчального процесу та підвищують його ефективність. Вони надають можливість учням:

- аналізувати навчальний матеріал, творчо підходити до його засвоєння й тому зробити цей процес більш доступним;
- навчитись формулювати власну думку, правильно її висловлювати, доводити власну точку зору, аргументувати й дискутувати;
- навчитись слухати іншу людину, поважати альтернативну думку;
- моделювати різні соціальні ситуації та переживати їх, тим самим збагачуючи їх власний соціальний досвід;
- вчитись будувати конструктивні відносини в групі, визначати своє місце в ній, уникати конфліктів, розв'язувати їх, шукати компроміси, прагнути діалогу;
- знаходити шляхи для спільного розв'язання проблем;
- розвивати навички проектної діяльності, самостійної роботи, виконання творчих робіт.

Автор також з власного досвіду стверджує, що застосування інтерактивних технологій навчання дійсно сприяє підвищенню ефективності уроку та зацікавленості учнів у вивченні розділу “Основи алгоритмізації та програмування”.

Література:

1. Антологія адаптованого досвіду. Сайт «Вісника програм шкільних обмінів» <http://www.visnyk.iatp.org.ua>. 30/01/2006
2. Баханов К.О. Інноваційні системи, технології та моделі навчання історії в школі: Монографія. – Запоріжжя: Просвіта, 2000. – 160с.
3. Глинський Я.М. Інформатика: 8–11 класи. Навчальний посібник для загальноосвітніх навчальних закладів: У 2-х кн.. – Кн. 1. Алгоритмізація і програмування. 3-тє вид. – Львів: Деол, СПД Глинський, 2003. – 200 с.
4. Инновационные методы обучения в гражданском образовании. – Минск: Медисонт, 1999. – 168 с.
5. Новые педагогические и информационные технологии в системе образования: Учеб. пособие для студ. пед. вузов и системы повыш. квалиф. пед. кадров / Е.С. Полат, М.Ю. Бухаркин и др.; Под ред. Е.С. Полат. – М.: Академия, 1999. – 224 с.
6. Освітні технології: Навч.-метод. посіб. / О.М. Пехота, А.З. Кіктенко, О.М. Любарська та ін.; Заг заг. ред. О.М. Пехоти. – К.: А.С.К., 2001. – 256 с.
7. Пометун О., Пироженко Л. Інтерактивні технології навчання: теорія, практика, досвід. – К., 2002. – 135 с.
8. Селевко Г.К. Современные образовательные технологии. – М.: Народное образование, 1998. – 230 с.
9. Сучасний урок. Інтерактивні технології навчання: Наук.-метод. посібник./ О.І. Пометун, Л.В. Пироженко. За ред. О.І. Пометун. – К.: Видавництво А.С.К., 2004. – 192 с.: іл.

ВИКОРИСТАННЯ ІННОВАЦІЙНИХ МЕТОДІВ НАВЧАННЯ В КУРСІ ПРОГРАМУВАННЯ

І.В. Настенко

м. Київ, Національний педагогічний університет імені М.П. Драгоманова
inastenko@mail.ru

За останнє десятиліття відбувся якісний стрибок можливостей комп'ютерної техніки і телекомунікаційних технологій. Потужність комп'ютерних систем відкриває можливості рішення серйозних, професійних задач і, хоча програмне забезпечення поки не встигає за ростом продуктивності комп'ютерів, широта можливостей сучасних додатків росте досить швидко. Але разом з ростом можливостей росте і складність їхньої розробки, використання і вивчення.

У сфері користувальницького програмного забезпечення з'являється усе більше спеціалізованих програм для розв'язання конкретних задач в різних предметних областях. Багато програм пишуться під конкретну задачу і конкретного користувача, а, отже, їхнє використання більш зручне, ніж робота у відповідній універсальній програмі. Що стосується програмування, то практично всі сучасні професійні мови засновані на об'єктно-орієнтованому підході: C++, Java та ін. Широке поширення одержали системи візуального проектування і програмування, засновані на об'єктно-орієнтованій технології. Дані системи містять у собі як бібліотеки готових компонентів, так і програмні конструкції об'єктно-орієнтованих мов програмування. Таке сполучення дозволяє не тільки прискорити процес розробки додатків, але й сприяє ефективному вивченню даних систем.

Для багатьох компаній зростає значення програмного забезпечення, росте рівень автоматизації роботи підприємства. Сучасна індустрія вирішує питання пошуку методів автоматизації виробництва програмного забезпечення, підвищення його якості, а також зниження вартості і часу випуску на ринок. Ці методи засновані на компонентних технологіях, візуальному програмуванні, використанні зразків (pattern) і інструментальних середовищ (framework). За останні роки об'єктно-орієнтована технологія стала однією з основних при розробці професійного програмного забезпечення. Розробка будь-якої прикладної програмної системи починається з аналізу вимог, яким вона повинна буде задовольняти. У процесі даного аналізу визначаються призначення й умови експлуатації системи, а також розробляється її попередній проект. Тому при вивченні в школі та університеті розділу об'єктно-орієнтованого програмування, доцільно приділити значну увагу проектуванню об'єктно-орієнтованих систем.

При об'єктно-орієнтованому підході аналіз вимог до системи зводиться до розробки моделей цієї системи. Моделлю системи вважається формальний опис системи, у якому виділені основні об'єкти, що складають систему,

і відносини між цими об'єктами. У процесі розробки моделі дозволяють:

- перевірити працездатність розроблювальної системи на ранніх етапах її розробки;
- спілкуватися з замовником системи, уточнюючи його вимоги до системи;
- вносити (у разі потреби) зміни в проект системи (як на початку її проектування, так і на інших фазах її життєвого циклу).

Об'єктно-орієнтована розробка програмного забезпечення пов'язана з застосуванням об'єктно-орієнтованих методологій. Звичайно ці об'єктно-орієнтовані методології підтримуються інструментальними програмними засобами (CASE-системами), але і без таких засобів вони корисні, тому що дозволяють добре зрозуміти різні аспекти і властивості розроблювальної програмної системи, що в наступному істотно полегшує її реалізацію, тестування, супровід, розробку нових версій і більш істотну модифікацію.

Що стосується навчання програмуванню, то очевидно є необхідність вивчення об'єктно-орієнтованих мов. Причинами перспективності вивчення об'єктно-орієнтованого підходу, є зокрема, наступні:

- сучасність об'єктно-орієнтованого підходу. Його вивчення – це необхідність, тому що практично всі сучасні системи програмування розроблені на основі даного підходу і надають широкі можливості створення програм, що також є об'єктно-орієнтованими;
- підтримка в учнів мотивації до вивчення програмування. Використання засобів візуального програмування приваблює швидкістю розробки додатків, а також припускає знання основ об'єктно-орієнтованого програмування для створення нових компонентів.

Навчання об'єктно-орієнтованого програмування вимагає особливу увагу приділити наступним аспектам:

- формуванню знань основ технології об'єктно-орієнтованого проектування і програмування;
- формуванню навичок аналізу готових проектів;
- формуванню навичок розробки проектів;
- формуванню представлення про різні засоби розробки проектів.

Практика показує, що вивчення сучасних професійних середовищ підвищує мотивацію учнів до вивчення програмування. Тому важливо розглядати на заняттях не тільки навчальні приклади для формування навичок роботи з конкретними структурами даних або алгоритмами, а також і реальні задачі, що допоможуть сформуванню представлення про сучасний процес проектування і розробки програмного забезпечення.

Назване реалізується через чітку мотивацію необхідності вивчення об'єктного проектування і програмування. Цьому ж сприяє приведення прикладів використання досліджуваного матеріалу в професійній діяльності. У процесі виконання практичних завдань роль викладача полягає в постановці задачі й у керуванні процесом її рішення на ключових етапах. Спе-

цифіка вивчення об'єктного програмування та проектування додатків полягає в тому, що учнем визначаються значимі функції системи і ставиться задача на її розробку, а вони самостійно виявляють основні складові проекту, їхнього взаємозв'язку і шукають шляхи оптимального рішення задачі.

Дослідження показують, що ефективність навчання підвищується, коли учні самі будують моделі, а не одержують їх у готовому виді (чим обумовлено вибір методу проектів як провідного в навчанні даного розділу). Дійсно, розробляючи проект, школярі та студенти відтворюють ознаки об'єкта, його структуру. У процесі побудови і подальшої роботи з проектом засвоюються теоретичні знання, створюється можливість для самостійної розробки.

Засоби візуального програмування дозволяють швидко і просто розробляти додатки. Однак у процесі вивчення даних середовищ важливо акцентувати увагу на формуванні в учнів розуміння структури об'єктів, представлених у середовищі графічними компонентами.

Процес навчання повинен стимулювати учнів використовувати отримані знання в рішенні практичних задач, аналізувати і перетворювати навколишню дійсність, виробляючи власні знання. Для цього використовується аналіз прикладів і ситуацій з реального життя [2]. Проблема відірваності освіти від реальних життєвих, і в тому числі професійних, проблем, що зустрічаються на шляху кожного випускника школи і в університету, актуальна сьогодні як в українській, так і в закордонній школі.

Необхідний розгляд на заняттях прикладів працюючих систем. Завдання на розробку повинні бути узяті з реальних ситуацій, нехай і в спрощеному виді. Учні повинні уявляти собі весь процес розробки, а не тільки його фрагменти.

Форми і методи навчання об'єктно-орієнтованому програмуванню багато в чому задаються цілями навчання, змістом курсу й умовами навчання [4]. Застосовувана система методів навчання спрямована на формування в учнів знань, умінь і навичок розробки проектів, а також застосування готових програмних і мовних засобів їхньої реалізації. Особлива увага приділяється формуванню творчих якостей особистості, дослідницьких умінь і навичок, вихованню самостійності учнів і позитивної мотивації в процесі вивчення навчального матеріалу.

Основним методом навчання є метод проектів, метод демонстраційних прикладів та ін. Це обумовлено, насамперед, практичною спрямованістю навчання. Крім того, ефективне вивчення середовищ розробки неможливо без розгляду та самостійного вирішення завдань з реального життя. Задача учителя інформатики полягає в тому, щоб за короткий час допомогти учню опанувати досвідом, спираючись на зразки, подані до розгляду за допомогою проблемних методів навчання, спрямовувати та контролювати дії учнів під час роботи над проектом.

Рекомендації щодо застосування методу проектів на уроках докладно

висвітлено в роботах авторів: О.Н. Арініної, Л.І. Богомолової, В.В. Гузеєва, Н.В. Матяша, П.А. Петрякова, Д. Пітта, Г.К. Селевко, Н.В. Турчиної, І.Д. Чечеля. Метод демонстраційних прикладів докладно описаний у роботах В.В. Лаптева, Л.М. Фрідмана та ін. [5].

Навчання об'єктному програмуванню включає розгляд готових проєктів і докладний аналіз послідовності їхньої розробки, а також створення учнями власних проєктів додатків.

Програмування, як елемент курсу інформатики, на різному рівні вивчається в багатьох навчальних закладах. Необхідність його вивчення не вимагає пояснення. Однак, програми, що розробляються на заняттях, як правило, не складні і призначені для формування навичок роботи з конкретними структурами даних або алгоритмами. Вони можуть бути задумані і розроблені одною людиною. Такі системи, як правило, мають дуже обмежену область застосування і короткий час життя. Розробка подібних програм сприяє формуванню навичок програмування, однак не дає уявлення про сучасний процес розробки програмного забезпечення. Професійні ж системи відрізняються високим рівнем складності: один розробник практично не в змозі охопити всі аспекти такої системи. Тому використання методу проєктів в навчанні даного розділу є найбільш доцільним, оскільки дозволяє сформулювати в учнів навички роботи в співробітництві, колективного вирішення проблем, сприяє згуртуванню команди. Наявність у професіонала вказаних навичок є необхідною умовою успішної роботи в сфері програмування великих додатків.

Важливо сформувати у школярів та студентів розуміння того, що програмування – це не тільки знання структур даних і окремих алгоритмів, але й уміння розробити проєкт системи, оптимально розділити його на складові, установити між ними зв'язки тощо. Ще однією розповсюдженою помилкою є представлення про те, що відсутність синтаксичних помилок у програмі і її виконання для одних даних є гарантією правильності програми.

Необхідно приводити приклади задач, таких як, наприклад, системи керування і контролю за реальними процесами (наприклад, диспетчеризація морського або повітряного транспорту). Це дозволить сформувати в студентів та учнів розуміння того, що ключовим питанням розробки програмних систем є аспекти проєктування й опису поведження системи, адекватними ідеології системи засобами.

Необхідно показати переваги об'єктно-орієнтованого підходу як більш сучасного і такого, що дає широкі можливості у професійній розробці програм. Зробити це на практиці не легко. Порівняння декомпозиції двох простих програм не буде мати наочного ефекту, а складних не знайде розуміння. Тому в даному випадку можливо теоретичне пояснення суті і переваг об'єктної декомпозиції. Один з них – зменшення розміру програми за рахунок повторного використання загальних механізмів. Вагомим аргументом на користь об'єктно-орієнтованого підходу може бути те, що він відбиває

топологію новітніх мов високого рівня.

Система лабораторних робіт являє собою *загальний проект, який має кінцевий результат, значущий для особистості учня* (такий, що впливає на моральний аспект – проекти, що розробляються для потреб та за прохання інших людей; надає матеріального стимулу – на замовлення фірми, для продажу через Інтернет та ін.). Відповідно, *послідовність і сутність системи лабораторних робіт* представлена таким чином, щоб включити *всі етапи роботи над проектом*, а саме [3]:

- 1) постановка проблеми і визначення цілей проекту;
- 2) визначення задач, очікуваних результатів і процедури їхньої оцінки;
- 3) дослідницька і/або практична діяльність, необхідна для одержання зазначених у проекті результатів;
- 4) аналіз отриманої інформації і формулювання висновків;
- 5) представлення результатів;
- 6) оцінка проектної діяльності і результатів проекту (колективне обговорення).

Прикладом фрагменту лабораторної роботи етапу проектування моделі системи може бути наступна (клас об'єднано в групи, кожна з яких виконує свій проект, але вимоги до результату проекту для всіх груп однакові). Наприклад, якщо одна з груп займається проектом, на меті якого є автоматизувати роботу шкільної бібліотеки, то учні визначають (лабораторна робота належить до третього етапу реалізації проекту, поданого вище):

1. Вимоги до системи:
 - визначають, з чого повинна складатися бібліотека (абонемент та читальний зал);
 - що система повинна зберігати (інформацію про книги, читачів, видачі книжок на руки і в читальному залі).
2. Визначають, яку звітну інформацію видаватиме система (спільна кількість читачів, список боржників).
3. Визначають головні функції системи (введення і обробка даних про книги, введення і обробка даних про читачів абонементу і читального залу).
4. Визначають класи, виходячи зі значущих функцій системи:

<i>Назва класу</i>	<i>Опис класу</i>
Бібліотека	Абстрактний клас. Основний клас системи, відповідальний за керування роботою
Абонемент	Абстрактний клас
Читальний зал	Абстрактний клас
Книга	Клас, відповідальний за зберігання та обробку інформації, пов'язаної з книгами
Читач	Клас, відповідальний за зберігання і обробку інформації, пов'язаної з читачами бібліотеки
Картка читача	Клас, відповідальний за зберігання і опрацювання інформації, пов'язаної з видачею книжок

5. Визначають атрибути класів, виходячи з вимог до системи:

Бібліотека (Номер, Адреса)

Читач (Номер абонементу, Прізвище, Ім'я, Адреса, Телефон)

Книга (Шифр, Назва, Автор, Видавництво, Рік видавництва, Ознака наявності)

Запис про книгу (Дата видачі, Строк, Дата повернення)

Замовлення книги (Дата)

6. Визначають зв'язки між класами, виходячи з вимог до системи:

<i>Клас</i>	<i>Тип зв'язку</i>	<i>Потужність зв'язку</i>	<i>Коментар</i>
Бібліотека–Читач	Агрегація	1–n	Читач є частиною бібліотеки
Бібліотека–Книга	Агрегація	1–n	Книга є частиною бібліотеки
Читач–Замовлення книги	Асоціація	1–n	Замовлень може бути багато
Читач–Запис про книгу	Асоціація	1–n	Записів про книги в абонементі читача може бути багато
Замовлення книги–запис про книгу	Агрегація	1–0...1	Після замовлення книга може бути узята читачем, а може й ні (якщо вона на руках).
Замовлення книги–Книга		1–1	Замовлення оформляється на одну книгу.

7. Визначають методи класів

8. Графічно зображають розроблену ними модель системи.

Кожен з етапів роботи учні обговорюють в групі, виносять спільне рішення та висвітлюють його на розсуд загалу. Графічно зображені зв'язки між об'єктами подають у звітній формі іншим групам класу, після чого на заключному етапі лабораторної роботи кожна з груп виказує свої зауваження та пропозиції.

Приклади вправ, що можуть надаватися вчителем до даного проекту:

1. Крім перерахованих вимог, необхідно зберігати дату народження читача. Змініть модель таким чином, щоб відбилася дана вимога.

2. Крім перерахованих вимог, необхідна можливість обчислення загального числа замовлень за термін. Змініть модель так, щоб відбилася дана вимога.

3. Крім книг, читачі можуть замовляти періодичні видання. Створіть відповідний об'єкт у моделі й встановіть потрібні зв'язки й атрибути.

В даному випадку дослідницька діяльність є основним змістом циклу лабораторних робіт. Учитель при цьому займає, крім керуючої і контролюючої, позицію співтворця проекту. Особлива увага при цьому приділяється такій інноваційній педагогічній технології, як навчання в співробітництві. Пропонована система лабораторних робіт дозволяє послідовно, від простого до складного, сформувати в учнів уявлення про об'єктно-орієнтоване про-

грамування і навички розробки проектів. Структура лабораторних робіт дає можливість ефективно навчати школярів і студентів з різним рівнем підготовки, тим самим, забезпечуючи диференційований підхід до навчання.

Система вправ, пропонованих учням в ході роботи, передбачають засвоєння ними двох способів роботи з проектами: вивчення готових проектів і їхня модифікація, та розробка нових проектів, що і є на меті навчання об'єктно-орієнтованому програмуванню.

Література:

1. Беспалько В.П. Слагаемые педагогической технологии. – М., 1989.
2. Монахов В.М. Информационная технология обучения и реформа школы // Вопросы психологии. – 1988. – №2.
3. Новые педагогические и информационные технологии в системе образования: Учеб. пособие для студ. пед. вузов и системы жизни повыш. квалиф. пед. кадров / Е.С. Полат, М.Ю. Бухаркина, М.В. Моисеева, А.Е. Петров; Под ред. Е.С.Полат. – М.: Издательский центр «Академия», 2001. – 272 с.
4. Інформатика. Програми для загальноосвітніх навчальних закладів. – Запоріжжя: Прем'єр, 2003. – 304 с.
5. Фридман Л.М. Наглядность и моделирование в обучении // Новое в науке и технике. Серия «Педагогика и психология». – М.: Знание, 1984.

МЕЖПРЕДМЕТНЫЕ СВЯЗИ ПРИ ОБУЧЕНИИ ИНФОРМАТИКЕ В СРЕДНЕЙ ШКОЛЕ

Т.В. Кабанова
г. Санкт-Петербург, СШ №58
school58@spb.edu.ru

Актуальность межпредметных связей в обучении обусловлена современным уровнем развития науки, в котором ярко выражена интеграция общественных, естественнонаучных и технических знаний.

Межпредметные связи разрешают существующие в предметной системе обучения противоречия между разрозненным по предметам усвоением знаний учащимся и необходимостью их синтеза, комплексного применения в практике трудовой деятельности и жизни человека. Комплексное применение знаний из различных предметных областей – это закономерность современного производства, решающего сложные технические и технологические задачи. Умение комплексного применения знаний, их синтеза, переноса идей и методов из одной науки в другую лежит в основе творческого подхода к научной, инженерной, художественной деятельности человека.

Проблема развития межпредметных связей уже давно разрабатывается преподавателями и методистами: как научить учащихся использовать знания, приобретенные при изучении не одного предмета, а полученные в целостной системе обучения [8]. Подходы к реализации межпредметных связей предлагают современные исследователи в области методики информатики, физики и математики Г.Д. Глейзер, Б.В. Гнеденко, Я.И. Грудёнов, В.А. Далингер, И.Д. Зверев, А.А. Коротченкова, В.Н. Максимова, А.Д. Мышкис, З.И. Слепкань и другие. Так, в исследовании В.Н. Максимова [5; 7] доказано, что систематизированная и целеустремленная реализация межпредметных связей является одним из дидактических принципов организации учебно-познавательной деятельности личности.

Целью данной статьи является показать возможность осуществления межпредметных связей информатики, математики и физики на примере изучения работы в табличном процессоре Excel, использования пакета Derive и других программных средствах.

В образовании межпредметные связи являются важным средством формирования практических навыков и умений применять свои знания из одного предмета при изучении других предметов: решать задачи, выполнять разные чертежи и рисунки, строить графики, делать расчеты с помощью специальных программных средств; позволяет углубить и расширить знания школьников. При таком подходе к обучению они овладевают рациональными приемами как практических, так и умственных действий, что является одним из рациональных путей развития умственных способностей учащихся. При осуществлении межпредметных связей достигается общее и одно-

временное овладение многими понятиями нескольких предметов. Знания, полученные во время изучения информатики и математики, являются базовыми для овладения другими дисциплинами, от них зависит прочность всей совокупности знаний, приобретенных при обучении в школе, поэтому изучение информатики должно быть серьезным и вдумчивым. Обучение должно вестись на уровне межпредметных связей, чтобы школьники в дальнейшем умели систематизировать, анализировать знания и применять их в будущем.

Изучение информатики в школе должно развивать алгоритмическое мышление, изучать методы информатики как науки, формировать элементы информационной культуры школьников. При этом важными являются разделы алгоритмизации и программирования, изучение которых позволяет решать как общеобразовательные задачи, задачи ознакомления с основными положениями и понятиями информатики, так вопросы подготовки квалифицированных пользователей.

Школьная информатика – это общеобразовательный предмет, целью которого является формирования элементов информационной культуры школьника, понимания им информационных процессов, происходящих в окружающем мире, развития умственных способностей учащихся, учебно-информационных умений. Это обеспечивает поиск, обработку и использование информации для решения учебных задач. Исходя из информатизированного понимания учебно-воспитательного процесса, выделять следующие группы учебно-информационных умений: умение воспринимать учебную информацию; умение библиотечно-библиографического поиска; умение работать с текстом учебника; умение работать с компьютером и другие информационные умения, которые являются основными компонентами информационной культуры [2].

Мы исходим из того, что этот курс должен быть подспорьем при самостоятельном овладении знаниями, направленным на систематическое развитие общих умственных способностей учащихся и подготовить их к эффективному изучению других предметов. Такое построение предмета информатики придаст ему гуманитарный, общеобразовательный характер, подготовит учащихся к дальнейшему образованию и к практическому использованию полученных знаний в жизни.

Так, с помощью электронной таблицы Excel ученик может решать многие задания, требующие выполнения различных математических операций, может предварительно проанализировать решение задачи и даже спрогнозировать те или иные результаты. На основе вычислительного эксперимента учащийся сможет прийти к формулированию гипотез относительно исследуемых закономерностей, получит возможность быстро проверить их правильность ли опровергнуть их, приведя контраргументы. Excel позволяет быстро овладеть умениями работать с компьютерными системами.

Применение графических возможностей процессора Excel позволяет в

максимальной степени учитывать образность и конкретность мышления школьников, уменьшить технические трудности при выполнении заданий, побуждать учеников к творчеству. При изучении графики у большинства учащихся появляется стойкий интерес к работе на компьютере. Многочисленные исследования психологов показали, что такие занятия необходимы для полноценного развития личности.

Умственная активность человека, его мыслительная деятельность заключается в понимании и переработке воспринимаемой информации, а также в установлении различных типов связей – причинно-следственных, логических, пространственных, временных. В психологии разработан ряд приемов умственной деятельности учащихся в процессе усвоения знаний. Проблемой формирования приемов умственной деятельности школьников занимались Л.С. Выгодский, В.В. Давыдов, П.Я. Гальперин, А.Н. Леонтьев, Д.Б. Богоявленская и др.

В своей работе мы следовали определению Д.Б. Богоявленской, которая под приемом умственной деятельности понимает «систему процессов или операций анализа, синтеза, абстракции, обобщения и других, специально организованных для решения задач – проблем определенного типа и разной степени обобщенности» [1].

На уроках при выполнении заданий учащимся предоставляется возможность выбора заданий уровня, который они в данный момент могут осилить. Это дает им возможность самокритично оценивать свои знания и четко знать основные требования к выполнению сложных заданий.

Деревянный брусок лежит на доске. Доску поднимают за один конец. Как зависит абсолютная величина силы трение, которая действует на брусок, от угла наклона доски α ? При каком значении угла наклона брусок начнет скользить вдоль доски? Коэффициент трения между бруском и доской $\mu=0,3$, масса бруска $m=1$ кг (рис. 1).

Решение системы уравнений
$$\begin{cases} F_{T.C} = 9.8 \cdot \sin \alpha \\ F_{T.K} = 0.3 \cdot 9.8 \cdot \cos \alpha \end{cases}$$
 даёт искомое значение α .

С помощью программы Derive графическим способом решим данную систему уравнений.

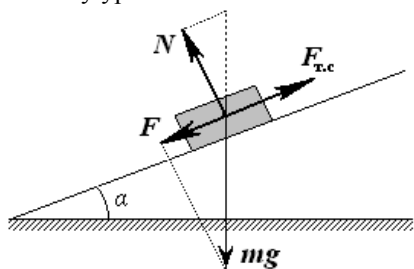


Рис. 1

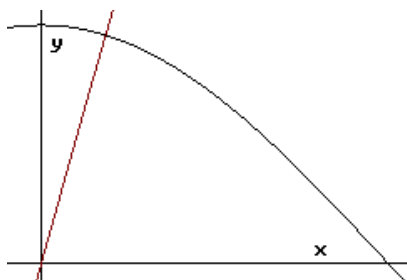


Рис. 2

Координата точки пересечения графиков зависимостей $F_{TC}=9.8 \cdot \sin \alpha$ и $F_{TK}=0.294 \cdot \cos \alpha$ относительно оси абсцисс и есть искомым значением угла α (рис. 2). Имеем: $\alpha=0.288$ ($\alpha=16.50^\circ$). От 0° до α сила трения возрастает по закону синуса, а потом, от 0° до 90° , она уменьшается по закону косинуса.

А.Н. Леонтьев фактически ставит знак равенства между знаниями и умственным развитием, так как последнее, по его мнению, полностью определяется характером «присвоенного» человеком родового опыта, приобретенного в тех социально-исторических условиях, в которых ребенок живет и развивается. По мнению А.Н. Леонтьева, для формирования умственной активности личности необходимы две системы знаний: о предметной действительности; о содержании и последовательности умственных действий, обеспечивающих овладение первой системой знаний.

Для развития мышления учащихся следует формировать у них обобщенные приемы рассуждений, обучать методам решения целого класса задач. Основой этого утверждения является теория В.В. Давыдова [3] о содержательных обобщениях, суть которой состоит в том, что с самого начала изучения предмета учеников нацеливают на построение и усвоение обобщенного способа ориентации в данной области знаний, обобщенного способа решения достаточно обширных классов задач. Ученикам может быть предложен обдумать задачу, определить способ решения, реализовать его на практике и показать найденный результат. Моделирование таких ситуаций общения на уроках математики включает в себя дидактические возможности для повышения интереса учеников, развития их учебно-коммуникативных умений.

Рассмотрим пример задачи

Источник тока с э.д.с. E внутренним сопротивлением r замкнут на переменное сопротивление R . Исследовать, как изменится полезная мощность P в цепи в зависимости от тока.

На уроке физики учащиеся выводят зависимость $P=iE-i^2 \cdot r$. Исследуют эту зависимость и строят график на уроках физики и математики (рис. 3). Но без компьютера на уроке сложно провести углубленный анализ зависимости полезной мощности, например, от э.д.с. и внутреннего сопротивления, развивать творческие способности учащихся. Применение компьютера умение самостоятельно овладевать любой обучающей системой и умение подбирать программные средства для выполнения задач; умение работать с программами Derive, Maxima, Matlab, Mathematica и др.

И, кроме того, анализ графиков (рис. 3) позволяет получить целый ряд интересных результатов. Например, максимальное значение полезной мощности наступает при значениях тока, который вдвое меньше тока короткого замыкания.

Пользуясь программами компьютерной математики, при наличии соответствующих условий, учитель может объединять при преподавании нового материала метод целесообразных задач с исследовательским методом как в

классах с углубленным изучением информатики, математики, так и в классах общеобразовательного профиля. Знание, полученные через открытие, более прочные и имеют значительное влияние на развитие умственных способностей личности.

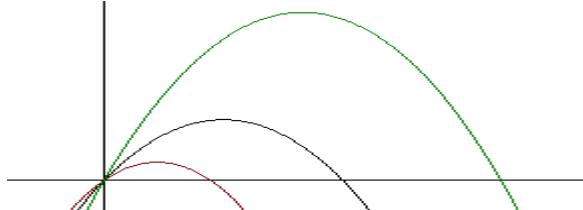


Рис. 3

Например, при изучении темы «Квадратный трехчлен» ученикам нужно выяснить и сделать вывод о том, как строить параболы с использованием простейших преобразований графиков. Для этого предлагаем школьникам построить по нескольким параболом любой квадратный трёхчлен с помощью преобразований типа $y=ax^2$, $y=(x+m)^2$ и других, и составить алгоритмы построения. Для школьников с высоким творческим потенциалом можно предложить темы научно-исследовательских работ: «Исследование квадратного трехчлена», «Квадратный трехчлен в задачах с параметрами» и прочие. Кроме того, необходимо учащимся научиться находить корни уравнения ax^2+bx+c с помощью графика. Для наглядности, поставленную задачу нахождения приближенных корней уравнение $f(x)=0$ с заданной точностью ε целесообразно изобразить графически. Представляется возможным геометрически проиллюстрировать такие понятия, как «простой корень», «кратный корень», «корень отсутствующий».

Выводы

Межпредметные связи информатики, математики и физики способствуют интеллектуальному развитию учащихся на основе формирования представлений о целостности знаний, позволяют сформировать навыки владения не только декларативными знаниями (что?), но и процедурными (как?). Реализация межпредметных связей с использованием межпредметных взаимосвязанных заданий позволяет учащимся достичь высокого уровня информационной и коммуникативной культуры. На этой основе у учеников создается стойкий интерес к обучению и уверенность в собственных силах и возможностях, потребность в самосовершенствовании.

Литература:

1. Богдавленская Д.Б. Психология творческих способностей: Учеб. пособие для студ. высш. учеб. заведений. – М.: Академия, 2002. – 320 с.
2. Гендина Н.И., Колова Н.И., Скипа И.Л., Старобубова Г.А. Формирование информационной культуры в библиотеках и образовательных учреждениях: Учебно-метод. пособие 2-е изд., перераб. – М.: Шк. б-ка, 2003. –

296 с.

3. Давыдов В.В. Теория развивающего обучения. – М.: ИНТОР, 1996. – 544 с.

4. Далингер В.А. Межпредметные связи математики и физики. – Омск: Изд-во ОГПИ, 1991. – 95 с.

5. Зверев И.Д., Максимова В.Н. Межпредметные связи в современной школе. – М.: Педагогика, 1981. – 160 с.

6. Коротченкова А.А. Межпредметные связи математики и информатики при подготовке специалистов экономического профиля: Автореф. дис. ... кад. наук: 13.00.02/ Орловский гос. ун-т. – Орел, 2000. – 16 с.

7. Максимова В.Н. Межпредметные связи в процессе обучения. – М.: Просвещение, 1988. – 190 с.

8. Психолого-педагогический словарь для учителей и руководителей общеобразовательных учреждений. – Ростов-на-Дону: Феникс, 1998. – 544 с.

ВИКОРИСТАННЯ КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ НА УРОКАХ У ПОЧАТКОВИХ КЛАСАХ

Л.І. Петрушина
м. Кривий Ріг, Криворізька загальноосвітня школа №99
school99@mail.ru

Із засобами інформаційних і комунікаційних технологій діти сьогодні зустрічаються значно раніше, ніж починається їх систематичне вивчення у старших класах. Це і комп'ютерні клуби, і персональні комп'ютери вдома та у близьких знайомих. Інформаційні та комунікаційні технології формують найближче інтелектуальне оточення дітей, впливають на формування їхнього навчального та побутового середовища, структуру навчальної діяльності, розвиток особистісних якостей та на результативність навчання.

У нашій школі впровадження викладання елементів інформатики розпочинається з початкових класів, але, звичайно, з урахуванням вікових особливостей учнів, специфіки динаміки їхнього розвитку, завдань навчання та виховання, які є головними в процесі формування особистості у дітей цього віку.

В першому класі формування основ комп'ютерної грамотності учнів розпочинається із засвоєння можливостей графічного редактора. Користуючись його інструментами, діти оволодівають навичками роботи з маніпулятором «миша», а звертання до команд графічного редактора сприяє формуванню основних навичок роботи за комп'ютером. Функціональні можливості редактора використовую на уроках навчання грамоти в букварний період. Це проведення диктантів на написання друкованих літер, пізніше, коли діти більш впевнено оволодівають роботою маніпулятором «миша» – рукописних букв.

Цікавим видом роботи для дітей на уроці є гра, бо в ній присутній елемент очікування несподіванки, або елемент змагання, елемент загадки, який інтригує дітей. Наприклад, гра *«Допиши елементи букв, яких не вистачає»*. На екрані монітора півовали, петлі, похилі лінії, які діти повинні перетворити на літери. Або інша гра – *«Прочитай тайнопис»*:

Л С В П P S O R Б У J Й W B S I R Д J Г A V Д A Й

Фраза складена з букв українського алфавіту, які перемішані з буквами інших мов. Щоб її зрозуміти, слід читати тільки наші літери, пропускаючи ті, яких немає в українському алфавіті.

Використовувати комп'ютер в школі можна під час викладання будь-яких предметів, адже комп'ютерна підтримка на уроці дає значний педагогічний ефект, полегшуючи, розширюючи, поглиблюючи теоретичну базу знань, надаючи результатам навчання практичного значення, творчого спрямування, розкриває творчий потенціал учнів і вчителя. Навчальні комп'ютерні ігри стають серйозною підготовкою учнів у засвоєнні елемен-

тів інформатики: командою, програмою, її виконанням т.п.

У 2-3 класах на уроках математики під час вивчення табличних випадків множення та ділення ми використовуємо комп'ютерну гру «*Математичний калейдоскоп*», яка в залежності від типу уроку допомагає перевірити рівень сформованості знань вивчених таблиць множення та виявити недоліки або тренувати дітей у її засвоєнні.

Гра розрахована на 3–5 хвилин уроку математики або позакласного заняття. Учні працюють колективно. На екрані монітора комп'ютера знаходиться таблиця з прикладами на табличне множення, наприклад, числа 9. Учитель називає ці випадки або показує. Діти намагаються правильно відповісти. Якщо відповідь була вірною, приклади з екрана зникають, і перед очима дітей поступово з'являється зимовий пейзаж (репродукція картини, ілюстрація). По закінченню гри комп'ютер оцінює роботу дітей.

У другому класі на підсумковому уроці читання за темою «Українські народні казки» цікаво проходить комп'ютерна гра «*Казкова вікторина*». Спочатку перед учнями на екрані монітора (на першому слайді) з'являється загадка до певної казки, яку вони повинні відгадати. Наступний слайд дає змогу перевірити вірність або, навпаки, невірність відповіді. На екрані – назва казки та ілюстрація до неї.

Комп'ютерні ігри «*Математичний калейдоскоп*» та «*Казкова вікторина*» створені на основі програми Microsoft PowerPoint.

Завдання та задачі, використані на уроках, пов'язані з практичною реалізацією функціональних можливостей комп'ютера і подаються із зростанням рівня їх складності.

Коли діти оволодівають навичками роботи з клавіатурою комп'ютера, його можна використовувати для перевірки вмінь писати словникові слова або засвоєних знань, умінь та навичок за темами: «Ненаголошені –е–, –и– в коренях слів», «Вживання апострофа», «Префікси з–, с–», для проведення мовних тем, тематичних оцінювань, поурочного контролю тощо.

На уроках української мови за темою «Будова слова» можна використати гру, складену за допомогою програми «Тести». У грі враховуються різні розумові здібності учнів, тому і завдання для дітей із середнім та низьким рівнем значно легші, ніж для учнів з достатнім та високим рівнем. Якщо перенести схему гри на папір, то вона буде мати вигляд трикутника, на якому поступово кружечками висвічується схема відповіді учня. Кожне наступне завдання дитина одержує від комп'ютера в залежності від вірності відповіді на попереднє запитання. В кінці гри комп'ютер підводить підсумок роботи кожного учня й оцінює відповіді словесно та в балах, що є найбільш цікавим для школярів.

Інша гра, «*Моє Криворіжжя*» – гра-подорож. За допомогою комп'ютера діти мандрують рідним містом, поглиблюючи свої знання про його сучасне та минуле, про історичні постаті, здобутки та надбання. Принцип гри полягає в тому, що свою подорож діти здійснюють від запитання до

запитання. Власні відповіді звіряють та уточнюють з відповідями комп'ютера, який ще й надає їм інструкції про подальші дії: повернутися до попереднього запитання чи перейти до наступного. У підсумку комп'ютер оцінює роботу учнів.

Ефективно використовувати інформаційно-комунікаційні технології у навчальному процесі можна на будь-якому уроці.

Комп'ютер допомагає вчителю стати для своїх вихованців справжнім чарівником, бо він може творити дива на кожному уроці: чи це читання, чи урок української мови, чи заняття з курсу «Я і Україна». Уявіть собі, коли важке правило підказує комп'ютер, а складна формула падає з неба, бумерангом з'являються завдання казкової вікторини або відкриваються жалюзі, а за ними ховаються приклади для змагання «Кращий математик». Це учні не в змозі забути.

ПІДГОТОВКА ДО ОЛІМПІАД З ПРОГРАМУВАННЯ СТУДЕНТІВ ПЕДАГОГІЧНИХ ВНЗ

О.Р. Гарбич

м. Дрогобич, Дрогобицький державний педагогічний університет

імені Івана Франка

moshora@westukrtrans.com

Реформа загальноосвітньої і професійної школи вимагає від майбутніх педагогів фізико-математичного профілю ґрунтовного оволодіння методами ефективного використання сучасної обчислювальної техніки в навчальному процесі. Загальна комп'ютеризація освіти призвела до виникнення непростих проблем, які позначилися на методі організації олімпіад з інформатики.

Основні завдання, що стоять перед олімпіадами, є:

- зацікавлення студентів інформатикою;
- виділення найздібніших студентів;
- ознайомлення студентів і викладачів з новими завданнями для подальшої роботи;
- аналіз курсів інформатики, визначення рівня підготовки студентів.

Олімпіада є також і навчальним засобом. Під час розв'язування, а особливо розбору задач студенти ознайомлюються з новими методами й алгоритмами.

Олімпіади з інформатики в Україні регулярно проводяться з 1987 року. Студентам для розв'язування пропонуються оригінальні задачі. За обмежений проміжок часу слід розробити алгоритми розв'язання задач і реалізувати їх, скориставшись будь-якою мовою програмування. Часто олімпіада проходить у два тури (заочний і очний).

Під час оцінювання програми велике значення надається її робочим якостям. Звертається увага на швидкодію програми, точність результату, кількість необхідної пам'яті. Різноманітні обмеження, що містяться в умовах, лімітовані час і об'єм пам'яті задають необхідний рівень ефективності розв'язання та якості алгоритму.

Для розв'язування олімпіадних задач необхідні метод частинних цілей, підйому та відпрацювання назад, метод відступу, метод розгалужень та границь, метод евристики та рекурсія.

Метод частинних цілей, підйому та відпрацювання назад пов'язаний із умовами складної задачі до послідовності більш простих задач. Звичайно ми сподіваємось на те, що більш прості задачі легше піддаються обробці, ніж початкова задача, а також на те, що розв'язок початкової задачі може бути одержано із розв'язків більш простих задач. Така процедура називається методом частинних цілей.

Цей метод виглядає дуже розумно. Але, як і більшість загальних методів розв'язання задач, його не завжди легко перенести на конкретну задачу.

Не існує загального набору правил для визначення класу задач, які можна розв'язати за допомогою такого підходу.

Метод розробки алгоритмів відомий під назвою метод підйому. Алгоритм підйому починається із прийняття початкового припущення або обчислення початкового розв'язку задачі. Потім починається рух “вгору” від початкового розв'язку в напрямку до кращих розв'язків. Коли алгоритм досягає такої точки, із якої більше неможливо рухатись вгору, алгоритм зупиняється. Немає гарантії, що кінцевий розв'язок, одержаний за допомогою алгоритму підйому, буде оптимальним. Цей дефект часто обмежує застосування методу підйому.

Назва “підйом” частково походить від алгоритмів знаходження максимумів функцій декількох змінних.

Метод відпрацювання назад: починаємо з цілі або розв'язку і рухаємось назад в напрямку до початкової постановки задачі. Якщо ці дії оборотні, рухаємось назад від постановки задачі до розв'язку.

Евристичний алгоритм визначається як алгоритм, що знаходить хороші, але необов'язково оптимальні розв'язки, швидший і простіший в реалізації. Мета побудови евристичного алгоритму – створити алгоритм, що гарантує виконання вимог, які легко задовольнити, але необов'язково тих, які не так легко задовольнити.

Метод розробки алгоритму, відомий як програмування з відступом назад, можна описати як організований вичерпний пошук, який часто дозволяє уникнути дослідження всіх можливостей. Цей метод особливо зручний для розв'язання задач, які потребують перевірки потенціального великого, але скінченного числа розв'язків.

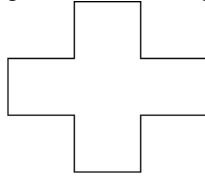
Метод розгалуження і границь подібний до методу з відступом назад тим, що він досліджує деревоподібну модель середовища розв'язків і застосовний до широкого кола дискретних комбінаторних задач. Алгоритми з відступами націлені на те, щоб знайти одну або всі конфігурації, які моделюються N -вимірними векторами, що задовольняють певним властивостям. Алгоритми розгалужень і границь орієнтовані в більшій мірі на оптимізацію. В розв'язаній задачі визначена числова функція ціни для кожної із вершин, які з'являються у дереві пошуку. Мета – знайти конфігурацію, на якій функція ціни досягає максимального або мінімального значення.

Приклади:

1. Зважування монет (підйом). Є 25 золотих монет. Всі вони однієї ваги, за винятком однієї монети з дефектом, яка має найменшу вагу. Розробіть алгоритм, який визначає дефект за три зважування. Яка максимальна кількість монет, для яких можна визначити монету з дефектом не більше, ніж за три зважування на терезах?

2. Задача розрізання паперу (часткові цілі). На малюнку показаний грецький хрест, який складається із п'яти однакових квадратів. Роблячи тільки два прямих розрізи, розріжте хрест так, щоб можна було скласти його час-

тини в прямокутник, одна сторона якого в два рази довша за іншу.



3. Гра (відпрацювання назад). Три людини грають у гру, в якій один виграв, а два інших програють. Кожен, хто програв, дає тому, хто виграв, суму, рівну кількості грошей, яка була у того, хто вигравав на початку гри. Було зіграно три гри, кожен гравець виграв один раз. У першого гравця виявилось 4 шекелі, у другого – 20 шекелів, а в третього – 6 шекелів. Скільки грошей було у кожного гравця на початку гри?

За декілька років роботи з проблемною групою у нас склалась певна система відбору і підготовки студентів до участі їх у олімпіадах. Аналіз завдань, які пропонуються на олімпіадах, показав, що:

– студентам необхідно мати велике бажання розв'язати задачу і бути наполегливим у пошуках її розв'язку;

– на олімпіадах успішно виступають ті студенти, які мають ґрунтовні знання, володіють нестандартним баченням і нешаблонним мисленням;

– задачі не є вправами на перевірку знань і умінь застосовувати стандартні методи, частіше всього вони вимагають значно більшої бази знань з окремих розділів інформатики, деякі з яких вивчаються в педуніверситетах мало або зовсім не вивчаються.

На даному етапі розвитку освіти робота із здібною, обдарованою молоддю поширюється, у багатьох вищих навчальних закладах почали працювати студентські гуртки з інформатики, а олімпіади стали традиційною формою підсумків позакласної роботи. Олімпіада – найвідповідальніший етап у роботі юного програміста, інформатика. З кожним роком зростає кількість учасників олімпіад, їх майстерність.

Отже, старання систематична робота в гуртку і в процесі самостійної роботи – необхідна умова успішного виступу на олімпіаді.

Література:

1. Богомолова Н.І. Досвід проведення олімпіад з інформатики // Комп'ютер в школі та сім'ї. – 2004. – №1.

2. Сканер В.О. Олімпіада: досвід, ідеї, побажання // Комп'ютер в школі та сім'ї. – 2003. – №8.

3. Петровський С. Підготовка і проведення учнівських олімпіад з програмування з використанням методу проектів // Інформатика. – 2004. – №31-32.

ВИКОРИСТАННЯ Е-ТЕХНОЛОГІЙ У СУЧАСНОМУ ОСВІТНЬОМУ ПРОСТОРИ

Д.М. Величко

м. Харків, Харківський національний педагогічний університет
імені Г.С. Сковороди
dmitro_v@mail.ru

Останнє десятиріччя ознаменувалося значним розвитком Інтернету, що відкрило нові горизонти для створення освітнього простору. Безперечно, це потребує підвищення компетентності майбутніх вчителів щодо основ сучасних е-технологій, зокрема, побудови динамічних інформаційних ресурсів. На нашу думку, до навчальної програми доцільно включити не тільки вивчення основ веб-програмування (мови HTML та PHP), але й водночас основи розробки сайтів та дистанційних курсів засобами інструментальних систем PHP-Nuke та Moodle.

PHP (рекурсивний акронім для “PHP: Hypertext Preprocessor”) – це скриптова мова програмування, яка була створена для генерації HTML-сторінок на стороні веб-серверу. PHP є однією з найпоширеніших мов, які використовуються у сфері веб-розробок (разом із Perl, C#, Python). Мова PHP підтримується переважною більшістю хостерів. PHP інтерпретується веб-сервером в HTML-код, що передається на сторону клієнта. На відміну від таких скриптових мов програмування, як JavaScript, користувач не має доступу до PHP-коду, що є перевагою з точки зору безпеки, але значно погіршує інтерактивність сторінок. Але ніщо не забороняє використовувати PHP для генерування JavaScript-кодів, які виконуватимуться вже на стороні клієнта.

Найкращою якістю цієї мови є те, що вона проста для новачка в програмуванні, але пропонує багато просунутих можливостей для програміста-професіонала. Тому ця мова стала дуже популярною.

Дослідники з pexen.net опублікували результати чергового дослідження (за липень) по використанню мов веб-програмування. Серед таких мов, по даним pexen.net, лідирує PHP з 35,65%, майже у два рази відстає ASP – 17,82%, решта відсотків приходить на всі інші. За статистикою використання PHP, Україна є однією із лідируючих країн у світі, бо 72,37% доменів працює на PHP. У порівнянні з червнем 2005 року, використання PHP в Україні зросло на 2%. Наведемо аргументи щодо зазначеної популярності PHP.

Традиційність. Помітно знижуються початкові зусилля програмістів при вивченні PHP, бо багато конструкцій мови запозичені із C, Perl, а часто-густо код PHP практично неможливо відрізнити від того, що зустрічається в типових програмах C або Pascal.

Простота. PHP-сценарій може складатися навіть з одного рядка, в залежності від специфіки задачі. Механізм PHP просто починає виконувати

код після першої екрануючої послідовності (<?) і продовжує виконання до того місця, коли він зустрине парну екрануючу послідовність (?>), за умови, що код має правильний синтаксис і при цьому немає необхідності у довантаженні бібліотек, вказування спеціальних параметрів компіляції тощо.

Ефективність. У PHP реалізований механізм виділення ресурсів і забезпечена підтримка об'єктно-орієнтованого програмування, засоби керування сеансом, а також механізм підрахунку посилань, що запобігає виділення зайвої пам'яті.

Безпека. PHP надає гнучкі та ефективні засоби безпеки, які умовно поділяються на дві категорії: засоби системного рівня та засоби рівня додатка.

Засоби безпеки системного рівня. У PHP реалізовані механізми безпеки, що керуються адміністраторами; при правильному налагодженні, PHP забезпечує максимальну свободу дій та безпеку. PHP може працювати в безпечному режимі, що обмежує можливості застосування PHP користувачами по ряду важливих показників. Наприклад, можна обмежити максимальний час виконання та використання пам'яті (неконтрольована витрата пам'яті негативно впливає на швидкодію сервера), встановлювати обмеження на каталоги, в яких користувач може переглядати та виконувати PHP-сценарії, а також використовувати їх для перегляду конфіденційної інформації на сервері.

Засоби безпеки рівня додатка. У стандартний набір функцій PHP входить ряд надійних механізмів шифрування, а вихідний текст сценаріїв PHP взагалі неможливо переглянути в браузері, оскільки сценарій компілюється до його відправлення по запиту користувача. Реалізація PHP на стороні сервера запобігає викраденню нетривіальних сценаріїв користувачами, знань яких вистачає для виконання команди View Source.

Гнучкість. Оскільки PHP є мовою, яка вбудовується (embedded), вона відрізняється винятковою гнучкістю відносно потреб розробника: рекомендується використовувати PHP в поєднанні з HTML, однак з аналогічним успіхом вона інтегрується й у JavaScript, WML, XML та інші мови.

PHP не містить коду, орієнтованого на конкретний web-сервер, тому користувачі не обмежуються певними серверами, а можуть використовувати Apache, Microsoft IIS, Netscape Enterprise Server, Stronghold та Zeus.

Безкоштовне поширення. Прийняття стратегії Open Source та безкоштовне поширення вихідних текстів PHP зробило неоціненну послугу користувачам. До того ж, сприйнятливий співтовариство користувачів PHP є свого роду «колективною службою підтримки», і в популярних електронних конференціях можна знайти відповіді навіть на самі складні питання.

Розглянемо процес виконання PHP-сценарію при зверненні браузера до сервера. Спочатку браузер запитує сторінку з розширенням *.php, після чого web-сервер пропускає програму через машину PHP і видає результат у вигляді html-коду. Причому, якщо взяти стандартну сторінку HTML, змінити розширення на *.php і пропустити її через машину PHP, остання просто

перешле її користувачеві без змін, але щоб включити в цей файл команди PHP, необхідно укласти команди PHP у спеціальні теги.

Найсильніші рис мови PHP – це підтримка великої кількості баз даних, таких як MySQL, ODBC, mSQL, PostgreSQL, Oracle тощо, підтримка взаємодії з іншими службами за протоколами LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM тощо. Також слід зазначити, що на мові PHP написано багато інструментальних систем, наприклад PHP-Nuke та Moodle.

PHP-Nuke – це програмне забезпечення, що дозволяє самостійно, без залучення розробників публікувати та змінювати інформацію на сайті. При цьому, для користування такою системою не вимагаються спеціальні знання; достатньо, як правило, навичок роботи лише з текстовим редактором.

Головні особливості PHP-Nuke: адміністрування через спеціальний інтерфейс; керування змістом головної сторінки; збір статистики; можливість налагодження системи для кожного відвідувача окремо, можливість надати право вибору дизайну для зареєстрованих користувачів; адміністративний інтерфейс для редагування статей та коментарів; можливість автоматичного імпортування новин з інших сайтів; підтримка 25 мов; механізм пошуку, подібний Yahoo; можливість коментування при опитуваннях; підтримка легкої зміни дизайну; керування файлами; менеджер частих питань; система оглядів; інформаційний бюлетень; категоризація статей та інші функції.

Слід зазначити, що PHP-Nuke – це безкоштовна система з відкритим кодом, яка написана на PHP, а це дозволяє нам орієнтуватися в коді програми та вдосконалювати під наші певні потреби в майбутньому.

Необхідно також звернути увагу на CMS систему Moodle (Modular Object-Oriented Dynamic Learning Environment – Модульна Об'єктно-Орієнтована Навчальна Система), що є пакетом програмного забезпечення для створення курсів дистанційного навчання та web-сайтів. Система поширюється безкоштовно як Open Source-проект за ліцензією GNU GPL, що дозволяється копіювати, використовувати та змінювати її код. Система може працювати на будь-якому комп'ютері, на якому працює PHP, і підтримує практично будь-яку базу даних (зокрема, MySQL).

Система Moodle спроектована з урахуванням досягнень сучасної педагогіки і придатна як для дистанційного навчання, так і для очного. Вона має простий та зрозумілий web-інтерфейс і створена так, що втручання у роботу адміністратора – мінімальне і при цьому зберігається висока безпеки.

В Moodle можливо переглянути повний звіт із графіками та деталями по діях користувача у системі, тобто останній його вхід, кількість прочитаних статей, повідомлення, записи в зошитах тощо. Існує можливість розсилання електронною поштою новин, форумів, оцінок і коментарів учителів. Підтримуються різні структури курсів: «календарний», «форум», «дерево».

Система Moodle має багатий набір модулів-складових для курсів:

Модуль «Завдання». Дозволяє призначати термін складання, максимальну оцінку та формат відповіді; студенти можуть завантажувати відповіді

на сервер; для кожного завдання можна відвести форум, у якому буде брати участь весь клас тощо.

Модуль «Опитування». Використовується для голосування студентами по певному питанню, або для одержання коментарів від кожного студента.

Модуль «Форум». Доступні різні типи форумів: “тільки для вчителів”, “форум новин”, “відкритий для всіх” тощо.

Модуль «Зошит». Зошит – це приватний контакт між учителем та учнем.

Модуль «Тест». За допомогою цього модулю вчитель може у web-інтерфейсі створити базу даних, яка буде містити питання для багаторазового використання у різних тестах; тести автоматично оцінюються та можуть бути переоцінені при зміні «цінності» питань; тести можуть мати обмежені тимчасові рамки; на вибір вчителя, тести можуть проходитися кілька разів, можуть показувати коментарі до відповідей та/або правильні відповіді; питання можуть містити HTML-текст і малюнки; питання, які припускають вибір з варіантів відповідей можуть мати як одну правильну відповідь, так і декілька; підтримуються питання з відповіддю у вигляді слова або фрази; підтримуються альтернативні питання.

Модуль «Ресурс». Підтримує будь-який контент в електронному виді; можуть використовуватися такі типи ресурсів, як HTML-текст, файл, посилання, зовнішня програма.

Модуль «Анкета». Убудовані анкети COLLES, ATTLS можуть використовуватися для тестування класів; складається онлайнвий звіт по анкетах, що містить графіки; дані можуть бути імпортовані в Excel або у вигляді CSV-файлу; після заповнення анкети студент може отримати усереднені результати загального опитування.

Однієї з основних проблем розвитку освітніх ресурсів є недостатня підготовка вчителів до використання сучасних технологій. У зв'язку з цим, доцільним є впровадження в навчальний процес засобів е-технологій, що реалізуються в ХНПУ імені Г.С. Сковороди для створення динамічного ресурсу “Науковий ФІЗМАТ” та розробки дистанційних курсів.

Література:

1. Новая статистика использования PHP по всему миру. Сайт редакции журнала "PHP Inside". <http://phpinside.ru/?q=node/104>

2. Учебник PHP.
<http://scripts.kiev.ua/ebooks/programming/php/phptutorial/>

3. PHP. Матеріал з Вікіпедії – вільної енциклопедії.
<http://uk.wikipedia.org/wiki/PHP>

4. Sterling Hughes. PHP Developer's Cookbook, Second Edition. – Sams Publishing, 2001. – 504 с.

МЕТОДИКА НАВЧАННЯ КРЕСЛЕННЯ У ПРОФЕСІЙНО-ТЕХНІЧНОМУ НАВЧАЛЬНОМУ ЗАКЛАДІ: ФОРМУВАННЯ ГРАФІЧНИХ ПОНЯТЬ

Н.І. Дорошенко

м. Київ, Інститут педагогіки і психології професійної освіти АПН України
labinf@edu-ua.net

Постановка проблеми. Однією з обов'язкових передумов забезпечення високої функціональності сучасного кваліфікованого робітника будь-якого профілю є належний рівень його графічної підготовки, що передбачає кваліфіковане опрацювання конструкторської і технологічної документації, зокрема, грамотне читання креслень, та точну і безпомилкову натурну реалізацію накресленого на папері. Нинішній рівень графічної підготовки випускників професійно-технічних навчальних закладів (ПТНЗ) є досить низьким і все менше задовольняє вимоги сучасного виробництва. Такий стан пояснюється низькою об'єктивних обставин, зокрема, відсутністю сучасних навчальних програм, відсталістю використовуваної методики формування графічних понять, нестачею якісної навчальної та навчально-методичної літератури, незадоволенням змістового наповнення існуючих підручників та навчально-методичного забезпечення запитів сучасної виробничої практики, педагогічної науки та оновлювально-модернізаційних процесів у суспільстві і освіті.

Креслення становить основу графічної підготовки фахівців технічного чи будівельного спрямування й належить до кола тих навчальних дисциплін, які забезпечують загальноосвітню, загальнотехнічну та професійно-орієнтовану підготовку кваліфікованого робітника і є найбільш значущими для ефективної реалізації цілісного навчально-виховного процесу, кінцевим результатом якого є формування соціально зрілої особистості і компетентного фахівця. Підвищені вимоги до графічної підготовки сучасних робітників потребують удосконалення методики навчання креслення, до змістової частини якої входять графічні поняття.

У тріаді подання узагальненого результату процесу навчання (*знання–уміння–навички*) поняття є складовою знань як продукту пізнання людиною предметів і явищ, діяльності, законів природи і суспільства. Терміни, поняття і категорії разом утворюють вид знань, без яких не можна зрозуміти жодного тексту, жодного висловлення, жодного графічного зображення.

Поняття є основною формою знань, що відображає істотні, необхідні ознаки і відношення предметів і явищ. Зміст поняття становить сукупність істотних ознак, властивостей і відношень предмета думки, відображених у певному понятті. Об'єм поняття визначається розмаїттям предметів, кожному з яких притаманні ознаки, відображені в змісті поняття.

Аналіз останніх досліджень. Методика викладання креслення має

свою історію і свої усталені традиції, характерні для певних періодів розвитку суспільства, техніки, технологій і окремих країн. Серед вітчизняних науковців-педагогів, які останнім часом зробили найбільш помітний внесок до методики викладання креслення, можна виділити Д.А. Тхоржевського, А.П. Верхолу, І.С. Вишнепольського, В.К. Сидоренка, Г.О. Гнітецьку, Г.М. Тропіну, Л.О. Гриценко, М.В. Анісімова, Л.М. Анісімову.

Внаслідок узагальнення результатів їхніх досліджень та вивчення й аналізу педагогічної практики можна констатувати, що графічна підготовка кваліфікованого робітника перш за все передбачає формування графічних понять. З формування графічних понять має розпочинатися навчання креслення й ними ж закінчуватися. В.М. Коротков в одній із своїх наукових праць відмітив, що “формування понять”, це “один із найскладніших аспектів майстерності вчителя, оскільки вимагає знання філософії, логіки, психології, педагогіки. І лише той вчитель, який успішно розв’язує це найскладніше завдання, досягає успіхів у навчанні й вихованні своїх учнів” [7, 61].

Процеси формування понять розкриваються у роботах О.К. Артемова, Д.Н. Богоявленського, О.М. Кабанової-Меллер, З.І. Калмикової, Н.О. Менчинської, П.О. Шеварьова та інших із врахуванням основних положень асоціативно-рефлекторної теорії.

Розробкою теорії поетапного формування розумових дій і понять займалися вчені-психологи О.М. Леонтьєв, П.Я. Гальперін, В.В. Давидов, Д.Б. Ельконін, Н.Ф. Талізін, Л.М. Фрідман. Саме ця теорія, яка бере свій початок ще у працях Л.С. Виготського, виявляється найбільш придатною для формування графічних понять у майбутніх кваліфікованих робітників будівельної галузі.

Поняття не можуть передаватися у готовому вигляді. Засвоєння знань має відбуватися у процесі виконання певної системи дій. У процесі певним чином організованої діяльності поняття засвоюється учнем і може бути використане ним для розв’язання відповідної групи задач. Зазначене підтверджується результатами досліджень Д.О. Тхоржевського [1; 2]. При цьому, акцент робиться на тому, що “під час вивчення загальнотехнічних дисциплін продуктивне формування понять не може ґрунтуватися на механічному їх запам’ятовуванні, а вимагає спеціального управління. Тому в процесі навчання мають створюватися спеціальні ситуації та умови, які сприяють ефективному формуванню в учнів знань та розвитку в них розумової діяльності” [2, 4]. Щодо цього, П.Я. Гальперін уточнював, що “теорія поетапного формування розумових дій і понять має чітко виражений педагогічний аспект, реалізація якого є одним із найбільш перспективних шляхів активізації навчального процесу” [3].

Таким чином, формування графічних понять є одним з основних складників методики навчання креслення, а послідовність і ефективність реалізації цього процесу з урахуванням сучасних вимог щодо необхідного рівня графічної підготовки кваліфікованого робітника, потребує відповідного на-

уково-педагогічного дослідження.

Формулювання цілей статті (постановка завдання). Метою статті є висвітлення підходів щодо послідовності формування в учнів ПТНЗ графічних понять як цілісного дидактичного процесу у структурі і контексті професійної підготовки майбутнього кваліфікованого робітника.

Основна частина. Отже, якість графічної підготовки майбутнього кваліфікованого робітника будівельного профілю визначається рівнем розв'язання дидактичних завдань викладання креслення.

Концептуальною основою дидактичної моделі навчального курсу з креслення у ПТНЗ має бути графічне (геометричне) моделювання, а головною метою навчання креслення, беручи до уваги особливості професійно-технічної освіти та необхідність фундаменталізації її змісту, слід визначити *формування* в учнів *графічних понять*. При цьому, за означенням Н. Чепелевої, під поняттям розумітимемо *логічну форму мислення, що відображає істотні зв'язки, властивості й відношення предметів та явищ у їхній суперечності й розвитку*. А за К. Марксом *“ідеальне є матеріальне, пересаджене у голову людини й перетворене в ній”*. Отже, основна функція поняття полягає у виділенні загального, що досягається абстрагуванням від неістотного.

Зазначене зумовлюється тим, що виробнича діяльність кваліфікованого робітника будівельного профілю полягатиме у точній реалізації задуму архітектора чи проектувальника, що передбачає, насамперед, розуміння накресленого на папері, а не розробку креслень власноруч. Формування і розвиток науково обґрунтованих графічних понять – складна робота, яка поєднує такі компоненти, як аналіз, синтез, узагальнення, які не зводяться до процесів класифікації. Процес засвоєння графічних понять повинен мати циклічний характер і передбачає активну розумову діяльність учня, а власне засвоєння знань має обов'язково поєднуватися з формуванням відповідних умінь і навичок.

Архітектоніка системи графічних понять та глибина їх вивчення визначається вимогами до рівня графічної підготовки кваліфікованого робітника в умовах сучасного виробництва (зокрема, будівництва), завданнями вивчення креслення у ПТНЗ та його змістом. Належний рівень сформованості в учнів графічних понять дозволить успішно розв'язати й вторинні, супутні основній меті навчання цілі, а саме, навчити учнів *прийомам графічних побудов*, технічно грамотно *виконувати прості креслення* деталей і виробів (механізмів), *ескізи і технічні рисунки* відповідно до свого фаху, швидко і безпомилково *читати робочі креслення*.

Процес засвоєння графічних понять передбачає активну розумову діяльність учня. Разом з тим, практичний досвід навчання кресленню свідчить, що засвоєння учнями предметних знань за умови переважно розумової діяльності недостатньою мірою забезпечує необхідний рівень графічної підготовки. У процесі навчання кресленню засвоєння знань має обов'язково по-

еднуватися з формуванням відповідних умінь і навичок. Тобто, процес формування графічних понять має відбуватися в результаті системного поєднання *перцептивної, абстрактно-логічної та чуттєво-практичної* діяльності. Причому, на занятті з креслення найбільше уваги слід приділяти і більшу частку навчального часу відводити власне для чуттєво-практичної діяльності учня як своєрідних засобу, умові, фактору і процесу, де відбувається узагальнення, систематизація, закріплення і опрацювання знань та має місце інтегрований синтез нової навчально-пізнавальної якості – сформованість в учня певної системи графічних понять.

Серед основних результатів графічної підготовки має бути формування і розвиток у молодій людини *“графічного”* мислення, що інтегрує в собі певні аспекти просторового, образного, візуального, модельного, алгоритмічного мислення, здатність до абстрагування, до аналізу і синтезу деякої просторової форми. Графічне мислення включає певні процеси розумової діяльності, серед яких, насамперед, – розпізнавання обрисів об’єктів за їх лінійним окресленням; визначення просторового співвідношення і взаємного розташування об’єктів; уявна реконструкція форми тривимірного об’єкта (зокрема, його уявлення як мисленнєвого тривимірного образу) за певною сукупністю його двовимірних (плоских) зображень; побудова системи плоских зображень (креслення) деякого тіла.

Просторове мислення ґрунтується на просторовому уявленні, спільне формування і розвиток яких відбуваються у процесі навчання графічного та геометричного моделювання об’єктів та власне тривимірного простору. Зокрема, на основі проєкційно-планарного метода Гаспара Монжа (площинна модель тривимірного простору).

Навчальний процес з графічної підготовки майбутнього кваліфікованого робітника, зокрема, будівельного профілю, у ПТНЗ загалом складається з таких інтегрованих етапів.

1. Формування в учнів позитивної мотивації, практичної потреби та стійкого інтересу до вивчення власне креслення (окремих його розділів, тем і питань) – як навчальної дисципліни і як сфери виробничої діяльності людини, – розуміння значення цього предмета, зокрема, його змісту, для успішного опанування професійно-орієнтованих дисциплін та подальшої продуктивної діяльності після закінчення ПТНЗ.

2. Визначення характерних ознак певних графічних понять, вивчення послідовності відповідних графічних побудов, розумова діяльність з реалізації необхідних просторових перетворень;

3. Безпосереднє виконання комплексних графічних побудов.

4. Закріплення й узагальнення сформованих графічних понять і умінь.

На кожному з виділених етапів учні здійснюють визначену навчальну діяльність з засвоєння графічних понять, а характер навчальних дій (як розумових, так і практичних) залежить від сукупності методів і методичних прийомів, які використовує викладач на певному уроці.

Кожний вид графічної діяльності спирається на певну підсистему відповідних понять та, у свою чергу, забезпечує закріплення та узагальнення цих понять. Тому опанування учнями способів розв'язання графічних завдань має відбуватися після первинного вивчення необхідних графічних понять та передувати узагальненню цих графічних понять. Таким чином, важливою передумовою якісної графічної підготовки учнів на уроках креслення є реалізація *циклічно-діяльнісного підходу* до формування графічних понять (через включення формування умінь).

Творча діяльність в сфері техніки і технології потребує графічного вираження у вигляді певних зображень, виконаних згідно з чинними правилами і унормованими домовленостями (стандартами). Сучасні технології розробки графічно-конструкторської документації поєднують як "ручні" методи і засоби графічних побудов, так і комп'ютерні. Це вимагає певної модернізації змісту й методики креслення, зокрема, інтеграції до методичної системи цієї дисципліни та активного застосування у навчальному процесі сучасних апаратних і програмних засобів комп'ютерної графіки та власне певних комп'ютерних графічно-інформаційних технологій. Тому, вже під час викладання матеріалу першої теми викладачеві слід ознайомити учнів з організацією навчальних занять і режимом роботи не тільки в кабінеті креслення, а й у комп'ютерному класі.

Затим, слід представити основні історичні відомості щодо розвитку креслярсько-графічних технологій та, окрім наведення традиційних цитат (висловлювань знаних науковців – Гаспара Монжа, В.І. Курдюмова та ін.), запропонувати учням ознайомитися з графічними роботами найкращих випускників навчального закладу, які виконано як на заняттях креслення, так і під час підготовки дипломної роботи. Доречно також продемонструвати зразки креслень і схем (зокрема, у комп'ютерному варіанті виконання), які використовуються у виробничих процесах на підприємствах та в будівництві. Дати загальну інформацію про графічні (креслярські) редактори та системи автоматизованого проектування, навести приклади виконання вправ, завдань і комплексних графічних робіт на комп'ютері у певному інструментальному програмному середовищі (КОМАС, AutoCAD тощо).

Знайомство учнів з історією розвитку креслення необхідно пов'язувати з появою, розробкою і використанням стандартів на різних етапах розвитку промисловості та будівництва. Необхідно навести приклади перших стандартів, зазначити, що їх накопичення зумовило систематизацію стандартів та зведення їх у Єдину систему конструкторської документації.

Методика пояснення основних правил з оформлення креслень має передбачати реалізацію такої послідовності вивчення графічних понять: лінії креслення; креслярські шрифти; формати для виконання креслень та їх оформлення; масштаби креслення. Учні при цьому розглядають, порівнюють, відтворюють у власній уяві й запам'ятовують образи виділених графічних понять.

Структура процесу формування окремих графічних понять (послідовність етапів) збігається з узагальненою етапністю навчального процесу з графічної підготовки майбутнього кваліфікованого робітника (див. вище). Покажемо зазначене на прикладі вивчення поняття “лінія”.

Під час формування поняття “лінія” на першому етапі (етапі мотивації навчання) демонстровані викладачем прийоми читання креслень мають викликати в учнів інтерес до навчального матеріалу. Спочатку учням пропонується розглянути робоче креслення деталі чи плану будівлі з акцентуванням уваги на використанні ліній різного типу.

Майбутні робітники візуально сприймають зазначене поняття, порівнюють, наводять приклади аналогічних зображень. Надалі постає необхідність демонстрації практичних застосувань різних типів ліній під час розробки конструкторських графічних документів. Тому під час виконання кожного креслення викладач має звертати увагу учнів й пояснювати – який тип лінії, де саме і чому використовується на зображенні. Наприклад, у проєкційному кресленні застосовують тонкі штрихові лінії для зображення невидимих контурів деталей. У процесі вивчення положення січних площин потовщеною розімкнутою лінією. З основними прийомами викреслювання ліній учні ознайомлюються на вступному занятті, а формування вмій і навичок побудови ліній відбуватиметься впродовж усього курсу вивчення технічного креслення.

На другому етапі здійснюється виділення характерних ознак поняття. Тут методичні прийоми навчання теж повинні спрямовуватися на активізацію розумових дій, усвідомлення, уявне перетворення та запам’ятовування виділених графічних понять. При цьому виключно візуальне сприйняття ознаки поняття є недостатнім, оскільки для повного засвоєння графічного поняття необхідно побудувати відповідний графічний образ.

Під час викладання нового матеріалу викладач постановкою запитань спонукає учнів проводити аналіз, синтез, узагальнення, пропонувати власні ідеї, що має сприяти більш глибокому усвідомленому запам’ятовуванню нового поняття. При цьому, виділяється основа графічних дій, створюються передумови для формування понять під час виконання практичних побудов.

Викладачу слід пам’ятати, що найважливішим етапом графічної підготовки є самостійне виконання учнем графічних побудов. Прийоми викладання на цьому етапі повинні включати постановку завдань на викреслювання ліній різних типів відповідно до потреб побудови певних зображень на кресленні. В такому разі засвоєння (закріплення) понять (як складової знань) відбувається через графічні дії. Виконання в подальшому робочих креслень і ескізів деталей та виробів з певної спеціальності потребуватиме застосування всіх відомих типів ліній і таким чином слугуватиме закріпленню отриманих знань та вмій.

На цьому ж етапі викладачеві рекомендується використовувати мето-

дичні прийоми, внаслідок застосування яких відбувається запам'ятовування, тривала фіксація в пам'яті учня послідовності графічних побудов і, як результат безпосереднього виконання побудов, власне зображення образу поняття на полі креслення. Таким чином, прийом спрямовано не стільки на запам'ятовування образу певного зображення, скільки на усвідомлене опанування необхідної послідовності графічних дій, які потрібно виконати під час зображення відповідних ліній (як графічного поняття) на площині.

Наступний етап формування поняття “лінія” – безпосереднє виконання комплексних графічних побудов, у результаті яких отримується цілісне зображення, утворене за допомогою ліній різних типів. Тут слід використовувати прийом проведення графічних робіт. За заданим викладачем завданням учні самостійно виконують побудови без опори на плакат, таблицю чи інші дидактичні засоби унаочнення. Учні вже знають послідовність графічних дій теоретично, проте їхня практична діяльність є ще не досить успішною. Такий прийом направлений на формування поняття під час певної практичної діяльності. Окрім цього, також розвиваються й вдосконалюються власне графічні вміння учнів.

Графічна робота учнів на уроці – один із центральних етапів реалізації не тільки розумової, а й практично-графічної діяльності. Виконання різних вправ, завдань, розв'язання графічних задач варіативного характеру мають спрямовуватися на формування у свідомості і пам'яті учнів стійких образів графічних понять.

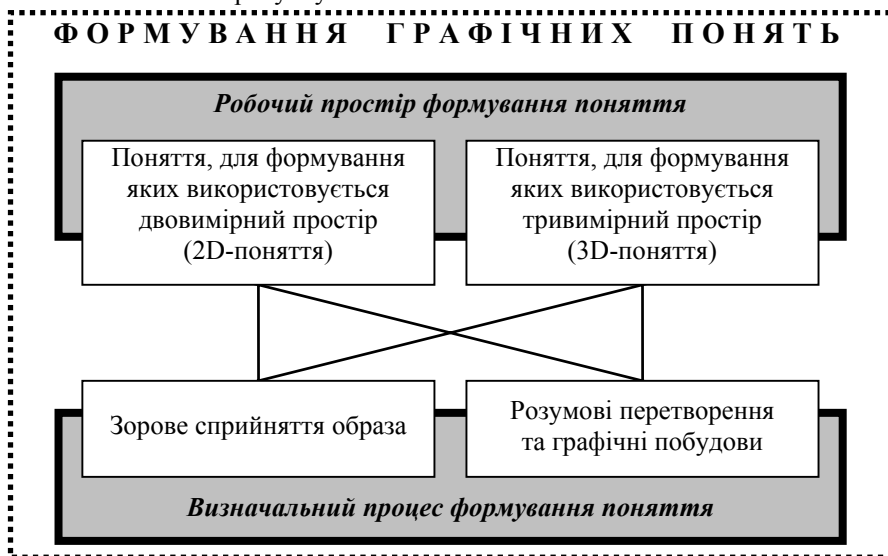
На цьому етапі формування понять “в дії” викладач може використовувати прийом видачі завдань на виконання тренувальних вправ варіативного характеру. Даний прийом повинен забезпечити виконання певних побудов, необхідних для відображення графічного поняття в нових умовах. Цей прийом використовувався й на першому етапі, проте він спрямовувався здебільшого на запам'ятовування графічного образу поняття. Тут же образи графічних понять на кресленні відсутні, їх потрібно пригадати, уявити, а затим побудувати. При цьому, відбувається створення необхідних умов для використання сформованого поняття в нових умовах.

Таким чином, учні в процесі формування понять мають виконувати графічні побудови на кресленні (у двовимірному просторі). Причому, перш ніж отримати закінчене зображення, вони мусять здійснити визначену послідовність графічних дій.

Завершальний етап формування графічних понять передбачає читання (як методичний прийом) незнайомого учневі креслення. Для цього відбираються кілька креслень, які б включали вивчені поняття і містили б якомога менше невивчених елементів. Учні повинні описати процес створення графічного образу поняття “лінія” (вказати всі можливі типи та навести послідовність необхідних побудов) та дати його основні визначення.

Висновки. Архітектоніка системи графічних понять визначається за-

вднаннями вивчення креслення у будівельному ПТНЗ та його змістом. Традиційно виділяють два типи графічних понять: поняття, формування яких пов'язано з розумовими діями і графічним зображенням плоских об'єктів, та поняття, формування яких пов'язано з розумовими діями й графічним зображенням тривимірних об'єктів. Кожний з цих типів графічних понять у свою чергу містить по дві групи графічних понять – поняття, формування яких пов'язано тільки з зоровим сприйняттям образу, і поняття, формування яких відбувається в результаті мислених (уявних) перетворень і графічних побудов. Структурну схему процесу системного формування графічних понять показано на рисунку.



Першочерговими завданнями оновлення методики навчання креслення у будівельному ПТНЗ є визначити і обґрунтувати підходи до виділення графічних понять та дидактичні принципи побудови системи понять, з'ясувати вимоги, які слід урахувати у процесі побудови та власне у структурі, складі й внутрішніх зв'язках моделі системи графічних понять, та на цій основі розробити модель такої системи.

Розроблена й теоретично обґрунтована модель системи графічних понять складе основу змісту навчання креслення та модернізованої методики викладання цієї дисципліни у ПТНЗ будівельного профілю. При цьому, мають застосовуватися сучасні засоби і технології комп'ютерної графіки.

Експериментальна апробація моделі системи графічних понять у складі авторської методики навчання креслення здійснюється на базі Київського вищого професійного училища будівництва і дизайну.

Література:

1. Тхоржевський Д.А. Методика преподавания общетехнических дисциплин и трудового обучения. – К.: Вища шк., 1990. – 104с.
2. Тхоржевський Д.О. Методика трудового та професійного навчання. – Ч. II. Загальні засади методики трудового навчання. – К.: НПУ ім. М.П. Драгоманова, 2000. – 184 с.
3. Управляемое формирование психических процессов/ Под ред. П.Я. Гальперина. – М.: Изд-во Моск. ун-та, 1977. – 198 с.
4. Виготський Л.С. Собрание сочинений: В 6-ти т. – Т. 1. – М.: Педагогика, 1982. – 487 с.
5. Дорошенко Н.І. Шляхи вдосконалення графічної підготовки кваліфікованих робітників у професійно-технічних навчальних закладах// Геометричне та комп'ютерне моделювання. – Вип. 12. – Харків: ХДУХТ, 2005. – С. 37–44.
6. Дорошенко Н.І. Навчання графічної грамоти майбутніх кваліфікованих робітників будівельного профілю із застосуванням інформаційних технологій // “Інформатизація освіти України: стан, проблеми, перспективи”: Матеріали третьої міжнародної конференції. – Херсон: Айлант, 2005. – С. 63.
7. Кротов В.М. Воспитывающее обучение. – М.: Просвещение, 1980. – 192 с.
8. Анісімов М.В., Анісімова Л.М. Креслення. – К.: Вища шк., 1998. – 239 с.
9. Гриценко Л.О. Формування графічних понять в учнів 8–9-х класів на уроках креслення (методичний аспект): Автореф дис. ... канд. пед. наук: 13.00.02. – Полтава, 2004. – 18 с.

РЕДАКТОР ФРАКТАЛОВ И ЕГО ПРИМЕНЕНИЕ НА ЗАНЯТИЯХ ПО ИНФОРМАТИКЕ

А.Л. Столяревская¹, А.Ю. Кузнецов²

¹ г. Харьков, Харьковский национальный педагогический университет
им. Г.С. Сковороды

² г. Харьков, Харьковский национальный университет им. В.Н. Каразина
stolyare@yahoo.com

Одной из наиболее привлекательных тем в разных курсах по информатике для студентов и для школьников является тема «Фракталы». Это обусловлено тем, что фракталы – это объекты завораживающей красоты, а также тем, что модели фракталов имеют широкое применение во многих областях науки, техники и искусства [1]. Появились теория фрактальных трещин, модель трения для фрактальных поверхностей, фрактальная механика древесно-полимерных композитов. Широкое распространение фрактальный подход нашёл в теории динамических систем. Фракталы с большой точностью описывают многие физические явления и природные образования: горы, облака, турбулентные течения, корни, ветви и листья деревьев, кровеносные сосуды.

На занятиях со студентами физико-математического факультета ХНПУ им. Г.С. Сковороды в некоторых курсах по информатике использовалась специальная программа «Редактор фракталов». Меню программы состоит из пунктов «Фрактал», «Изображение», «Текстовый код», «Окно», «Справка».

Подменю пункта «Фрактал» предоставляют возможности: 1) сохранить в специальном формате .fmt или в текстовом формате .txt такие характеристики фрактала как масштаб, начальная точка, точность, глубина, цвет, координаты отрезков и узлов; 2) загрузить фрактал из файла с расширением .fmt с помощью диалогового окна.

Подменю пункта «Изображение» позволяют запустить процедуру рекурсивного построения фрактала в окне «Изображение», а также сохранить построенный фрактал в формате BMP

Подменю пункта «Текстовый код» обеспечивают следующее: 1) загрузить текстовый код фрактала из файла с расширением .txt и сохранить введенные изменения, 2) сформировать код, опираясь на введенные данные в окнах «Установки» и «Фигуры» или на данные, загруженные из файла .fmt, 3) скомпилировать код для обновления данных в вышеупомянутых окнах.

В специальном окне «Фигуры» осуществляется создание эскиза будущего фрактала – с помощью отрезков и/или окружностей, а также узлов. Режим работы с окном включен по умолчанию при загрузке редактора. Чтобы построить отрезок, нужно отметить с помощью мыши его начало и конец. Для задания окружности следует отметить ее центр и любую точку на

окружности. Наиболее важным является режим задания узлов. Для построения из текущего узла задаются коэффициент уменьшения фигуры, угол поворота при построении из текущего узла. При этом можно получить изображения как, например, на рис. 1.

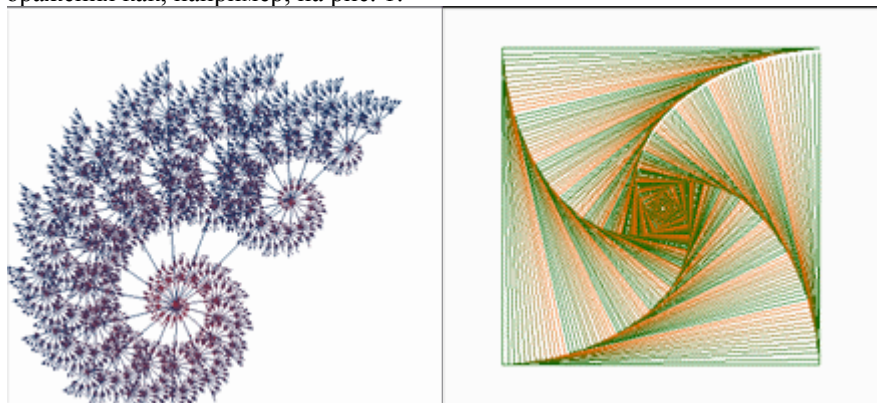


Рис. 1. Примеры фракталов

Сама идея – как создания редактора фракталов, так и использования фракталов в преподавании информатики и математики – не является новой. О том, что тема «Фракталы» является очень популярной, свидетельствует поиск, проведенный в Интернет: по ключевому слову «фрактал» найдено сайтов 11410, документов 141846; по ключевым словам «редактор фракталов» найдено сайтов 268, документов 10850. Остановимся на некоторых из них. Об опыте преподавания темы «Фракталы» в школе рассказывается в работе [2]. В работе [3] указаны темы работ для школьников, которые представляют интерес не только в смысле освоения языка программирования, но и в смысле повышения общей культуры школьников в научном и художественном плане. Это моделирование процесса роста дендритов, построение множества Жюлиа и множества Мандельброта. Непосредственно программы и Java-апплеты для построения фракталов находятся на сайтах [4].

Рассмотрим подробнее, как реализуется построение фракталов на трех примерах – это базовый, пользовательский и более сложный варианты. В качестве базового примера выбран треугольный ковер Серпинского, пользовательского – спираль, более сложного – крест Серпинского.

Треугольный ковер Серпинского имеет вид, показанный на рис. 2.

Алгоритм построения:

1. В окне «Фигуры» (рис. 3) создайте новую фигуру (кнопка левее от надписи «Фигура:»). Справа от надписи 0 сменится на 1.
2. Проверьте, что справа от надписи «Линия/узел» утоплена левая кнопка, т. е. выбран режим «линия».
3. На белом поле 80 x 80, которое находится справа внизу окна «Фигу-

ры» нарисуйте треугольник, отмечая одинарным щелчком левой кнопки мыши начало и конец каждой из сторон.

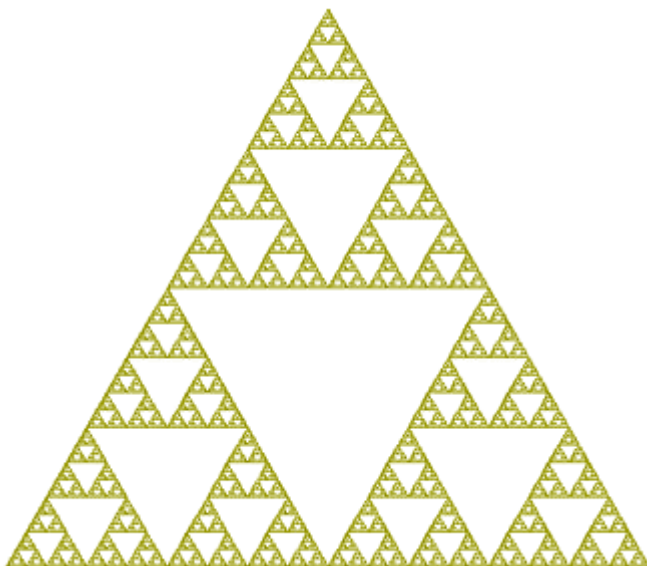


Рис. 2. Треугольник Серпинского



Рис. 3. Окно «Фигуры»

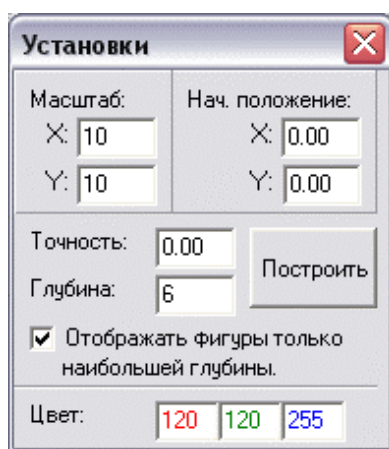


Рис. 4. Окно «Установки»

4. Перейдите в режим «Узел» (справа от надписи «Линия/узел» вторая кнопка).
5. Создайте три новых узла, нажав три раза на кнопки, которая нахо-

дится слева от надписи «Узел:». С помощью стрелок, которые находятся справа, можно переключаться между узлами. По умолчанию узлы имеют следующие параметры: координаты (0,0), коэффициент уменьшения 1, угол поворота 0, фигура 1, цвет (0,0,0).

6. Переключитесь на первый узел и приблизительно отметьте нажатием на левую кнопку мыши на поле, на котором рисовали треугольник, точку, находящуюся на медиане, опущенной из верхней вершины треугольника, и отстоящую от этой вершины на $1/3$ длины медианы.

7. Внесите изменение в параметры узла: Коэфф.: 0.5

8. Аналогично первому узлу, соответствующему верхней вершине треугольника, задайте второй и третий узел, которые соответствуют правой и левой вершинам.

9. В окне «Изображение» фрактал прорисовывается с помощью рекурсивной процедуры. Базовой фигурой является первая. С нее начинается прорисовка. Потом в узлах первой фигуры прорисовываются фигуры, которые соответствуют данным узлам. И так далее, пока глубина рекурсии не достигнет оговоренного максимума. Напротив надписи «Глубина:» выставьте значение 6.

10. Убедитесь, что в окне «Установки» (рис. 4) отмечен флажок «Отображать фигуры только наибольшей глубины». В этом случае на меньшей глубине рекурсии обрабатываются только узлы, а сами фигуры на экран не выводятся.

11. В окне «Установки» нажмите кнопку «Построить».

12. В окне «Установки» можно выставить оптимальный масштаб картинки. Например, параметры (X: 5, Y: 4) соответствуют растяжению изображения по горизонтали в 5 раз и по вертикали в четыре раза. Также можно сдвигать «центр» рисунка – точку, куда помещается центр первой фигуры. В этом же окне можно поменять цвет базовой фигуры. Число, соответствующее красному, зеленому или синему каналу, задается в пределах от 0 до 255.

13. Если Вы не ленитесь проводить расчеты точных значений координат вершин треугольника и узлов, то ввести Ваши численные данные Вы можете при помощи текстового кода фрактала. Выберите в меню «Текстовый код» опцию «Сформировать». В появившемся тексте теперь можно поменять любые параметры фрактала. После того, как Вы внесете все необходимые изменения, выберите в меню «Текстовый код» опцию «Компилировать». Теперь все Ваши изменения учтены. Проверьте параметры узлов, в случае незначительных погрешностей при компиляции (например, не те параметры последнего узла фигуры) исправьте ошибки. В окне «Установки» нажмите «Построить». Получится фигура, показанная на рис. 2.

14. Изображение можно сохранить в формате BMP. Для этого щелкните правой кнопкой мыши на изображении фрактала и выберите опцию всплывающего меню.

Построение спирали

1. Создайте новую фигуру.
2. Нарисуйте квадрат размером во всё белое поле для рисования.
3. Создайте узел. Отмечать его не потребуется, так как нас устраивают координаты по умолчанию – (0,0).
4. Внесите изменения в параметры узла: Коэфф 0.99, Угол 2 (угол измеряется в градусах), Цвет красный (-10).
5. В окне «Установки»:
 - 5.1. Снимите флажок "Отображать фигуры только наибольшей глубины".
 - 5.3. Задайте масштаб (5,5).
 - 5.3. Задайте точность 0.02, глубину 1000.
 - 5.4. Цвет: (255,100,220).
6. Постройте фрактал.
7. Воспользовавшись текстовым кодом, Вы можете откорректировать фигуру – сделать квадрат настоящим квадратом. После компиляции не забудьте проверить, не сбились ли параметры узла. В результате получится изображение, показанное на рис. 5.

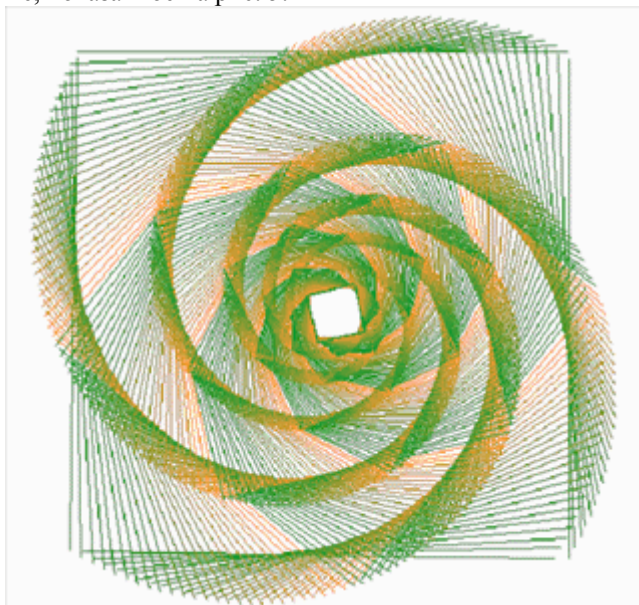
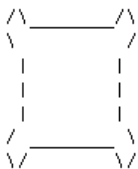


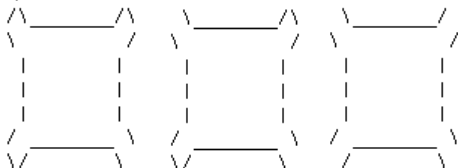
Рис. 5. Спираль

Крест Серпинского

1. Создайте новую фигуру. Нарисуйте примерно следующее:



2. Создайте еще три фигуры. В них рисунки похожи на первый, с некоторыми отличиями:



3. В фигуре 1 создайте 5 узлов: четыре – в вершинах большого квадрата, один – в центре.

4. Во всех пяти узлах поменяйте параметры: Коэфф. везде 0.5, Фигура везде 2, Угол: 0, -90, 180, 90, 0 (для первого узла 0, второго -90 и т.д.).

5. В фигуре 2 создайте такие же пять узлов. Параметры этих узлов отличаются от узлов первой фигуры номерами фигур. Там фигуры 2,2,2,2,2, а здесь задайте 2,2,3,2,4.

6. В фигуре 3 создайте такие же пять узлов. Номера фигур: 3,2,3,2,4.

7. В фигуре 4 создайте один узел – в центре большого квадрата. Номер фигуры: 4. Не забудьте выставить коэффициенты для всех узлов на значение 0.5.

8. Задайте масштаб (4,4), глубину 4.

9. С помощью текстового кода откорректируйте фигуры.

10. При достаточно точных координатах получаем фигуру на рис. 6:

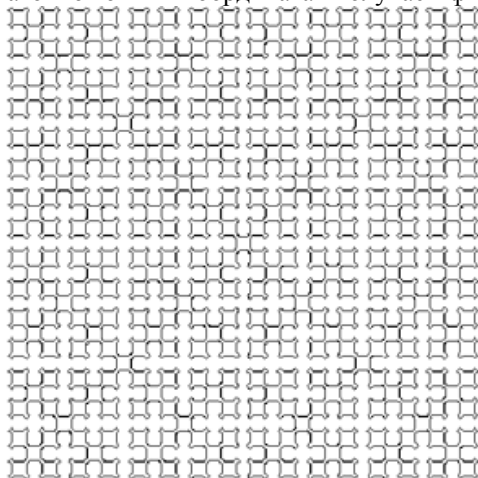


Рис. 6. Крест Серпинского

Точный математический расчет следует применить при выполнении проектирования исходной фигуры для создания определенных фрактальных изображений с помощью редактора.

Помимо редактора фракталов, реализованы программы построения дендритов, фракталов с временным порогом (множества Мандельброта и Жюлиа). Программа Mandel (файл Mandel.exe) позволяет воспроизвести примеры из известной книги [1]. Программа Dendrit (файл Dendrit.exe) обеспечивает построение дендритов.

Соблюдение правил построения фракталов позволяет на уроках информатики выполнять, например, следующие задания.

Задание 1. Использовать программу для построения множеств Жюлиа со следующими параметрами [1]:

$C=0.31+0.04i$	$C=-0.11+0.6557i$
$C=-0.120.74i$	$C=+i$
$C=-0.194+0.6557i$	$C=-0.74543+0.11301i$
$C=-1.25$	$C=-0.481762-0.531657i$
$C=-0.39054-0.58679i$	$C=-0.15862-1.03225i$
$C=0.11031-0.67037i$	$C=0.277334+0.00742i$

При значениях параметра $C=-0.194+0.6557i$ получим рис. 7.

Задание 2. Использовать программу Dendrit.exe для слежения за процессом роста дендрита. Результат работы программы – на рис. 8.

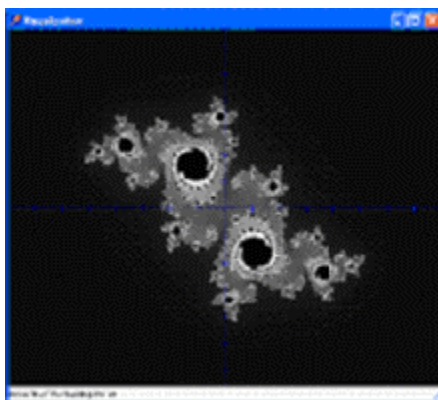


Рис. 7. Множество Жюлиа

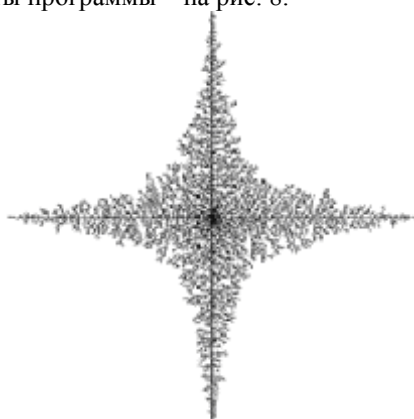


Рис. 8. Дендрит

Задание 3. Построить фрактальную кривую как на рисунке 9 с использованием графического редактора Fractals.exe.

Задание 4. Использовать программу Adobe Photoshop для создания цветных иллюстраций из рисунков, полученных с помощью Fractals.exe. Пример – на рис. 10.

В представленной в настоящей работе программе «Редактор фракталов» используется один из возможных алгоритмов генерирования линейных

фракталов – системы итерируемых функций. Этот алгоритм использует самоподобие, так как фрактал – это структура, состоящая из частей, которые в каком-то смысле подобны целому. С помощью программы создаются как реалистичные изображения природных объектов, так и всем известные фракталы: треугольник и ковер Серпинского, снежинки Кох, кривая Пеано, кривая дракона и другие. Фракталы – это, прежде всего, воплощение точного математического расчета.

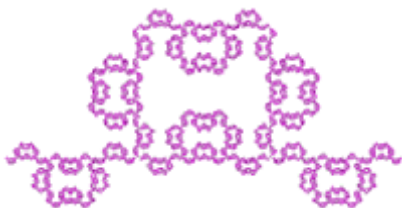


Рис. 9. Ожерелье



Рис. 10. Дерево

Весь комплекс программ используется на занятиях по информатике по теме «Компьютерная графика» и ориентирован на развитие визуального мышления студентов.

Литература:

1. Пайтген Х.-О., Рихтер П.Х. Красота фракталов. – М.: Мир, 1993. – 176 с.
2. <http://ito.edu.ru/index.html>
3. <http://www.museum.ru/museum/cga/education.html>
4. <http://www.nocnit.ru/2st/olimp/web1/index.htm>,
<http://fractals.nsu.ru/fractals.chat.ru/index.htm>,
<http://www.informica.ru/text/infteach/edu/edujava1/mathematics/chaos/>.

ВИВЧЕННЯ ЛОГІЧНИХ ОСНОВ ЕОМ В ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ

Л.О. Лісіна, І.С. Мінтій

м. Кривий Ріг, Криворізький державний педагогічний університет

Практика роботи зі школярами та студентами перших курсів показала, що більшість з них не мають цілісної картини щодо логічних основ роботи комп'ютера. Якщо більшість шкільних підручників [1; 2; 3; 4] дають поняття про системи числення та опрацювання двійкових чисел, то тільки [1; 3; 5] звертають увагу на те, що комп'ютер працює не з самими двійковими числами, а з їх кодами (так звана доповнювальна арифметика). Окрім числових даних в інформатиці існують і інші типи даних, наприклад, логічні.

Логічні дані та дії, визначені над ними, утворюють булеву алгебру, названу на честь її творця – англійського математика Джорджа Буля. Уже значно пізніше виявилось, що функціонування багатьох електронних пристроїв (лічильників, суматорів, шифраторів, дешифраторів, тригерів тощо) описується формулами булевої алгебри [1; 5; 7].

Логічні формули застосовують для побудови (синтезу) автоматів. Конструювання (синтез) автоматів – одна із основних задач кібернетики.

Знайомство з загальними принципами побудови автоматів ми розпочинаємо з прикладу. Його узагальнення дасть можливість сформулювати алгоритм синтезу найпростіших автоматів в цілому.

Приклад. Для оповіщення глядачів, котрі спостерігають за змаганнями з важкої атлетики, виготовлено світловий транспарант зі словами: “Вагу взято вірно”. Підсвічування транспаранту здійснюється за командою, що її видає автомат. Цей автомат обробляє сигнали, що надходять від трьох суддів: A , B і C . Суддя A – головний. Якщо суддя вважає, що вагу взято вірно, то він натискає кнопку, розміщену на його столі, подаючи тим самим в автомат сигнал “1”. Сигнал на підсвічування транспаранту подається тоді, коли натиснуть кнопки всі троє суддів, або двоє, один з яких – головний. Необхідно накреслити відповідну функціональну схему.

Опис задачі, яку повинен розв'язати автомат у наведеній вище формі, прийнято називати словесною формою подання автомату.

Автомат виступає при цьому як деяка “чорна скриня”: ще невідомо, як майбутній автомат буде побудований всередині, але вже зрозуміло, як він буде працювати, відомо, як він пов'язаний із зовнішнім середовищем.

В даному випадку автомат має три входи – це лінії, по яких надходять сигнали A , B і C та один вихід Y (сигнал для підсвічування транспаранту).

Складемо таблицю роботи автомату:

A	B	C	Y
1	1	1	1
1	1	0	1

A	B	C	Y
1	0	1	1
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

Маючи цю таблицю, можна записати структурну формулу автомату:

$$Y = ABC + \overline{A}\overline{B}C + A\overline{B}\overline{C} = AB + AC = A(B + C).$$

Функціональна схема буде мати вигляд (рис. 1).

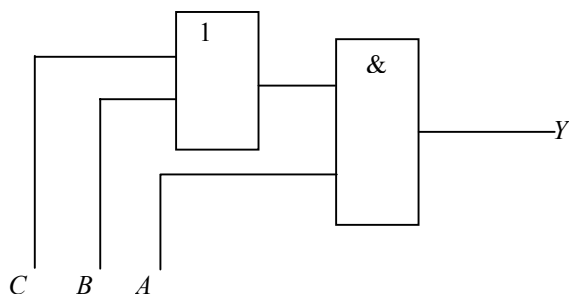


Рис. 1

Таблиця істинності – це табличний спосіб задання логічної функції.

Аналітичний спосіб – це запис логічної функції рівняннями, які виводять з таблиці істинності.

Існують дві форми запису в аналітичному вигляді: ДДНФ (досконала диз'юнктивна нормальна форма) і ДКНФ (досконала кон'юнктивна нормальна форма).

Для запису ДДНФ необхідно:

- 1) вибрати набори аргументів, при яких функція дорівнює 1;
- 2) записати формулу логічного множення всіх аргументів стільки раз, скільки таких наборів існує і об'єднати їх знаками логічної суми;
- 3) поставити знак інверсії над аргументами у вибраних наборах, які дорівнюють нулю.

Для запису ДКНФ необхідно:

- 1) взяти набори, де $Y = 0$;
- 2) записати формулу логічної суми всіх аргументів, всі суми об'єднати знаком логічного множення;
- 3) над аргументами, які дорівнюють 1, поставити знак інверсії.

Використовуючи закони логіки, можна довести, що обидві форми запису рівнозначні. Якщо в значеннях функції більше одиниць, то беремо ДДНФ, якщо більше нулів, то ДКНФ.

Підведемо підсумки роботи по створенню автомату у вигляді наступного алгоритму:

1. Задати алгоритм словесно.
2. Визначити кількість входів та виходів автомату.
3. Скласти таблицю роботи автомату.
4. Використовуючи таблицю, вписати структурну формулу автомату.
5. Накреслити функціональну схему автомату.

Узагальнюючи, можна сказати, що побудований в нашому прикладі автомат реалізує функцію $Y = F(X)$, де X – конкретне слово на вході, а Y – відповідне слово на виході.

Такі автомати називаються одноктактними автоматами або комбінаційними схемами. З таких одноктактних автоматів складаються складні автомати, з яких, у свою чергу, складаються обчислювальні машини.

Розглянемо ще один одноктактний автомат, який називається напівсуматором.

Напівсуматор – автомат, що забезпечує додавання двох двійкових однорозрядних чисел без урахування розряду переносу.

Подамо напівсуматор як одноктактний автомат:

а) складемо таблицю:

A	B	P	S
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	0

б) запишемо ДДНФ:

$$S = A\bar{B} + \bar{A}B \quad \text{для виходу } S,$$

$$P = AB \quad \text{для виходу } P.$$

Спростимо перший вираз:

$$S = A\bar{B} + \underbrace{A\bar{A}}_0 + \bar{A}B + \underbrace{B\bar{B}}_0 =$$

$$A(\bar{A} + \bar{B}) + B(\bar{A} + \bar{B}) = (\bar{A} + \bar{B})(A + B) = (A + B) \cdot \overline{A \cdot B}$$

в) накреслимо функціональну схему (рис. 2).

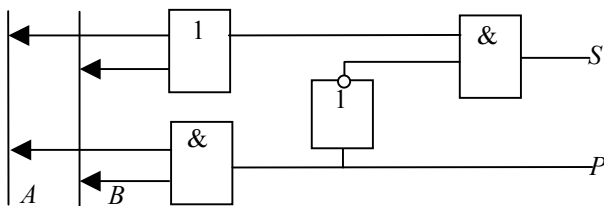


Рис. 2

Далі ми розглядаємо суматор як автомат, який складає два однорозряд-

них двійкових числа з урахуванням розряду переносу.

Суматор задається таблицею:

<i>A</i>	<i>B</i>	<i>C</i>	<i>S</i>	<i>P</i>
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

Умовно напівсуматор позначається так:

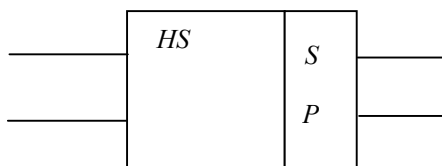


Рис. 3

Маючи таблицю, виписуємо дві формули – одну для виходу *S*, іншу – для виходу *P*.

$$S = ABC + \overline{A}BC + A\overline{B}C + \overline{A}\overline{B}C \quad (1)$$

$$P = ABC + \overline{A}BC + A\overline{B}C + \overline{A}\overline{B}C \quad (2)$$

Використовуючи задачу про тотожне перетворення, спростимо формулу (1):

$$S = AB + AC + BC \quad (3)$$

Структурні формули (1) та (2) запишемо так:

$$P = AB + AC + BC$$

$$S = (A + B + C)\overline{P} + ABC$$

Спрощена функціональна схема подана на рис. 4. Така схема використовується на практиці і має позначення (рис. 5).

Підкреслимо, що в будь-який момент часу кожен автомат розглядає “слово”, складене з “букв”, що подаються на кожен його вхід; кожному слову на вході відповідає єдине слово на виході.

Учням пропонується скласти схему суматора на базі напівсуматорів. Ця робота виконується самостійно і через 10-15 хвилин обговорюється її результат. Відповідна схема наведена на рис. 6.

Далі розглядаємо моделювання пам’яті в кібернетиці [4]. У вище згаданих автоматах інформація подавалась у вигляді слів двійкового алфавіту. Однотактні автомати “вміють” добре розрізняти такі слова, а тепер необхідно навчити їх запам’ятовувати. Перш, ніж автомат навчиться за-

пам'ятовувати, згадувати (і забувати) слова, його треба навчити запам'ятовувати окремі букви двійкового алфавіту.

Пристрій, який може запам'ятовувати букви "0" і "1", демонструвати їх, а у випадку необхідності і забувати, називається *тригером*.

Ми знайомимо учнів і студентів з так званим тригером з парним входом.

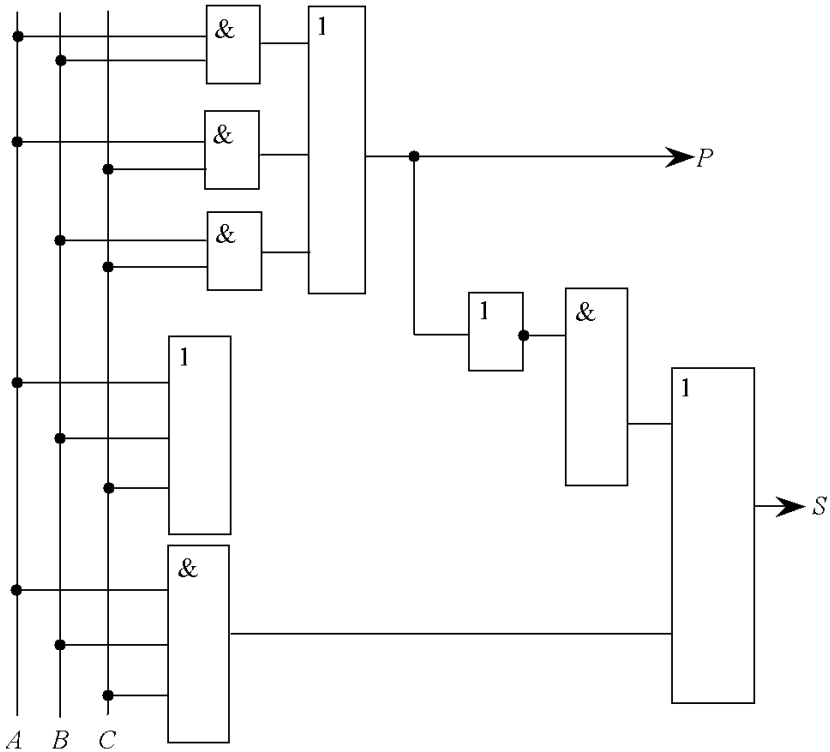


Рис. 4

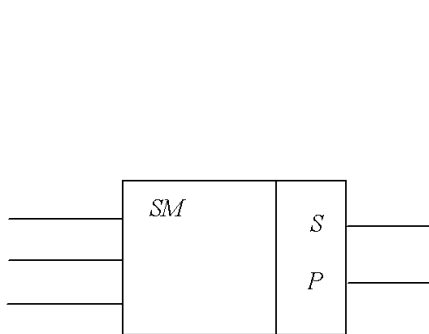


Рис. 5

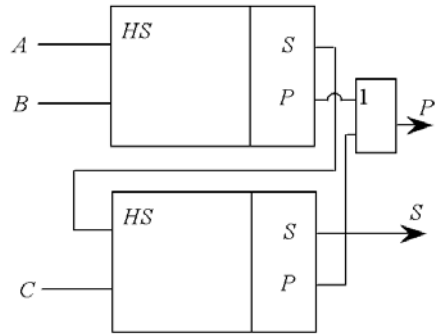


Рис. 6

Умовне позначення тригера показано на рис. 7.

Такий тригер має один вхід і два виходи. В будь-який момент часу на його виходах демонструються сигнали 0 і 1, причому якщо $X_1=1$, то $X_2=0$, і навпаки.

Як керувати тригером?

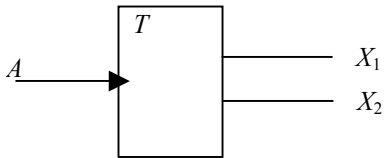


Рис. 7

Якщо тепер на вхід подати на короткий час (на долі секунди) сигнал 1, то на виходах X_1 і X_2 сигнали зміняться на протилежні, тепер $X_1=0$, а $X_2=1$. Ці сигнали будуть демонструватись до тих пір, поки на вхід A знову не буде подано сигнал 1. Кожна короткочасна подача сигналу 1 на вхід A “переключає тригер”, викликає зміну сигналів на його виходах. При необхідності сигнали, що знімаються з виходів X_1 і X_2 , можна передавати по провідниках і використовувати в інших схемах.

Декілька тригерів можна об'єднати в групи – так звані регістри.

Розглянемо регістр з трьох тригерів (рис. 8).

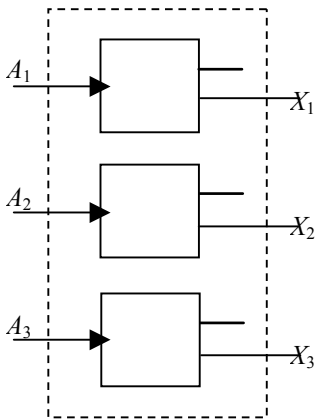


Рис. 8

Він може служити для запам'ятовування та демонстрування двійкових слів, що складаються з трьох букв, і може розглядатись як один із варіантів найпростішого запам'ятовуючого пристрою – ЗП.

Нехай в деякий момент часу $X_1=X_2=X_3=0$, це означає, що ЗП пам'ятає і демонструє слово “000”.

Подано 1 на вхід A_1 , це викличе зміну сигналу X_1 . Слово в пам'яті з “000” зміниться на “001”.

Якщо тепер подати сигнал на вхід A_3 , то зміниться X_3 . На виході буде демонструватись слово “101”.

Підкреслимо, що якщо в регістр входить n тригерів, то на такому ЗП можна організувати запам'ятовування та демонстрацію $2n$ двійкових слів.

Це не єдиний спосіб моделювання пам'яті.

Таким чином, учні та студенти отримують поняття не тільки про логічні елементи, але й про елементарні автомати, які складаються з цих елементів.

Література:

1. Глинський Я.М. Інформатика: 8–11 класи: Навч. посіб. для загальноосвіт. навч. закл.: У 2-х ч. – Ч. 2. Інформаційні технології. 4-ге вид. – Львів: Деол, СПД Глинський, 2004. – 264 с.
2. Гончаренко С.У., Хаїмзон І.І. Учням про цифрову електроніку. – К., Рад. шк., 1981.
3. Есипов А.С. Информатика. Учебник по базовому курсу общеобразовательных учебных заведений. Изд. 2-е, доп. и перераб. – С.-Пб.: Наука и техника, 2001. – 384 с.
4. Зарецька І.Т., Колодяжний Б.Г. Інформатика: Навч. посібник для 10 – 11 кл. загальноосвітн. шк. – Х.:Факт; К.: Гала, 1998. – 384 с.
5. Касаткин В.Н. Введение в кибернетику: Пособие для факультатив. занятий в 9 кл. – 3-е изд., перераб. и доп. – К., Рад. шк., 1986. – 176 с.
6. Руденко В.Д., Макарчук А.М., Патланжоглу М.А. Курс информатики / Под ред. Мадзигона В.Н. – К.: Феникс, 2001. – 370 с.
7. Токхейм Р. Основы цифровой электроники. – М., Мир, 1988.

СПРОЩЕННЯ ЛОГІЧНИХ ВИРАЗІВ ЗА ДОПОМОГОЮ КАРТ КАРНО В ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ

Л.О. Лісіна, І.С. Мінтій

м. Кривий Ріг, Криворізький державний педагогічний університет

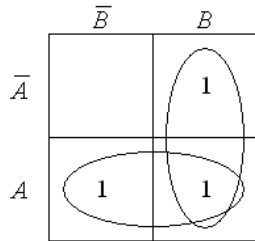
При конструюванні автоматів виникає необхідність для зменшення кількості елементів функціональної схеми спростити логічний вираз автомату. Оскільки на цей момент школярі та студенти перших курсів алгеброю логіки ще не володіють, то ми пропонуємо спростувати логічні вирази за допомогою карт Карно. Карта Карно – це таблиця, кількість комірок у якій дорівнює 2^n , де n – кількість висловлювань, що входять до логічного виразу. Назви комірок проставляються на зовнішній стороні таблиці певним чином, змінювати який не можна. Кожна складова логічного виразу проставляється одиницею на перетині відповідних комірок. Далі всі сусідні одиниці об'єднуються групами по дві, чотири або вісім одиниць. Контури будуються до тих пір, поки всі одиниці не опиняться всередині контуру. Якщо у контурі є логічне висловлювання і його заперечення, то вони взаємно знищуються. Кожен контур представляє собою новий член спрощеного логічного виразу [7].

Розглядаємо такі випадки:

1. Два входи (два висловлювання або дві змінних). Структурна формула, наприклад, має такий вигляд:

$$\overline{A}B + A\overline{B} + AB = y.$$

Побудуємо карту Карно за правилами, які ми розглянули раніше.



Об'єднаємо сусідні одиниці в контури (по дві).

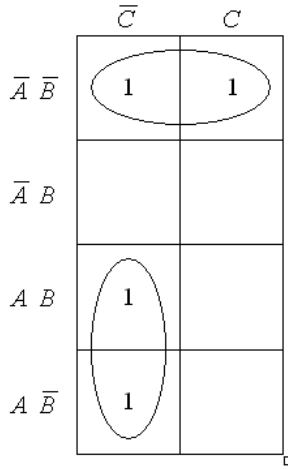
У результаті спрощення отримаємо

$$A + B = y.$$

2. Три входи (три змінних).

2.1. Розглянемо вираз:

$$\overline{A}BC + \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + A\overline{B}\overline{C} = y.$$



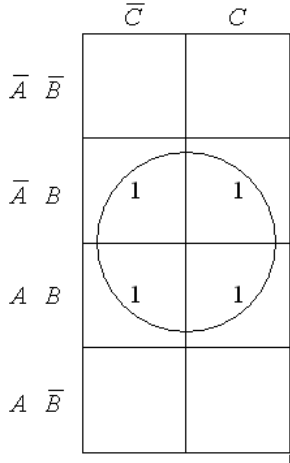
Розглянемо отримані два контури:

B, \bar{B} доповнюють одна одну, їх можна не писати, C, \bar{C} також. Тому в результаті спрощення:

$$A\bar{C} + \bar{A}B = y$$

2.2. Розглянемо вираз:

$$\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC = y$$



В цьому прикладі чітко вимальовується один контур на чотири одиниці, тому в результаті спрощення маємо:

$$B = y$$

3. Чотири входи (чотири змінні).

Розглянемо вираз:

$$\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}BC\overline{D} + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} = y$$

Намалюємо карту Карно і визначимо на ній відповідні контури.

	$\overline{C}\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\overline{B}$		1	1	
$\overline{A}B$		1	1	
AB				
$A\overline{B}$	1	1		

В результаті спрощення отримаємо такий вираз:

$$\overline{A}\overline{B}\overline{C} + \overline{A}D = y$$

4. Інші різновиди карт Карно.

4.1. Розглянемо вираз:

$$A\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + A\overline{B}C\overline{D} = y$$

Нанесемо одиниці у відповідні клітинки карти Карно.

	$\overline{C}\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\overline{B}$				
$\overline{A}B$	1			1
AB	1			1
$A\overline{B}$				

Таку карту можна “скрутити” у вертикальну трубочку і розглядати один контур із чотирьох одиниць. В результаті отримуємо:

$$B\overline{D} = y$$

4.2. Розглянемо вираз:

$$\overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD = y.$$

	$\overline{C}\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\overline{B}$	1	1		
$\overline{A}B$				
AB				
$A\overline{B}$	1	1		

Таку карту можна “скрутити” у горизонтальну трубочку і розглядати один контур із чотирьох одиниць. В результаті спрощення отримуємо:

$$\overline{B}\overline{C} = y$$

4.3. Розглянемо такий вираз:

$$\overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD = y$$

	$\overline{C}\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\overline{B}$	1			1
$\overline{A}B$				
AB				
$A\overline{B}$	1			1

Таку табличку можна “скрутити” у кульку, і знову отримуємо контур із чотирьох одиниць. В результаті спрощення:

$$\overline{BD} = y$$

Розглянувши всі можливі випадки, можемо записати алгоритм спрощення логічних виразів за допомогою карт Карно:

1. Записати булевий вираз в ДДНФ.
2. Нанести одиниці на карту Карно.
3. Об'єднати всі сусідні одиниці контурами по дві, чотири, вісім одиниць.
4. Провести спрощення, виключити всі члени, що доповнюють один одного всередині контуру.
5. Об'єднати члени виразу (по одному в кожному контурі) логічною функцією «або».
6. Записати спрощений вираз в ДНФ.

Як показала практика роботи зі школярами та студентами першого курсу запропоновані схеми спрощення логічних виразів зрозумілі і доступні для використання.

Література:

1. Глинський Я.М. Інформатика: 8–11 класи: Навч. посіб. для загальноосвіт. навч. закл.: У 2-х ч. – Ч. 2. Інформаційні технології. 4-те вид. – Львів: Деол, СПД Глинський, 2004. – 264 с.
2. Гончаренко С.У., Хаїмзон І.І. Учням про цифрову електроніку. – К., Рад. шк., 1981.
3. Зарецька І.Т., Колодяжний Б.Г. Інформатика: Навч. посібник для 10–11 кл. загальноосвітн. шк. – Х.: Факт; К.: Гала, 1998. – 384 с.
4. Касаткин В.Н. Введение в кибернетику. – К., Рад. шк., 1986. – 176 с.
5. Касаткин В.Н., Верлань А.Ф., Переход И.А. Элементы кибернетики – школьнику. (Сборник упражнений и задач). – К., «Радянська школа», 1974. – 112 с.
6. Руденко В.Д., Макачук А.М., Патланжоглу М.А. Курс информатики / Под ред. Мадзигона В.Н. – К.: Феникс, 2001. – 370 с.
7. Токсейм Р. Основы цифровой электроники. – М., Мир, 1988.

ИСПОЛЬЗОВАНИЕ НОВЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В КУРСЕ «ОСНОВЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»

А.Л. Столяревская

г. Харьков, Харьковский национальный педагогический университет
им. Г.С. Сковороды
stolyare@yahoo.com

Искусственный интеллект является подобластью информатики. Еще в 80-е годы XX столетия академик А.П. Ершов определил информатику как самостоятельную науку, которая «вступает в права тогда, когда в рамках соответствующей частной науки строится информационная модель того или иного фрагмента действительности...» [2] или, более широко [1]: «Мы трактуем информатику как становящуюся фундаментальную естественную дисциплину, изучающую законы и методы накопления, обработки и передачи информации в природных, технических и социальных системах. Эта наука опирается на философское учение об информации в процессах отражения и состоит из таких конкретных областей как теоретические основы вычислительной и коммуникационной техники, алгоритмика, программирование, искусственный интеллект, теория когнитивных процессов, включая вычислительный эксперимент, информологию, или учение об информационных процессах в обществе». А.М. Довгялло, Г.С. Поспелов, Д.А. Поспелов [3–6], другие авторы рассматривали искусственный интеллект как новую информационную технологию, в своих работах они указывали на целесообразность применения идей и методов искусственного интеллекта в обучении. Зарубежными специалистами А. Барром и Э. Фейгенбаумом было дано определение [7]: «Искусственный интеллект – это область информатики, которая занимается разработкой интеллектуальных компьютерных систем, т.е. систем, обладающих возможностями, которые мы традиционно связываем с человеческим разумом, – понимание языка, обучение, способность рассуждать, решать проблемы...». Целый ряд работ психологов содержат описательные (вербальные, графические) модели интеллекта. Они изображаются символами, поэтому их можно выразить на языке алгоритмов для компьютера. Интеллект можно определить как «систему, познающую окружающий мир», «основным атрибутом интеллекта является способность производить правильный выбор» [8], т.е. деятельность. Психологи утверждают о «зависимости деятельности от «установки» и «факторов» и о «формировании факторов для достижения определенного вида деятельности» [9]. Это непосредственно связано с обучением.

Проблемой настоящей работой является формирование факторов для достижения следующей цели – научить студентов принципам общения с компьютером на естественном языке, принципам создания игровых и обучающих программ. **Цель** работы – определить совокупность знаний, умений

и навыков при обучении магистров (направления «Информатика» в педагогическом университете) дисциплине «Основы искусственного интеллекта».

Большинство современных курсов по информатике имеют динамично изменяющееся наполнение. К их числу принадлежит курс «Основы искусственного интеллекта». Базой этого курса, несомненно, являются важные разработки середины прошлого столетия [10]. Именно в то время идеи об использовании искусственных механизмов, вынашиваемые в течение веков, впервые смогли найти свое реальное воплощение в программах для компьютера. Последние несколько лет внесли свои коррективы в содержание курса по основам искусственного интеллекта, так как акцент в разработках интеллектуальных систем сместился на их практическое использование [10]. Эти изменения отразились на подготовке курса «Основы искусственного интеллекта» для магистров по информатике.

Курс, разработанный ранее для специалистов в области информатики [11; 12], был ориентирован на создание программ с помощью языка логического программирования Пролог в областях общения с компьютером на естественном языке, создания игровых и обучающих программ, разработки экспертных систем.

В силу небольшого объема курса для магистров (18 часов лекций и 18 часов лабораторных занятий) пришлось заменить изучение языка логического программирования Пролог демонстрацией его основных возможностей на примерах программ распознавания структуры предложения естественного языка (синтаксический анализ предложения), экспертных систем (известные программы «Угадай животное», «Расследование преступления»). В курсе для магистров акцент перенесен на логический подход к построению систем искусственного интеллекта: логика как формальная модель для представления данных и знаний [13], логический вывод, логические игры [14]. Составление программ на языке Пролог заменено созданием программ с элементами искусственного интеллекта на языках Паскаль, Delphi и JavaScript [15]. Это обусловлено отчасти тем, что в группу магистров вошли студенты с разным уровнем знания языков программирования в силу специфики предыдущего обучения в двух разных группах в течение четырех лет. Базовым курсом программирования для всех студентов являлся язык программирования Паскаль. Потом изучались языки Delphi (или Visual Basic), HTML. Только в одной из групп студенты в курсе «Информационные и коммуникационные системы» использовали язык JavaScript при создании интерфейсных элементов для информационных систем. Поэтому целесообразным для двух категорий студентов в течение первого семестра обучения в магистратуре было повторение основ языка HTML, создание программ-сценариев для таких задач как составление анкеты, составление нагрузки преподавателя [15].

Курс «Основы искусственного интеллекта» для магистров начался с рассмотрения задачи обработки строк на JavaScript, так как именно сим-

вольное представление является основным для задач искусственного интеллекта. Предлагались следующие задачи символической обработки строк: 1) выяснить, является ли заданная последовательность символов идентификатором; 2) найти правильную расстановку знаков арифметических операций в выражении $((((1?2)?3)?4)?5)?6=35$ (использовать полный перебор); 3) проанализировать выражения, записанные в префиксной и постфиксной нотациях; построить синтаксический анализатор для разбора предложений, описанных простой грамматикой; оценить арифметическое выражение в инфиксной нотации; оценить логическое выражение, (применить метод рекурсивного спуска); 4) реализовать логический вывод методом резолюций; осуществить другие методы логического вывода.

При этом закреплялись навыки, полученные ранее в области составления эффективных алгоритмов, при изучении темы «Рекурсия». Важным моментом было приобретение знаний и навыков в области обработки естественного (или ограниченного естественного) языка, так как именно естественный язык является базой для построения естественного интеллекта.

Далее в учебном курсе внимание было уделено такой перспективной интеллектуальной технологии, как применение программ-агентов для реализации пользовательского интерфейса на естественном языке.

С развитием компьютерных технологий произошло изменение способов общения пользователя с компьютером, разработаны новые технологии интерфейса, а также системы понятий-метафор, которые отображают процессы, происходящие в компьютере, в понятные обыкновенному пользователю абстракции, такие как, например, документ.

В настоящее время наибольшее распространение получил графический интерфейс пользователя, который базируется на метафорах рабочего стола, меню, программы, папки, панели инструментов. Этот вид интеллектуально-графического интерфейса позволил получить доступ к компьютеру новым категориям пользователей, которые эффективно его используют, не вдаваясь в тонкости внутреннего устройства компьютера и происходящих в нем процессов. Подобному интерфейсу присущи недостатки: 1) графический интерфейс не так интуитивен, как рекламируется, так как для его использования необходимо некоторое начальное обучение, а также понимание системы метафор и возможностей компьютера; 2) графический интерфейс по большей части пассивен, т.е. пользователю необходимо постоянно указывать компьютеру, что делать. Обратный поток информации содержит в себе практически только подтверждения сделанных действий, так что использование такого интерфейса нельзя назвать общением с компьютером, это скорее последовательность приказаний и откликов; 3) этот интерфейс не позволяет автоматизировать рутинные и повторяющиеся операции.

Метафора программ-агентов пришла на смену метафоре рабочего стола. Это новая технология. При ее использовании пользователь, помимо работы с объектами-метафорами созданной компьютером виртуальной реаль-

ности, также общается с субъектами-агентами. Агент является антропоморфным: он похож по поведению и, возможно, по внешнему виду на человека или другое живое существо [16]. Агент персонализирован: он обладает определенными личностными характеристиками. Общению с агентом практически не нужно учиться. При работе с агентами применяются навыки, используемые при каждодневном общении.

Такой интерфейс является действительно интуитивным. Он активен, поскольку программы-агенты, помимо исполнения приказаний, могут брать на себя выполнение рутинных или не требующих присутствия пользователя операций, отчитываться в их исполнении, накапливать информацию, обучаться, предугадывать потребности и желания пользователя, т.е. вести себя как умные и доброжелательные помощники.

Программные агенты могут использоваться: 1) в роли партнера в играх; 2) в роли тьютора при интерактивном обучении и тренировке; 3) в роли подсказчика (как анимированные персонажи в Microsoft Office); 4) в роли секретаря или помощника; 5) в роли референта при выполнении поиска, отбора и представления информации.

Технологическая основа построения агентских интерфейсов состоит из двух составляющих: интерфейса на естественном языке и искусственного интеллекта, обрабатывающего и накапливающего знания. Первая составляющая включает в себя технологии анализа естественного языка, которые занимаются преобразованием текста на естественном языке в некое формализованное представление, пригодное для последующей обработки. Семантический и последующие этапы анализа естественного языка выдают в качестве результата семантическую структуру текста, то есть его смысловое содержание. Эти структуры затем используются компонентой искусственного интеллекта, преобразуются ею в факты и знания и используются для рассуждений, выводов и объяснений.

В лекции для магистров включены вопросы «Технология MS Agent», «Создание анимированных персонажей», в лабораторные работы – темы «Интерактивный помощник – Assistant», «Использование MS Agent».

В MS Office уже давно используются помощники (или ассистенты), которые представляют собой усеченный вариант анимированных персонажей. Помощники не могут говорить и воспринимать голосовые команды. Нельзя одновременно использовать несколько персонажей. Следующий шаг в развитии помощников – это агент MS Agent. Он является элементом ActiveX, поэтому его можно использовать на HTML-страничках при помощи языков JavaScript, VBScript. Для включения MS Agent в состав веб-страницы используется тег <OBJECT> в области заголовка HEAD или в теле документа BODY следующим образом

```
<Object ID="Agent"  
  ClassID="CLSID:D45FD31B-5C6E-11D1-9EC1-00C04FD7081F"  
  CodeBase="#VERSION=2,0,0,0">  
</Object>
```

Если ядро MS Agent не установлено на компьютере, оно автоматически загружается с веб-узла Microsoft. Это стандартное поведение ActiveX-компонента. После того, как сам компонент загружен, нужно указать имя файла, содержащего персонаж. Это может быть файл на локальном компьютере или URL-адрес файла с расширением *acf*. Формат ACF специально разработан для загрузки персонажей через Интернет.

На компьютере файлы персонажей с расширением *acs* находятся в каталоге *C:\WINDOWS\msagent\chars*, персонажами управляет приложение *agentsvr.exe*, которое расположено в каталоге *C:\WINDOWS\msagent* вместе с необходимыми библиотеками *dll*.

Для вызова персонажа достаточно указать имя файла с расширением *acs* при вызове метода *Load*. Это реализуется, например, так:

```
Agent.Characters.Load ("Merlin", "merlin.acs");
```

или с помощью процедуры

```
MyAgent.Characters.Load "MyAgent", "merlin.acs"
```

Если персонаж находится не в папке *C:\WINDOWS\msagent\chars*, то нужно указывать полный путь к файлу.

После загрузки персонажа можно использовать различные методы для его анимации. Сначала используется метод *Show*, который выводит персонаж на экран. После этого можно использовать метод *Play*, указывая точное имя анимации, закрепленное за этим действием. Так, метод *Speak* позволяет персонажу говорить. Для создания персонажей используется инструмент Microsoft Agent Character Editor (ACE), доступный по адресу [17]. Редактор персонажей ACE способен собрать серию анимаций в один файл для дальнейшего использования в MS Agent. Для этого нужно создать предварительно серию рисунков, установить настройки и продолжительности анимации, переходов, включить, если нужно, звуковые и речевые эффекты.

Таким образом, в курсе «Основы искусственного интеллекта» для магистров в области информатики изучаются интеллектуальные и Web-технологии. Курс обеспечивает как приобретение знаний в области воспроизведения творческих процедур (технологии анализа текста на естественном языке, игровые программы [12]), интеллектуализации компьютеров (использование ассистентов и анимированных персонажей), так и приобретение умений и навыков в применении новых принципов обработки информации и решения задач (языки HTML и JavaScript, работа в сети WWW по представлению собственной информации в Интернет).

Литература:

1. Ершов А.П. Компьютеризация школы и математическое образование // Программирование. – 1990. – № 1. – С. 5-25.
2. Ершов А.П. О предмете информатики // Вестник АН СССР. – 1984. – № 2. – С. 112-113.
3. Гриценко В.И., Довгялло А.М. Пути развития информатизации образования // Информатика и образование. – 1989. – № 6. – С. 3-12.

4. Поспелов Г.С. Искусственный интеллект. Новая информационная технология // Вестник АН СССР. – 1983. – № 6. – С. 31-42.
5. Поспелов Г.С., Поспелов Д.А. Искусственный интеллект – прикладные системы. – М.: Знание, 1985. – 48 с. (Новое в жизни, науке, технике. Серия «Математика, кибернетика», № 9).
6. Поспелов Д.А. О «человеческих» рассуждениях в интеллектуальных системах // Логика рассуждений и ее моделирование. – 1983. – С. 5-37.
7. Barr A., Feigenbaum E. Handbook of Artificial Intelligence. – HeurisTech Press (Stanford, Calif.). 1981.
8. Кибернетика и управление производством. – М.: Наука, 1965. – 391 с.
9. Фресс П., Пиаже Ж. Экспериментальная психология. – М.: Прогресс, 1978. – 302 с.
10. Люгер Дж.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. Пер. с англ. - 4-е изд. - М.: Вильямс, 2003. - 864 с.
11. Столяревська А.Л. Про досвід викладання мови Пролог у курсі інформатики в педагогічному вузі // Педагогіка та психологія. Збірник наукових праць ХДПУ. – 1999. – Вип. 13. – С. 89-96.
12. Столяревська А.Л. Стандартизація навчальної дисципліни «Штучний інтелект» як компонента підготовки фахівця в галузі інформатики // Науковий часопис НПУ імені М.П.Драгоманова. Серія №2. Комп'ютерно-орієнтовані системи навчання: Зб. наук. праць / Редкол. – К.: НПУ ім. М.П.Драгоманова, 2005. – №3(10). – С. 234-243.
13. Логический подход к искусственному интеллекту: от классической логики к логическому программированию: Пер. с франц. / Тейз А. и др. – М.: Мир, 1990. – 432 с.
14. Корнилов Е.Н. Программирование шахмат и других логических игр. – СПб.: БХВ-Петербург, 2005. – 272 с.
15. Дмитриева М.В. Самоучитель JavaScript. – СПб.: БНВ-Санкт-Петербург, 2003. – 512 с.
16. Климов А.П. MS Agent. Графические персонажи для интерфейсов. – СПб.: БХВ-Санкт-Петербург, 2005. – 352 с.
17. <http://www.microsoft.com/msagent/downloads/developer.asp>.

Розділ III

Захист інформації та інформаційна безпека

ІНТЕРНЕТ-АДИКЦІЯ, ЯК ФОРМА ЗАЛЕЖНОЇ ПОВЕДІНКИ

Н.М. Бугайова

м. Київ, Інститут психології ім. Г.С. Костюка АПН України

nitelabor@hotmail.com

Активне поширення комп'ютерних технологій, що міцно ввійшли в професійну діяльність і приватне життя сучасних людей, сприяло виникненню такого нового виду поведінкової адикції, як комп'ютерна залежність.

У західній науковій літературі цей феномен одержав назву Internet Addiction Disorder (IAD) – «Інтернет-адиктивний розлад». За матеріалами закордонних досліджень, даний вид нехімічної адикції серед західних користувачів мережі Інтернет наприкінці ХХ століття становив від 1% до 5% (K.S. Young, 1998, M.D. Griffiths, 2000) і в цей час наближається до 10% (H. Cash, J. Parker) [13; 17]. Безумовно, стрімке зростання числа активних користувачів ПК і доступність інтернет-послуг впливають на ступінь поширеності цього виду поведінкової адикції. По даним моніторингу аудиторії користувачів мережі в Росії за період 1992-2004 р. число підлітків, що користуються Інтернетом, збільшилося з 2,0% до 25,0% (www.monitoring.ru).

Комп'ютерні технології, у силу своїх унікальних особливостей, є найбільш привабливими, тому що мережа Інтернет надає користувачам: анонімність, доступність, комфортність, безпеку віртуального середовища, простоту використання Інтернету, інтерактивність, широкі комунікативні можливості, необмеженість інформаційного простору, віртуальну волю (конструювання аутоідентичності, моделювання безліч «Я», трансформацію статі, віку, свободу висловлень, вираження думок, емоцій, почуттів й ін.), альтернативу реальності (коли індивід у віртуальному світі може одержати те, чого в реальному житті не має).

Захоплення являє собою безперервний динамічний, процес, який характеризується усвідомленістю мотиву й цілі, у результаті якого формується упереджене, емоційне відношення до об'єкта або предмета захоплення. Захоплення у вигляді надцінних утворень припускає надцінне захоплення предметом або об'єктом, у результаті якого будь-який інший вид діяльності набуває другорядне значення й/або повністю блокується.

Інтернет-середовище дозволяє індивідові реалізувати три основні види діяльності: комунікативну, пізнавальну та ігрову. Будь-яка діяльність, для якої є характерним прагнення відходу від реальності за допомогою зміни свого психічного стану без застосування хімічних речовин (тобто об'єктом якої є певний поведінковий патерн), що займає домінуюче положення в житті й свідомості індивіда та яка здійснюється їм на шкоду соціальним зв'язкам й іншим, життєво важливим сферам, може викликати нехімічну залежність. І. Марксом (I. Marks, 1990), що досліджував дану проблему, були запропоновані критерії діагностики нехімічних залежностей: наявність

тяги (спонукання до контрпродуктивної поведінкової діяльності); наростаюча напруга аж до завершення діяльності; тимчасове зняття напруги внаслідок завершення діяльності; симптоми абстиненції (повторна тяга, виникнення й посилення напруги через певний час); унікальність зовнішніх проявів, характерних для даного синдрому адикції; виникнення внутрішніх і зовнішніх проявів, якими згодом визначається існування індивіда (дисфорія, тривога); ранні стадії адикції носять гедоністичний характер [15].

До нехімічних адикцій прийнято відносити: гемблінг (патологічну схильність до азартних ігор), працеголізм (Outs, 1972, Mentzel, 1979, Ц.П. Короленко, 1993) [3]; харчові адикції (нервову анорексію, і булімію) (Ц.П. Короленко, Н.В. Дмитрієва, 2000; Gold et al., 1997; Reid, Burr, 2000) [5]; адикції відносин: любовні, сексуальні й уникнення (Faber, O'Guinn, 1992, Simon, 1982, Ц.П. Короленко, Н.В. Дмитрієва, 2000) [5]; спортивну адикцію (адикцію вправлянь) (P. Baekeland, 1970, M. Sachs, D. Pargman, 1984), екстремальні види спорту; прагнення до невинного ризику й створення небезпечних ситуацій (т.зв. стан перманентної війни) (В.В. Постнов, В.А. Дереча, В.В. Карпець, 2004) [8]; інтернет-адикцію (M. Griffiths, 1995, I. Goldberg 1996, M. Orzack, 1998, K. Young, 1998) [16; 17]. Ц.П. Короленко і Н.В. Дмитрієва (2000) також відносять до адиктивних розладів ургентну адикцію (звичку перебувати в стані постійної нестачі часу) [5], а В.В. Постнов і В.А. Дереча (2004) – нав'язливий духовний пошук [7].

Комп'ютерна залежність містить у собі: обесивну пристрась до роботи з комп'ютером (програмуванню, хакерству, іграм); нав'язливий веб-серфінг, інформаційний пошук у вилучених базах даних; гіперзахопленість мережними азартними іграми; кібероніоманію (прагнення робити нові покупки в інтернет-магазинах, без необхідності їхнього придбання й обліку власних фінансових можливостей, нав'язлива участь в інтернет-аукціонах); кіберкомунікативну залежність (спілкування в чатах, участь у телеконференціях і групових іграх); кіберсексуальну залежність (непереборний потяг до обговорення сексуальних тем на еротичних чатах і телеконференціях, відвідуванню порнографічних сайтів і занять віртуальним сексом).

Основними ознаками інтернет-адикції є: всепоглинання Інтернетом; потреба неухильного збільшення часу on-line сеансів; наявність кількаразових, малоефективних спроб скорочення використання інтернет-ресурсів; поява симптомів абстиненції при припиненні інтернет-зв'язку; виникнення проблем з найближчим оточенням (колегами по роботі, однокласниками, членами родини, друзями); прагнення до приховання реального часу, проведеного on-line; залежність настрою від використання Інтернету [10].

В осіб, що страждають інтернет-адикцією частіше відзначаються: депресивні розлади з підвищеним ризиком розвитку суїциду (K. Kim, E. Ryu, M. Chon, 2005, С. Вигонський, 2002) [1; 14], порушення аутосоматичного сприйняття, емоційна нестійкість (С. Вигонський, 2002), схильність до інтелектуалізації, негативізму, нетовариськості, занижена самооцінка, гіперболі-

зовані уяви про ідеальне «Я», схильність до відходу від проблем й уникнення відповідальності, існування однієї або більше фрустрованих потреб (Н.В. Чудова, 2002) [9], почуття самотності (Н.В. Чудова, 2002, Y. Amichai-Hamburger, E. Ben-Artzi, 2003) [9; 11].

Інтернет-простір, маючи практично необмежені ресурси, надає користувачам можливість одержання різноманітної наукової, соціальної, політичної, розважальної й приватної інформації, що зберігається в текстових, відео, аудіо та ін. форматах. Однією з головних проблем, одержуваної через інтернет інформації, є її якість, вірогідність і безпека. Активність індивіда в інтернет-середовищі спрямована на задоволення своїх потреб. З одного боку, Інтернет створює середовище, що дозволяє вирішувати різні завдання інформаційного, комунікативного, професійного характеру. З іншого боку, через відсутність інформаційного контролю, в інтернет-просторі можуть розміщуватись матеріали дезінформаційної, агресивної або протизаконної спрямованості.

Користувач не завжди здатний реально оцінити рівень вірогідності й безпеки інформаційних матеріалів, які знаходяться в інтернет мережі. Так, інформаційні web-ресурси агресивної й аутоагресивної спрямованості створюють підвищену небезпеку для осіб з нестійкою психікою і є теоретичною базою для агресивно та аутоагресивно настроєних осіб. Нерідко, відсутність у користувачів Інтернет культури й елементарних навичок правильної роботи в мережі створює ряд проблем психологічного й соціального характеру. Неконтрольоване й нераціональне використання інтернет-ресурсів учнями й студентами спричиняє неуспішність у навчанні й виникнення академічних заборгованостей. Використання робочого доступу в Інтернет в особистих цілях працівниками й службовцями знижує продуктивність праці, провокує виникнення виробничих конфліктів. У результаті дослідження, проведеного Harris Interactive серед американських компаній, було встановлено, що в середньому, один день на тиждень витрачається працівниками на відвідування сайтів, не пов'язаних із професійною діяльністю. При цьому 67% опитаних зізналися у використанні Інтернету на робочому місці по особистим мотивам: 24% співробітників здійснюють інтернет-покупки, 23% – переглядають новини, 18% – відвідують порнографічні сайти, 8% грають в азартні ігри, 6% беруть участь в on-line аукціонах [2].

Стрімкий розвиток комп'ютерних технологій веде до видозміни й ускладнення інформаційного середовища. Кіберпростір, будучи невід'ємною складовою сучасного світу, містить у собі все більше число людей, діяльність яких у мережі Інтернет має свої специфічні особливості. Феномен інтернет-адикції як нового виду поведінкової залежності, вимагає серйозного всебічного вивчення з метою розробки психологічних програм і технологій, спрямованих на корекційну роботу осіб, що страждають даним видом нехімічної адикції й профілактику серед користувачів Інтернету.

Література:

1. Выгонский С. Психиатрия недооценивает интернет-зависимость 4.01. 2002. <http://www.membrana.ru/articles/interview/2002/01/04/165300.html>
2. Доступ в Сеть на рабочем месте ведет к интернет-зависимости. 22.08.2002 20:29. <http://news.gala.net/?cat=14&id=69823>
3. Короленко Ц.П. Аддиктивное поведение. Общая характеристика и закономерности развития // *Обозрение психиатр и мед. психол.* – 1991. – №1. – С. 8-15.
4. Короленко Ц.П. Работоголизм – респектабельная форма аддиктивного поведения // *Обозрение психиатрии и мед. психологии.* – 1993. – №1. – С. 17-29.
5. Короленко Ц.П., Дмитриева Н.В. Социодинамическая психиатрия. – М.: Академический Проект; Екатеринбург: Деловая книга, 2000. – 460 с.
6. Менделевич В.Д. Наркозависимость и коморбидные расстройства поведения. – М.: МЕДпресс-информ, 2003. – С. 328.
7. Постнов В.В., Дереча В.А. Духовный поиск как вариант нехимической аддикции у больных алкоголизмом в ремиссии // *Новые методы лечения и реабилитации в наркологии / Сб. мат-лов междунар. конф. Под. общ. ред. проф. В.Д. Менделевича.* – Казань, 2004. – С. 287-291.
8. Постнов В.В., Дереча В.А., Карпец В.В. Аддиктивное поведение в форме «состояния перманентной войны» в структуре расстройств адаптации у больных алкоголизмом – ветеранов боевых действий // *Новые методы лечения и реабилитации в наркологии / Сб. мат-лов междунар. конф. Под. общ. ред. проф. В.Д. Менделевича.* – Казань, 2004. – С. 291-295.
9. Чудова И.В. Особенности образа «Я» «Жителя Интернета» // *Психологический журнал.* – 2002. – Т. 22, № 1. – С. 113-117.
10. Янг К.С. Интернет-зависимость // *Мир Интернет.* – 2000. – №2. – С. 24-29.
11. Amichai-Hamburger Y., Ben-Artzi E. Loneliness and internet use // *Computers and Human Behavior.* – 2003. – Jan., V.1, N.19. – P. 71-80.
12. Greenfield D.M. Virtual Addiction: Help for Netheads, Cyberfreaks, and Those Who Love Them. – Oakland: New Harbinger Publ, 1999.
13. Griffiths M.D. Internet addiction – time to be taken seriously? // *Addiction Research.* – 2000. – Oct. V. 8. N. 5. – P. 413-419.
14. Kim K., Ryu E., Chon M.Y., Yeun E.J., Choi S.Y., Seo J.S., Nam B.W. Internet addiction in Korean adolescents and its relation to depression and suicidal ideation: A questionnaire survey // *International J. of Nursing Studies.* – 2005.
15. Marks I. Behavioural (non-chemical) addictions // *British J. Addict.* – 1990. – V.85. – P. 1389–1394.
16. Orzack M.H. Computer addiction: What is it? // *Psychiatric Times.* – August. – 1998. – V.15, N8.
17. Young K.S. Internet addiction: The emergence of a new clinical disorder // *CyberPsychology and Behavior.* – 1998. – V.1. – P. 237-244.

РЕАЛІЗАЦІЯ МІЖПРЕДМЕТНИХ ЗВ'ЯЗКІВ РОЗДІЛІВ «ГЕОМЕТРИЧНА ОПТИКА» ТА «ТЕХНІЧНІ КАНАЛИ ВИТОКУ ІНФОРМАЦІЇ»

А.О. Москаленко¹, Ю.П. Бендес¹, Ю.І. Мазнюк²

¹ м. Полтава, Полтавський військовий інститут зв'язку

² м. Полтава, Полтавський державний педагогічний університет

ім. В.Г. Короленка

artem_mo@mail.ru

Проблема захисту інформації є досить актуальною на даний час, про що свідчить велика кількість публікацій у засобах масової інформації відносно порушення інформаційної безпеки та величезні збитки, які є наслідком цих порушень.

Як було зазначено в роботі [1], головною причиною порушення інформаційної безпеки є недостатність знань фахівців усіх сфер професійної діяльності в області захисту інформації. На думку авторів, існують наступні шляхи подання основ захисту інформації у розрізі фундаментальної загальної фізичної освіти: демонстрації, професійно орієнтовані задачі, лабораторні роботи, гурткова та самостійна роботи.

Захист інформації – діяльність, спрямована на забезпечення таких властивостей інформації, як цілісність, доступність та конфіденційність [2]. Прийнято розділяти захист інформації на три взаємопов'язані складові: технічний захист інформації, криптографічний захист інформації та захист інформації в автоматизованих системах. Взаємозв'язок складових захисту інформації з дисциплінами шкільного курсу зображено на рис. 1.

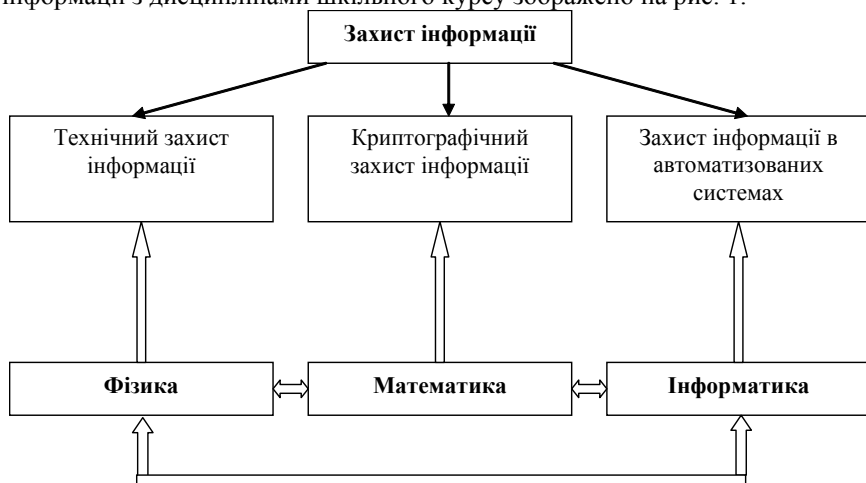


Рис. 1. Взаємозв'язок захисту інформації з дисциплінами шкільного курсу

Як видно з нього, захист інформації забезпечується трьома дисциплінами, що вивчаються в старшій школі: фізика, математика та інформатика. В свою чергу, технічний захист інформації поділяється на технічні канали витоку інформації та технічні методи і засоби захисту інформації [3]. Під технічним захистом інформації розуміють діяльність, спрямовану на запобігання порушенню цілісності, блокуванню та (чи) витоку інформації по технічних каналах [4].

Класифікація технічних каналів витоку інформації зображено на рис. 2.

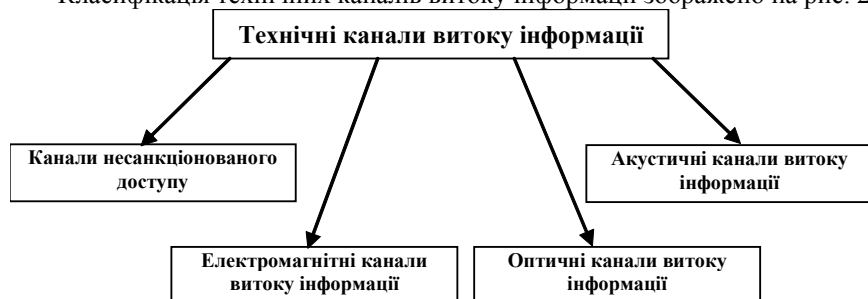


Рис. 2. Класифікація технічних каналів витоку інформації

Знання в галузі фізики дають можливість глибше зрозуміти принципи утворення технічних каналів витоку інформації. А наочне застосування абстрагованих фізичних закономірностей, понять і процесів, що проявляються на практиці в реальних технічних каналах витоку інформації, доповнює і систематизує набуті знання з фізики. Органічне поєднання знань, набутих при вивченні різних дисциплін, та глибоке розуміння існуючих між ними зв'язків допомагає учням усвідомити фундаментальні закони природи.

Отже, вивчення основ захисту інформації, як засобу формування фізичних і професійних знань можливе лише у контексті міжпредметних взаємозв'язків цих дисциплін.

В даній роботі розглядається реалізація міжпредметних зв'язків оптичних каналів витоку інформації та розділу фізики «Геометрична оптика», які представлені на рис. 3.

Далі зупинимось на можливостях реалізації міжпредметних зв'язків геометричної оптики та оптико-акустичного каналу витоку інформації.

Одним з найбільш перспективних напрямків реалізації такого підходу є професійно орієнтовані задачі. До їх складу повинна входити теорія із основ захисту інформації, що пов'язана з даною задачею та, безпосередньо, фізична задача. Така побудова професійно орієнтованих задач надасть наступні переваги:

- ознайомлення з теоретичними положеннями основ захисту інформації;

- формування світоглядних і прикладних знань за рахунок їх доповнення та систематизації набутих фізичних знань;
- зацікавленість та активізація пізнавальної діяльності у процесі навчання фізиці;
- допомога в усвідомленні фундаментальних законів природи, за рахунок поєднання знань, набутих при вивченні різних дисциплін.



Рис. 3. Міжпредметні зв'язки розділу фізики «Геометрична оптика» та захисту інформації «Технічні канали витоку інформації»

Прикладом такого підходу є задача на оптико-акустичний канал витоку інформації.

Задача. Лазерний засіб акустичної розвідки складається із джерела та приймача лазерного випромінювання, що розташовані на відстані $d=40$ см один від одного. При виконанні лазерно-локаційного зондування об'єкту лазерний промінь переходить з повітря в скло ($n=1,5$), причому кут заломлення складає 1° . Знайти відстань від засобу акустичної розвідки до віконного скла.

Хід розв'язку задачі та аналіз її результатів доречно супроводжувати інформацією про оптико-акустичний канал витоку.

Перехоплення акустичної інформації із приміщень може здійснюватися за допомогою лазерних засобів акустичної розвідки. В цьому випадку застосовується дистанційне лазерно-локаційне зондування об'єктів, що являються потенційними джерелами таємної акустичної інформації. В якості таких об'єктів можуть виступати віконне скло та інші поверхні, що здатні відбивати світлові промені та коливаються під дією акустичних хвиль.

Структурна схема оптико-акустичного каналу витоку інформації зображена на рис. 4.

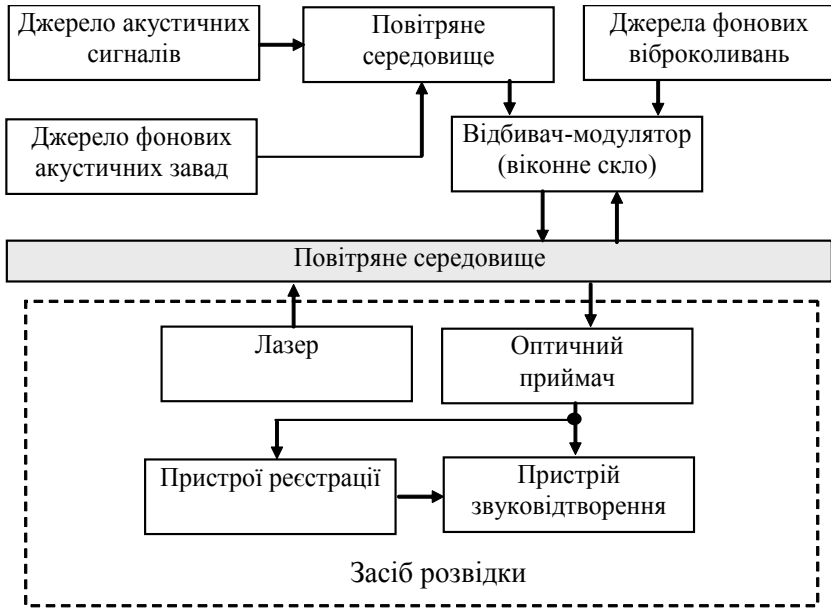


Рис. 4. Структурна схема оптико-акустичного каналу витоку інформації

Коливання, що генерується лазерним передавачем, спрямовується на віконне скло приміщення, в якому обговорюється інформація.

Акустичні хвилі, що виникають при розмові, розповсюджуючись в повітряному середовищі, діють на віконне скло і викликають його коливання в діапазоні частот, що відповідає мовному повідомленню. Таким чином, відбувається віброакустичне перетворення мовного повідомлення в мембрані, роль якої виконує віконне скло. Лазерне випромінювання, що падає на зовнішню поверхню віконного скла (мембрани), в результаті вібро-оптичного перетворення модулюється сигналом, що викликає коливання мембрани. Відбитий оптичний сигнал приймається оптичним приймачем, в якому відбувається відтворення повідомлення.

На теперішній час створені різноманітні системи лазерних засобів акустичної розвідки, що мають дальність дії від десятків метрів до одиниць кілометрів. Наприклад, система SIPE LASER 3-DA SUPER складається із джерела випромінювання (гелій-неонового лазера), приймача цього випромінювання з блоком фільтрації шумів, двох пар головних телефонів, акумулятора живлення та штатива. Наведення лазерного випромінювання на віконне скло необхідного приміщення здійснюється за допомогою телескопічного візира. Використання спеціальної оптичної насадки дозволяє регулювати кут розходження вихідного світлового пучка. Система забезпечує перехоплення мовної інформації з доброю якістю на відстані до 250 м. В лазе-

рному пристрої НРО150 в якості передавача також використовується гелій-неоновий лазер. До складу приймача включено блок компенсації завад і касетний пристрій магнітного запису, а дальність ведення розвідки складає близько 1000 м.

Запропонований у даній статті підхід дозволяє надавати у розрізі фундаментальних фізичних знань елементи професійної спрямованості через застосування орієнтованих фізичних задач при проведенні, як аудиторних занять, так і при організації самостійної та гурткової роботи. Крім того, міжпредметні зв'язки фізики та захисту інформації, якщо їх використовувати при викладанні у вищих навчальних закладах інженерного профілю, дають можливість студентам та курсантам зрозуміти напрямки прикладного застосування набутих фізичних знань і трансформувати ці знання на інші сфери діяльності.

Література:

1. Москаленко А.О., Струць В.А, Бендес Ю.П. Впровадження елементів захисту інформації у курс фізики // Матеріали всеукраїнської науково-практичної конференції “Кредитно-модульна технологія навчання та методичне забезпечення контролю якості успішності” – Полтава, 2006. – 162 с.
2. НДТЗІ 1.1-003-99 «Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу».
3. Технические методы и средства защиты информации / Ю.Н. Максимов, В.Г. Сонников, В.Г. Петров и др. – СПб.: ООО «Издательство Полигон», 2000. – 320 с., ил.
4. ДСТУ 3396.2-97 Захист інформації. Технічний захист інформації. Терміни та визначення.

НАВЧАННЯ КРИПТОГРАФІЧНИМ МЕТОДАМ ЗАХИСТУ ІНФОРМАЦІЇ

О.О. Кононова, Н.В. Моїсеєнко
м. Кривий Ріг, Криворізький державний педагогічний університет
n_v_moiseenko@yahoo.com

В сучасному світі рівень комп'ютеризації відіграє значну роль у розвитку країни. З еволюцією обчислювальної техніки людство пов'язувало надії на різкий стрибок в інтелектуальній і духовній областях, інформатиці, сервісі, обороноздатності і багато чому іншому, що може зробити наше життя безтурботніше. Але, на жаль, далеко не всім надіям вдалося здійснитися. Разом з безперечними благами комп'ютеризація ще більш порушила нашу безпеку. Кількість помилок, пов'язаних з переоцінкою надійності і захищеності обчислювальної техніки, постійно зростає.

В персональних комп'ютерах і в обчислювальних мережах зосереджується інформація, виняткове право на використання якої належить певним особам або групам осіб, діючим в порядку особистої ініціативи або відповідно до посадових обов'язків. Така інформація повинна бути захищеною від всіх видів стороннього втручання. Цю проблему слід розглядати глобально. Необхідно захищати інформацію на всьому шляху її руху від відправника до одержувача.

Основні напрями використання криптографічних методів – передача конфіденційної інформації по каналах зв'язку (наприклад, електронна пошта), встановлення автентичності повідомлень, що передаються, зберігання інформації (документів, баз даних) на носіях в зашифрованому вигляді.

Криптографічні методи захисту інформації в автоматизованих системах можуть застосовуватися як для захисту інформації, яка підлягає комп'ютерній обробці, так і для закриття інформації, що передається між різними елементами системи по лініях зв'язку. На теперішній час розроблена велика кількість різних методів шифрування.

При підготовці спеціалістів за напрямком «прикладна математика» в криворізькому державному педагогічному університеті викладається дисципліна “Криптографічні методи захисту інформації”.

В лекційній частині курсу викладаються такі теми:

1. Криптографія та оточуюче середовище.
2. Вступ до криптографії.
3. Симетричні криптоалгоритми (скремблери, блочні шифри)
4. Симетричні криптосистеми
5. Асиметричні криптоалгоритми
6. Управління ключами

Під час лекцій, в яких розбираються алгоритми шифрування (хоча б найпростіші), використовуються активні форми роботи. Наприклад, студен-

там пропонується провести криптоаналіз тексту зашифрованого підстановочним шифром. В результаті підвищується не тільки розумова активність, але і мотивація студентів при вивченні матеріалу.

Для підтримки курсу використовується гіпертекстовий електронний підручник, створений в рамках виконання дипломної роботи С.В. Кукліною.

В лабораторній частині курсу значний час приділяється програмній реалізації криптографічних систем, оскільки вона заслуговує особливої уваги при підготовці інженерів-програмістів. Однак, в межах семестрового курсу неможливо виконати повну програмну реалізацію всіх алгоритмів, які треба вивчити, тому деякі алгоритми вивчаються за допомогою готових програм. Бібліотека криптографічних програм ефективно використовується для перевірки засвоєння знань, але ще більший ефект дають спеціалізовані навчаючі програми.

Наприклад для вивчення блочного шифру DES використовується програма Destutor (рис. 1), яка працює в різних режимах (рис. 2).

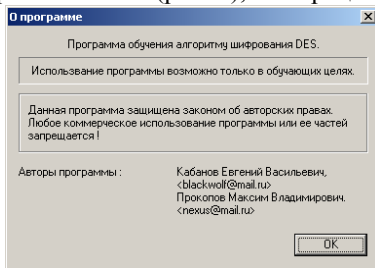


Рис. 1. Про авторів

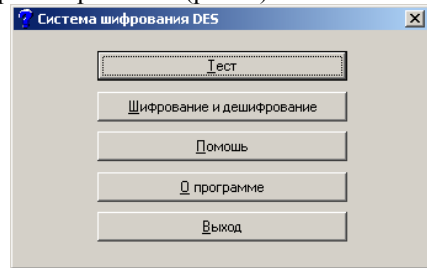


Рис. 2. Головне меню

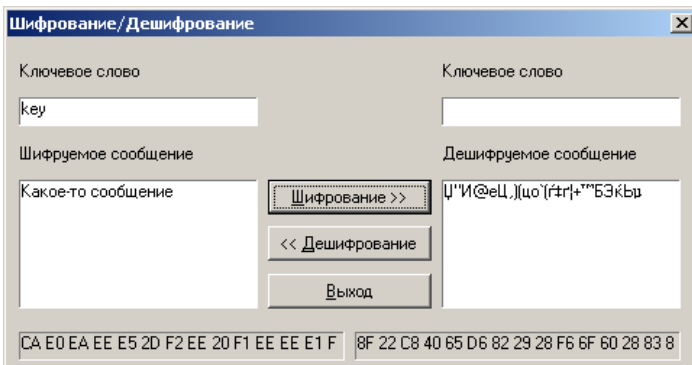


Рис. 3. Режим шифрування та дешифрування

В режимі шифрування та дешифрування потрібно ввести вихідний текст і ключ та натиснути потрібну кнопку (рис. 3).

В режимі тестування спочатку пропонується ввести повідомлення, що шифрується, та ключ (рис. 4).

Після натиснення кнопки “Вперед” виконуються етапи шифрування DES. На кожному етапі при натисненні кнопки “Демонстрація” відбувається демонстрація першої його частини (рис. 5), що дає можливість студентам згадати алгоритм, або вперше з ним ознайомитись. Далі пропонується продовжити виконання поточного етапу самостійно (рис. 6). Це, по-перше, сприяє закріпленню теоретичного матеріалу, а по-друге, може бути використане викладачем для перевірки студентів.

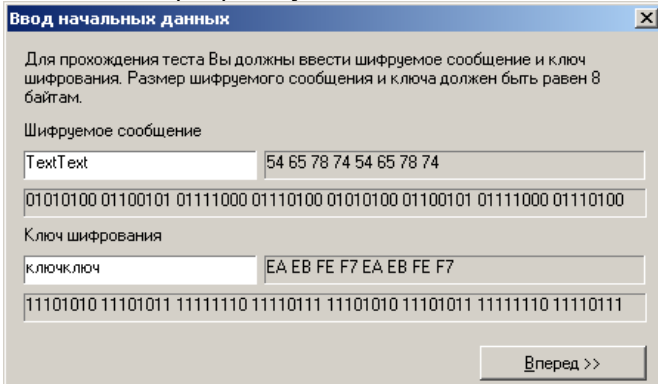


Рис. 4. Экран введения початкових даних

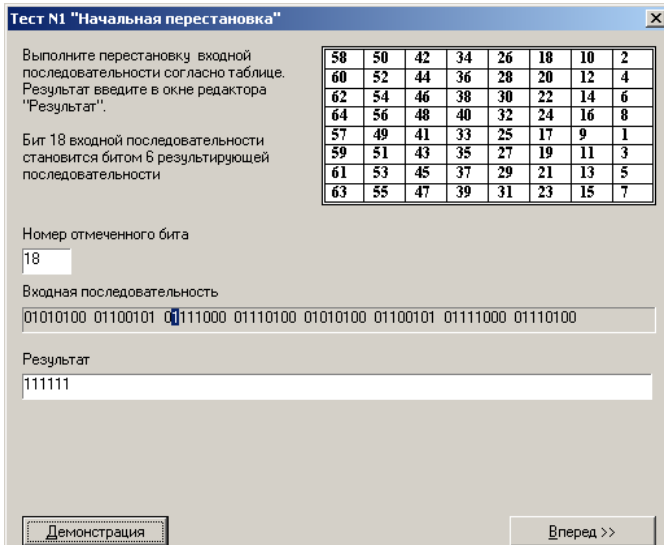


Рис. 5. Демонстрація виконання I етапу DES (початкова перестановка)

Після виконання всіх етапів демонструється результат виконаних дій (рис. 7).

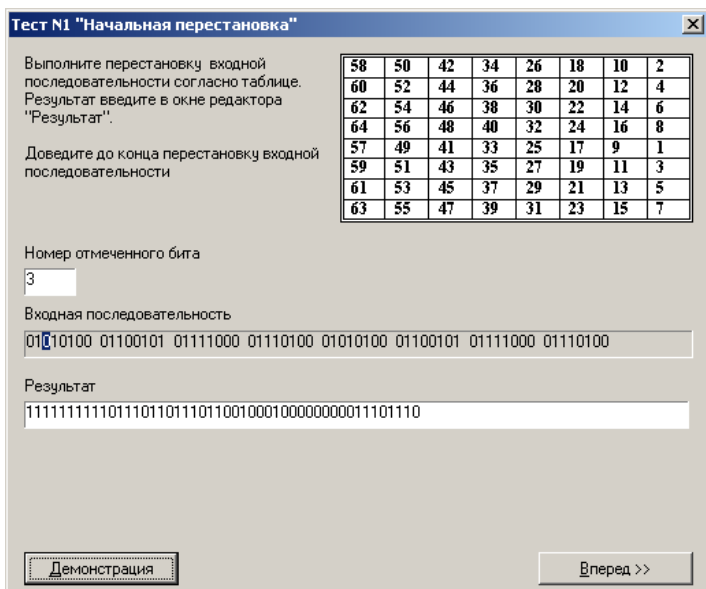


Рис. 6. Продовження виконання I етапу DES (початкова перестановка)

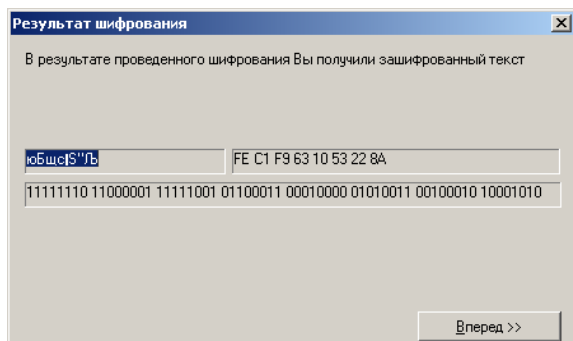


Рис. 7. Результат шифрування

Наприкінці тесту виводяться результати: вказується скільки бітів з самостійно знайдених є невірними і відсоток помилок у відповідях (рис. 8).

В програмі також наявний режим "Допомога", який представляє собою опис алгоритму, що може бути корисним для самостійної роботи студентів.

Незважаючи на те, що подібні навчачі програми здебільшого дають хороший результат, в їх використанні треба знати міру. Воно є виправданим для студентів, спеціальністю яких не є програмування, але при підготовці інженерів-програмістів все ж таки більше уваги треба приділяти тонкощам реалізації криптографічних алгоритмів.

Результаты теста		
Просмотрите результаты тестирования и оцените свои знания !		
Название	Количество неверных битов	Ошибочность, %
Начальная перестановка	64	100
Получение L(0) и R(0)	0	0
Последовательность В	56	100
Получение C(0) и D(0)	0	0
Получение C(i)	0	0
Получение D(i)	8	28
Получение K(i)	48	100
Получение S(i)	0	0
Последовательность E	48	100
Функция шифрования	64	100
Конечная перестановка	64	100

Вперед >>

Рис. 8. Результаты тесту

Література:

1. Баричев С. В. Криптография без секретов. – М.: Наука, 1998.
2. Введение в криптографию / Под общ. ред. В.В. Яценко. – М.: МЦНМО: «ЧеРо», 2000. – 288 с.
3. Вербицкий О.В. Вступление в криптологию. – Львов: Издательство научно-технической литературы, 1998.
4. Дориченко С.А., Яценко В.В. 25 этюдов о шифрах. – М.: Теис, 1994.
5. Жельников В. Криптография от папируса до компьютера. – М.: Наука, 1996.
6. Криптология – наука о тайнописи // Компьютерное обозрение. – 1999. – №3 – С. 10–17.
7. Фергюсон Н., Шнайер Б. Практическая криптография. – М.: Диалектика, 2005. – 424 с.
8. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Издательство Триумф, 2003. – 816 с.

АНАЛІЗ НЕНАДІЙНОСТІ КРИПТОСИСТЕМ

В.М. Джулій

м. Хмельницький, Хмельницький національний університет
fortuna777@ukr.net

У даний час криптографічні методи захисту використовуються в інформаційних системах будь-якого ступеня складності й призначення. Криптографічними методами захищається державна таємниця, забезпечується законність електронного документообігу, запобігають спробам шахрайства в системах електронної торгівлі. Без криптографії забезпечити необхідний ступінь безпеки в сучасному комп'ютеризованому світі вже не представляється можливим. Згодом її роль і значення обіцяють стати ще більшими.

Наскільки б не були складні й надійні криптографічні системи, їхнє слабке місце при практичній реалізації – проблема розподілу ключів. Для того, щоб був можливий обмін конфіденційною інформацією між двома суб'єктами інформаційної системи (ІС), ключ повинен бути згенерований одним з них, а потім якимось чином знову ж у конфіденційному порядку переданий іншому. Таким чином, у загальному випадку, для передачі ключа знову ж потрібне використання якоїсь криптосистеми. Для вирішення цієї проблеми на основі результатів, отриманих класичною й сучасною алгеброю, були запропоновані системи з відкритим ключем (рис. 1). Суть їх полягає в тому, що кожним адресатом ІС генеруються два ключі, зв'язані між собою за певним правилом. Один ключ оголошується відкритим, а інший закритим. Відкритий ключ публікується й доступний кожному, хто бажає послати повідомлення адресатові. Секретний ключ зберігається в таємниці. Вихідний текст шифрується відкритим ключем адресата й передається йому. Дешифрування повідомлення можливо тільки з використанням закритого ключа, що відомий тільки самому адресатові. Криптографічні системи з відкритим ключем використовують так звані необоротні або односторонні функції, які мають наступну властивість: при заданому значенні x відносно просто обчислити значення $f(x)$, однак якщо $y=f(x)$, то немає простого шляху для обчислення значення x . Множина класів необоротних функцій і породжує всю розмаїтість систем з відкритим ключем. У самому визначенні необоротності присутня невизначеність. Під необоротністю розуміється не теоретична необоротність, а практична неможливість обчислити зворотне значення, використовуючи сучасні обчислювальні засоби за доступний для огляду інтервал часу.

Тому, щоб гарантувати надійний захист інформації, до систем з відкритим ключем (СВК) пред'являються дві важливих і очевидних вимоги:

- перетворення вихідного тексту повинне бути необоротним і виключати його відновлення на основі відкритого ключа.
- визначення закритого ключа на основі відкритого також повинне

бути неможливим на сучасному технологічному рівні. При цьому бажана точна нижня оцінка складності (кількості операцій) розкриття шифру.

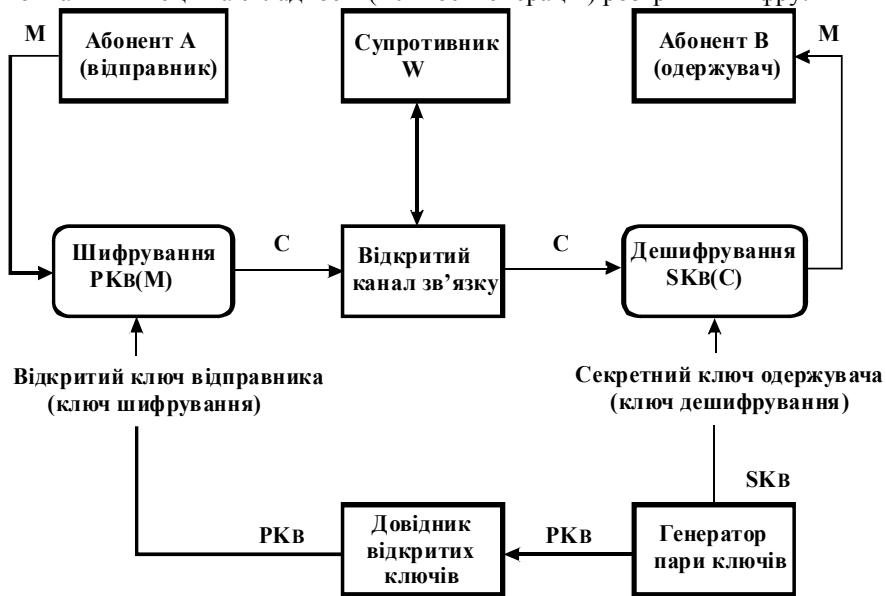


Рис. 1. Модель криптосистеми з відкритим ключем

Алгоритми шифрування з відкритим ключем одержали широке поширення в сучасних інформаційних системах.

Сучасна криптографія надає всі необхідні алгоритми, методи й засоби, які дозволяють побудувати систему захисту, витрати на злом якої такі, що в супротивника з обмеженими фінансовими й технічними можливостями для одержання інформації, що його цікавить, залишаються тільки дві можливості:

- використання людського фактора;
- використання особливостей конкретної реалізації криптоалгоритмів

і криптопротоколів, що найчастіше залишається бажати кращого.

Саме такий висновок можна зробити, аналізуючи приклади реальних успішних атак на криптосистеми. Відомі лише одиничні випадки злomu з використанням винятково математичних методів. У той же час різних прикладів зломів реальних систем так багато, що їхнім аналізом змушені займатися цілі компанії, найбільш відома з яких Counterpane Systems Б. Шнайера. Система захисту в цілому не може бути надійніше окремих її компонентів. Іншими словами, для того щоб перебороти систему захисту, досить зламати або використати для злomu самий ненадійний з її компонентів. Найчастіше причинами ненадійності реальних систем криптографічного захисту є людський фактор та особливості реалізації.

Сама ненадійна ланка системи – людина. Типові помилки користувачів, що порушують безпеку всієї системи захисту:

- надання свого секретного пароля колегам по роботі для рішення невідкладних завдань під час відсутності власника пароля;
- повторне використання секретних паролів у несекретних системах;
- генерація паролів самими користувачами, вибір паролів за критерієм зручності запам'ятовування;
- несвоєчасне інформування про компрометацію ключової інформації, наприклад про втрату smart-карт.

Одержують поширення атаки типу відмова в обслуговуванні (denial of service), що провокують користувача відключати систему захисту, «що зависає», при вирішенні невідкладних завдань (рис. 2б).



Активна атака - роз'єднання (порушення доступності компонента або ресурсу інформаційної системи)

Рис. 2. Атака типу відмова в обслуговуванні

Можна виділити наступні причини ненадійності криптосистем, пов'язані з особливостями їхньої реалізації:

- застосування нестійких криптоалгоритмів;
- неправильне застосування криптоалгоритмів;
- помилки в реалізації криптоалгоритмів.

В деяких випадках, особливо в системах реального часу, застосування стійких алгоритмів принципово неможливо в силу їхньої низької швидкодії й тому вимушено використовуються менш стійкі, але швидкі криптоалгоритми. Багато якісних криптографічних засобів підпадають під дію експортних обмежень, що штучно знижують якість цих засобів. Наприклад, у США заборонений експорт симетричних криптоалгоритмів з довжиною ключа більше 56 біт. Всі програмні засоби, зроблені в США й легально експортовані за рубіж, забезпечують ослаблений криптографічний захист. Аналогічна ситуація має місце й у Європі. Так, наприклад, існує дві версії алгоритму

потокowego шифрування A5 (стандарт GSM) – надійна A5/1 і істотно менш стійка A5/2 для поставок у країни, що розвиваються.

Багато розроблювачів програмного забезпечення (ПЗ) включають у свої продукти власні криптоалгоритми, самовпевнено вважаючи себе фахівцями, забуваючи, що сучасна криптографія, заснована на глибоких дослідженнях у таких розділах математики, як вища алгебра, теорія чисел, теорія інформації, теорія складності обчислень і ін. Якщо розроблювачі роблять ставку на власні методи, шанси зломщиків, навіть у випадку повної відсутності на початковому етапі інформації про використаний алгоритм, багаторазово зростають.

Основні помилки при застосуванні криптоалгоритмів:

- недостатня довжина ключа, неякісна процедура керування ключами;
- неякісний генератор псевдовипадкової послідовності або неправильна його ініціалізація;
- використання криптоалгоритмів не по призначенню, наприклад зберігання паролів у зашифрованому, а не в хешованому виді;
- використання на практиці моделі довірчих відносин, відмінної від тієї, розраховуючи на яку проектувалася система.

Помилки в реалізації криптоалгоритмів. Дана причина ненадійності криптосистем у силу своєї нетривіальності й різноманіття вимагає окремого розгляду, тому обмежимося лише коротким перерахуванням основних проблем, що виникають при реалізації криптоалгоритмів. Надійна система захисту повинна вміти оперативно виявляти несанкціоновані дії для мінімізації можливого збитку. У випадку виявлення ушкоджень у системі повинні включатися ефективні процедури відновлення зруйнованих елементів. Система не повинна втратити живучість навіть у випадку проведення успішної атаки на неї. Причини наявності більшості «дір» у програмному забезпеченні, тобто не описаних у документації можливостей роботи з ним, очевидні; безпам'ятність розроблювачів, які в процесі відлагодження продукту створюють тимчасові механізми, що полегшують її проведення (наприклад, за рахунок прямого доступу до відлагоджуємих частин програми). По закінченні відлагодження частина «дір» убирається, а про частину розроблювачі благополучно забувають або залишають їх свідомо, особливо в ранніх версіях продукту, коли в майбутньому досить імовірна його доробка. «Діри» можуть бути наслідком застосування технології розробки програм «зверху вниз», коли програміст відразу приступає до написання керуючої програми, замінюючи передбачувані в майбутньому підпрограми «заглушками», що імітують реальні підпрограми або просто позначають місце їхнього майбутнього приєднання. Дуже часто ці «заглушки» залишаються в кінцевій версії програми – або знову ж через безпам'ятність, або розраховуючи на майбутню модифікацію продукту, або, наприклад, якщо в процесі розробки з'ясовується, що якась підпрограма не потрібна, а видалити заглушку не

представляється можливим/ У випадках виявлення такої заглушки злоумисник може скористатися нею для підключення до програми своєї підпрограми, що працює аж ніяк не в інтересах законного користувача.

Наступне джерело «дір» – неправильна обробка (або її відсутність) нестандартних ситуацій, які можуть мати місце при роботі програми: невизначене введення, помилки користувачів, збої й т.п. В цьому випадку супротивник може штучно викликати в системі появу такої нестандартної ситуації, щоб виконати потрібні йому дії. Наприклад, він може викликати аварійне завершення програми, що працює в привілейованому режимі, щоб, перехопивши керування, залишитися в цьому привілейованому режимі. Нарешті, відомі випадки, коли «дірки» у ПЗ або апаратурі – перший крок до атаки системи безпеки. Розроблювач навмисне залишає його в кінцевому продукті, щоб у майбутньому, наприклад, мати можливість модифікувати інформацію непомітно для законного користувача, розшифрувати її, не знаючи ключа, і т.п.

Існують програми, споконвічно призначені для руйнівних дій: це комп'ютерні віруси, комп'ютерні черв'яки й програми-закладки, серед останніх, у свою чергу, можна виділити логічні бомби, троянські коні та ін. З повною на те підставою вони одержали узагальнену назву руйнуючих програмних впливів (РПВ). РПВ можуть виконувати одну або декілька з перерахованих дій, небезпечних для системи захисту:

- несанкціоноване копіювання й знімання секретної інформації;
- приведення в непрацездатний стан системи захисту;
- знищення або модифікація секретної інформації;
- спостереження за процесами обробки секретної інформації й принципами функціонування засобів захисту;
- постійна або короткочасна (що небезпечніше) підміна або зниження стійкості використовуваних криптоалгоритмів;
- постійна або короткочасна зміна ступеня захищеності секретної інформації;
- створення прихованих каналів передачі інформації;
- ініціювання раніше впроваджених РПВ;
- ініціювання режимів роботи, що збільшують обсяг фрагментів, що залишаються після обробки секретної інформації.

Ще більш різноманітні шляхи впровадження РПВ. Можна виділити наступні засоби, призначені для боротьби із РПВ, без яких будь-яка програмна реалізація криптоалгоритма практично беззахисна:

- засоби, що перешкоджають впровадженню, і засобу виявлення РПВ до використання програмних продуктів по призначенню;
- засоби, що забезпечують оперативне виявлення РПВ у процесі реального функціонування ПЗ, споконвічно вільного від них;
- засоби видалення РПВ;
- засоби визначення факту наявності або відсутності РПВ у ПЗ, що

додається в систему.

Апаратуру легше фізично захистити від проникнення ззовні. Крипто-модулі можуть розміщуватися в особливі контейнери, які унеможливають зміну алгоритму функціонування. Інтегральні схеми можуть покриватися спеціальним хімічним складом, при цьому будь-яка спроба подолання захисного шару приводить до самознищення їхньої внутрішньої логічної структури. Проте відомі випадки виявлення і апаратних закладок.

Крім того, виникає проблема захисту від екзотичних атак, застосовуваних до реалізацій в smart-картах, – часового аналізу й аналізу споживаної потужності. Ці атаки засновані на використанні того факту, що різні операції, виконувані на мікропроцесорі, вимагають різного часу, а також приводять до різного споживання потужності. Загальна ідея цих атак у тім, що, аналізуючи часові характеристики алгоритму (час відповіді) або споживання потужності, ми можемо скласти картину виконання різних операцій і навіть приблизно обчислити їхні аргументи.

Приблизна аналіз уразливості різних операцій з погляду тимчасових характеристик дає наступні результати:

- пошук по таблицях – неузвзимий для часових атак;
- фіксовані зсуви – неузвзимі для часових атак;
- булеві операції – неузвзимі для часових атак;
- додавання/віднімання – важко захистити від часових атак.
- множення/ділення – найбільш уязвимі для часових атак операції.

Стійкість до атак такого роду, спрямованих не на криптоалгоритм, а на його реалізацію, також треба враховувати. Захищеність стосовно часового аналізу можна підвищити шляхом введення додаткових затримок. Більш складною є проблема захисту від аналізу потужності, але її можна вирішити декількома шляхами. Це, по-перше, балансування алгоритму (рівномірний розподіл різних операцій по коду), по-друге, введення спеціальних «шумових» операцій або просто вибір іншого мікропроцесора.

Останнім часом одержали поширення атаки на апаратуру криптосистем, засновані на аналізі електромагнітного випромінювання й інших побічних джерел інформації. Одержують поширення по суті «біологічні» методи злому, що розглядають криптосистеми як складні об'єкти, певним чином реагуючі на зовнішні подразники. Атаки подібного роду засновані на аналізі поведінки системи після випадкових або навмисних збоїв у роботі.

Основною перевагою програмних методів реалізації захисту є їхня гнучкість, тобто можливість швидкої зміни алгоритмів шифрування. Основним же недоліком програмної реалізації є істотно менша швидкодія в порівнянні з апаратними засобами (приблизно в 10 разів). Останнім часом стали з'являтися комбіновані засоби шифрування, так звані програмно-апаратні засоби. У цьому випадку в комп'ютері використовується своєрідний «криптографічний співпроцесор» – обчислювальний пристрій, орієнтований на виконання криптографічних операцій (додавання по модулі, зсув й т.д.).

Міняючи програмне забезпечення для такого пристрою, можна вибрати той або інший метод шифрування. Такий метод поєднує в собі переваги програмних і апаратних методів.

Незважаючи на успіхи сучасної криптографії, задача побудови надійної системи криптографічного захисту комплексна, вона значно складніша, ніж здається на перший погляд. Надійна система захисту може бути побудована тільки з врахуванням всіх перерахованих факторів. Таким чином, вибір типу реалізації криптозахисту для конкретної ІС в істотній мірі залежить від її особливостей і повинен опиратися на всебічний аналіз вимог, пропонуваних до системи захисту інформації.

Вибір криптозахисту для конкретних ІС повинен бути заснований на глибокому аналізі слабких і сильних сторін тих або інших методів захисту. Обґрунтований вибір тієї або іншої системи захисту взагалі ж повинен опиратися на критерії ефективності. На жаль, дотепер не розроблені підходящі методики оцінки ефективності криптографічних систем. Найбільш простий критерій такої ефективності – ймовірність розкриття ключа або потужність множини ключів. Однак, даний критерій не враховує інших важливих вимог до криптосистем:

- неможливість розкриття або осмисленої модифікації інформації на основі аналізу її структури;
- досконалість використовуваних протоколів захисту;
- мінімальний обсяг використовуваної ключової інформації;
- мінімальна складність реалізації, її вартість;
- висока оперативність.

Бажане використання деяких інтегральних показників, що враховують зазначені фактори. Часто більш ефективним при виборі і оцінці криптографічної системи є використання експертних оцінок і імітаційне моделювання. У будь-якому разі обраний комплекс криптографічних методів повинен сполучити як зручність, гнучкість і оперативність використання, так і надійний захист від зловмисників циркулюючої в ІС інформації.

Література:

1. Галицкий А. В., Рябо С. Д., Шаньган В. Ф. Защита информации в сети – анализ технологий и синтез решений. – М.: ДМК Пресс, 2004. – 616 с.: ил.
2. Норткатт С., Новак Д., Маклахлен Д. Обнаружение вторжений в сеть. – М.: ЛОРИ, 2001. – 384 с.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ КРИПТОАЛГОРИТМОВ DES И АВТОРСКОГО АЛГОРИТМА ENIGMA-R10

Н.Е. Губенко^а, Д.В. Рябченко^б

г. Донецк, Донецкий национальный технический университет

^а gubenko@cs.dgtu.donetsk.ua

^б ryabchenko@maus.donetsk.ua

В настоящее время существует множество разновидностей криптоалгоритмов, которые условно можно разделить на симметричные и асимметричные. Традиционное шифрование, которое также называется симметричным, долгое время было единственным методом шифрования и, несмотря на открытие метода шифрования с открытым ключом, этот метод остается самым распространенным.

Классификация криптографических систем в общем случае строится на основе трех независимых характеристик.

1. Тип операций по преобразованию открытого текста в шифрованный.

Все алгоритмы шифрования основываются на использовании двух операций: *замены*, означающей замещение каждого элемента открытого текста некоторым другим элементом, и *перестановки*, означающей изменения порядка следования элементов открытого текста. При этом главным требованием оказывается отсутствие потерь информации (т.е. обратимость всех операций).

2. Число применяемых ключей.

Если и отправитель, и получатель используют один и тот же набор ключей, система называется симметричной или схемой традиционного шифрования. Если отправитель и получатель используют разные ключи, система называется асимметричной или схемой шифрования с открытым ключом.

3. Метод обработки открытого текста.

Блочное шифрование предполагает обработку открытого текста блоками, так что в результате обработки каждого блока получается блок шифрованного текста. *Поточное шифрование* подразумевает шифрование всех элементов открытого текста последовательно, одного за другим, в результате чего на каждом этапе получается по одному элементу шифрованного текста. Симметричные алгоритмы используют один и тот же ключ для прямого и обратного преобразований и требуют защищенный канал, для передачи ключа от отправителя шифрованной информации к получателю. Асимметричные алгоритмы (т.н. алгоритмы с открытым ключом) оперируют двумя однозначно взаимосвязанными ключами, один из которых используется для прямого преобразования информации (открытый ключ), а второй – для обратного (закрытый ключ). При этом задача нахождения закрытого ключа из открытого до сих пор не решена математически. Данный тип алгоритмов

выгодно отличается от симметричных отсутствием необходимости защищенной пересылки ключа получателю, но является значительно более трудоемким.

Асимметричные алгоритмы целесообразно применять при передаче данных от одного лица – другому. Если же, необходимо шифровать данные, не предназначенные для передачи другим лицам (например, личную или финансовую информацию), то целесообразно использовать симметричный алгоритм (т.к. отпадает необходимость передачи ключа). Более того, в связи с математической и, соответственно, аппаратной трудоемкостью шифрования с открытым ключом, для шифрования сообщений, имеет смысл использовать симметричный шифр со случайным ключом, а относительно короткий ключ шифровать, используя асимметричный алгоритм. Именно так работает популярная криптосистема для защиты электронной почты PGP (Pretty good privacy).

Разработанный алгоритм Enigma-R10 представляет собой симметричный криптоалгоритм, оперирующий с блоками данных размером 2^{10} байт = 1 килобайт. Данный алгоритм назван в память о первой шифровальной машине “Enigma”, разработанной и использовавшейся в Германии, во время Второй мировой войны. Данные шифруются 1-килобайтными блоками с использованием одного или двух ключей произвольной длины, вводимых пользователем. Тот же алгоритм с теми же ключами служит для обратного преобразования шифрованного текста в открытый. Ключи, вводимые пользователем, преобразуются таким образом, что их длина и содержимое меняются, в зависимости от первоначальных значений. Если пользователь вводит только один ключ, из него автоматически генерируется второй ключ.

На первом шаге данные преобразуются посредством циклического сдвига битов блока. Оба ключа представляются в виде массивов цифр, определяющих величину сдвига. Элементы первого массива циклически распределяются последовательно от начала блока к его концу. Элементы второго массива распределяются циклически, но скачкообразно, от начала к концу и обратно (метод «краевой атаки»), приближаясь к середине блока. Таким образом, посредством сочетания двух методов распределения элементов образуется неповторяющийся результирующий ключ, длиной равный блоку. Этот принцип повышает криптоустойчивость закрытого текста, т.к. из него нельзя определить длину исходных ключей. Затем биты каждого байта блока сдвигаются на величину, определяемую соответствующим элементом результирующего ключа. Направление сдвига зависит от того, в каком направлении работает алгоритм (шифрования или дешифрования блока).

На втором шаге к каждому байту блока применяется побитовая операция исключающего ИЛИ с каждым элементов массива первого ключа. Таким образом, достигается т.н. лавинный эффект, повышающий криптоустойчивость шифрованного текста. Лавинный эффект – высокая чувстви-

тельность результата к изменению начальных данных – любые малые изменения ключа приводят к значительным изменениям в результирующем тексте. В частности, изменения значения всего одного бита ключа отражается в изменении значений многих битов в зашифрованном тексте

Для определения эффективности алгоритма Enigma-R10 было проведено сравнение с известным алгоритмом DES (Data Encryption Standard) с длиной блока 64 бит и длиной ключа 56 бит. В результате изменения одного бита ключа, после 16 раундов DES выдает блок, отличающийся на 35 бит от первоначального, т.е. результат на 45,31% совпадает с оригиналом (см. [2, 109], таблица 3.5(б)). В аналогичной ситуации Enigma-R10 выдает результат, совпадающий с оригиналом на 0–12,2%. Таким образом, лавинный эффект Enigma-R10 почти в 4 раза больше, чем лавинный эффект DES.

Результаты дешифрования файла размером 254 075 байт, зашифрованного с ключом “aq123”, представлены в таблице 1.

Таблица 1. Лавинный эффект Enigma-R10

<i>Ключ</i>	<i>Идентичность</i>	<i>Ключ</i>	<i>Идентичность</i>	<i>Ключ</i>	<i>Идентичность</i>
aq120	0.00 %	aq12l	0.00 %	aq12G	0.00 %
aq121	0.00 %	aq12m	0.00 %	aq12H	0.33 %
aq122	0.00 %	aq12n	0.00 %	aq12I	0.00 %
aq123	100.00 %	aq12o	0.00 %	aq12J	0.00 %
aq124	0.31 %	aq12p	0.37 %	aq12K	0.00 %
aq125	0.00 %	aq12q	0.00 %	aq12L	0.00 %
aq126	0.00 %	aq12r	0.00 %	aq12M	0.27 %
aq127	0.00 %	aq12s	0.00 %	aq12N	0.00 %
aq128	0.00 %	aq12t	0.34 %	aq12O	0.00 %
aq129	0.25 %	aq12u	0.34 %	aq12P	0.00 %
aq12a	0.28 %	aq12v	0.00 %	aq12Q	0.01 %
aq12b	0.00 %	aq12w	0.00 %	aq12R	0.33 %
aq12c	0.00 %	aq12x	0.00 %	aq12S	0.00 %
aq12d	0.00 %	aq12y	12.20 %	aq12T	0.00 %
aq12e	0.15 %	aq12z	0.29 %	aq12U	0.00 %
aq12f	0.34 %	aq12A	0.00 %	aq12V	0.00 %
aq12g	0.00 %	aq12B	0.00 %	aq12W	0.29 %
aq12h	0.00 %	aq12C	0.30 %	aq12X	0.00 %
aq12i	0.00 %	aq12D	0.00 %	aq12Y	0.00 %
aq12j	0.00 %	aq12E	0.00 %	aq12Z	0.00 %
aq12k	0.32 %	aq12F	0.00 %	-	-

В связи с тем, что алгоритм DES использует 16 раундов, а Enigma-R10 – 2 раунда, время преобразования данных алгоритмом Enigma-R10 в 3-5 раз меньше, по сравнению с DES. Например, для ЭВМ с 256 Мб ОЗУ и ЦП

Athlon XP 2600+ Enigma-R10 преобразует 2000 килобайт в секунду, а DES – 480 килобайт в секунду. Для алгоритма DES, при длине ключа 56 бит имеется 2^{56} вариантов различных ключей, что приблизительно равно $7,2 \times 10^{16}$. Для алгоритма Enigma-R10 с пользовательским ключом “aq123” – преобразованный ключ имеет длину 15 байт = 96 бит. Соответственно существует 2^{96} вариантов различных ключей, что приблизительно равно $7,9 \times 10^{28}$. По сравнению с современными блочными алгоритмами, такими как 3DES (Tripple DES), Enigma-R10 менее криптоустойчив. Но он может быть применен для защиты личных данных, которые вряд ли будут интересовать спецслужбы, имеющие в своём распоряжении вычислительные и интеллектуальные ресурсы для взлома алгоритмов данного класса.

Литература:

1. Аршинов М.Н., Садовский Л.Е. Коды и математика. – М.: Наука, 1983. – 144 с.
2. Столлингс Вильям. Криптография и защита сетей: принципы и практика, 2-е изд. Пер. с англ. – М.: Издательский дом “Вильямс”, 2001. – 672 с.

ХЭШ-ФУНКЦИЯ НА ОСНОВЕ ДВОИЧНОГО ДЕРЕВА

И.А. Кривель¹, А.Н. Моргун²

¹ г. Черкассы, Факультет рыночных, информационных и инновационных технологий Киевского национального университета технологий и дизайна

² г. Черкассы, Черкасский институт пожарной безопасности имени Героев Чернобыля
a_n_m@rambler.ru

В соответствии с [1], хэш-функцией $h(m)$ в криптографии называется некоторый алгоритм, в результате применения которого исходное сообщение m произвольной длины преобразуется в последовательность фиксированного числа байт. Одно из основных применений хэш-функции состоит в образовании сжатого образа данного сообщения, используемого для его цифровой подписи.

Хэш-функция должна быть стойкой как в смысле её обращения, так и в смысле возникновения коллизий при её вычислении. Первое из указанных требований означает, что по известному значению хэш-функции $h(m)$ должно быть невозможно (достаточно сложно) определить значение её аргумента m . В соответствии со вторым требованием, для данного аргумента m должно быть невозможно (достаточно маловероятно) найти другой аргумент m_1 такой, что $h(m) = h(m_1)$.

Интересные возможности построения хэш-функции для числовых данных предоставляет двоичное дерево Штерна-Броко [2], содержащее в качестве вершин рациональные числа в виде неотрицательных несократимых дробей.

Построению двоичного дерева Штерна-Броко (дерево SHB) предшествует получение последовательности Штерна-Броко. Рассмотрим стартовую последовательность двух дробей $0/1$ и $1/0$. Новую дробь $1/1$ образуем, сложив числители и знаменатели стартовых. В новой последовательности $0/1$, $1/1$, $1/0$ рассматриваем все соседние пары и вставляем в неё очередные дроби по тому же принципу. Получаем последовательность $0/1$, $1/2$, $1/1$, $2/1$, $1/0$. На следующем этапе последовательность приобретает вид $0/1$, $1/3$, $1/2$, $2/3$, $1/1$, $3/2$, $2/1$, $3/1$, $1/0$. Описанный процесс можно продолжать бесконечно.

Дроби последовательности можно расположить на двоичном дереве SHB, которое для удобства обозрения представлено в виде нижеследующей таблицы. При этом вершину $1/1$ будем считать корнем.

Основное свойство дерева SHB состоит в том, что в нём представлены все рациональные числа, причём каждое – только один раз. Пронумеруем вершины дерева, начиная от корня и обозначая нулями перемещения по левым ветвям, а единицами – по правым. Например, дробь $2/3$ будет иметь двоичный номер 01 , а дробь $3/1$ – двоичный номер 11 , дробь $2/5$ – двоичный

номер 001, дробь $5/3$ – двоичный номер 101 и т.д. Таким образом, мы получили возможность сопоставлять двум числам (числителю и знаменателю дроби) одно число – номер дроби.

0/1															1/0
							1/1								
			1/2							2/1					
		1/3				2/3				3/2				3/1	
	1/4		2/5		3/5		3/4		4/3		5/3		5/2		4/1

Одно из свойств такого сопоставления состоит в том, что номера некоторых дробей имеют в своём составе ведущие нули. В результате этого численно равные номера могут соответствовать разным дробям. Например, номера дробей $2/3$ и $2/5$ имеют численно равные представления 01 и 001 соответственно, дроби $3/1$ и $3/4$ также имеют численно равные номера 11 и 011. Благодаря этому становится невозможным однозначное обратное сопоставление и таким образом обеспечивается требование стойкости хэш-функции в смысле её обращения.

В приведённых примерах обращает на себя внимание то, что дробь не имеет в составе своего двоичного номера ведущих нулей тогда, когда её числитель больше знаменателя, то есть дробь – неправильная. Все неправильные дроби содержатся только в правом поддереве дерева SHB.

Ниже представлены программы на языке Паскаль, позволяющие осуществить взаимное преобразование “дробь – двоичный номер”.

Листинг 1. Поиск дроби на дереве SHB по её двоичному номеру

```
{ $B+, D+, E+, I+, L+, N+, Q+, R+, X- }
Program Drob_Derevo_SHB;
  Procedure Drob_SHB(S: String; Var m,n: LongInt);
    Var m1,n1,i: LongInt;
  Begin
    m:=0; n:=1; m1:=1; n1:=0;
    For i:=1 To Length(S) Do Begin
      If S[i]='0' Then Begin m1:=m+m1; n1:=n+n1 End;
      If S[i]='1' Then Begin m:=m+m1; n:=n+n1 End End;
    m:=m+m1; n:=n+n1
  End;
  Var
    S: String; m,n: LongInt;
Begin
  WriteLn('Введите двоичный номер дроби S=');
  ReadLn(S);
  Drob_SHB(S,m,n);
  WriteLn('m/n = ',m,' / ',n);
  ReadLn
End.
```

Листинг 2. Поиск двоичного номера номера заданной дроби на дереве SHB

```
{ $B+, D+, E+, I+, L+, N+, Q+, R+, X- }
```

```

Program Nomer_Derevo_SHB;
  Procedure Nomer_SHB(m,n: LongInt; Var S: String);
  Begin
    S:='';
    While m<>n Do
      If m>n Then Begin S:=S+'1'; m:=m-n End
      Else Begin S:=S+'0'; n:=n-m End
    End;
  Var
    S: String; m,n: LongInt;
Begin
  Write('Введите m и n: '); ReadLn(m,n);
  Nomer_SHB(m,n,S);
  WriteLn('Двоичный номер S=',S);
  ReadLn
End.

```

В нижеследующей таблице представлены примеры неправильных дробей, для которых получены их двоичные номера, преобразованные затем к десятичной системе счисления.

Дробь	Номер дроби	Десятичное представление номера
3/1	11	3
5/3	101	5
7/3	1100	12
11/7	10100	20
14/5	110111	55

Данная таблица показывает, как двум десятичным числам (числителю и знаменателю дроби) сопоставляется одно десятичное число (номер дроби в десятичном представлении). При этом суммарное количество разрядов двоичных представлений числителя и знаменателя оказывается больше количества разрядов двоичного номера. Например, компоненты дроби 11/7 имеют двоичные представления 1011 и 111 соответственно, что составляет в сумме 7 двоичных разрядов. А вот двоичное представление номера этой дроби 10100 содержит всего 5 двоичных разрядов.

На возможности сопоставить двум десятичным числам одно с меньшим количеством разрядов двоичного представления и базируется алгоритм хэш-функции на основе двоичного дерева Штерна-Броко.

Некоторое сообщение представляется в виде последовательности целых десятичных чисел. Теоретически это всегда возможно, если использовать, например, таблицу кодов ASCII. Каждая пара соседних чисел, рассматриваемая как числитель и знаменатель некоторой дроби, заменяется одним числом. Во вновь образованной последовательности чисел каждая пара соседей снова рассматривается как дробь и также заменяется одним числом. Процесс продолжается до тех пор, пока таким образом не будет получено единственное число с нужным числом разрядов. Именно это число и рассматривается как значение хэш-функции данного сообщения.

Пример применения рассмотренной хэш-функции приведён в таблице, каждая строка которой представляет собой постепенно сжимаемое исходное сообщение, содержащее изначально 16 чисел.

7	4	5	4	5	3	3	1	8	5	4	1	4	1	3	1
11		8		5		3		10		7		7		3	
18				5				19				12			
58								41							
307															

Каждая пара соседних чисел некоторого исходного сообщения (первая строка таблицы) рассматривается как дробь, которой сопоставляется её двоичный номер. Полученный номер преобразуется в десятичное число. Процесс продолжается до получения единственного числа, которое следует рассматривать как значение хэш-функции данного сообщения. Например, воспользовавшись второй из приведённых программ, для дроби $7/4$ получим двоичный номер 1011, что соответствует десятичному числу 11. Для следующей дроби $5/4$ двоичный номер равен 1000, что соответствует десятичному числу 8. В свою очередь, для дроби $11/8$ получим номер 10010, что равно 18, и т.д. В последней строке таблицы находится значение хэш-функции, равное десятичному числу 307.

Эффект сжатия, обеспечиваемый рассмотренной хэш-функцией, оценивается следующим образом. Подсчитав количество конкретных двоичных разрядов, необходимых для представления каждого из чисел данного сообщения, получим минимально возможное значение 38. Подсчёт по максимуму предполагает использование одинакового количества разрядов для каждого числа сообщения. Поскольку наибольшее число сообщения, равное 8, требует для своего представления четырёх двоичных разрядов, то подсчёт по максимуму даёт значение 64. Сжатое сообщение 307 в двоичной форме выглядит как 100110011 и, таким образом, занимает всего лишь 9 двоичных разрядов.

Литература:

1. Гундарь К.Ю., Гундарь А.Ю., Янишевский Д.А. Защита информации в компьютерных системах. – К.: Корнейчук, 2000.
2. Грэхем Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики. – М.: Мир, 1998.

ДОСЛІДЖЕННЯ МОЖЛИВОСТІ ВІДТВОРЕННЯ ВИКОНУВАНИХ ФАЙЛІВ ПЕРСОНАЛЬНИХ ЕОМ

О.О. Гайша

м. Миколаїв, Національний університет кораблебудування
імені адмірала Макарова
physics2005@mail.ru

Проблема створення систем захисту програмного забезпечення (ПЗ) від несанкціонованого копіювання є актуальною і ключовою у боротьбі з комп'ютерним піратством [1]. Усі системи захисту, що претендують на реальне практичне застосування, повинні мати зашифровані ділянки коду. Спочатку доведемо це твердження, а потім проаналізуємо можливість збору і відтворення коду, що є у файлі зашифрованим.

Найперше вихідне твердження для доказу полягає в тому, що у незаконного користувача є копія ПЗ, у якій присутні всі програмні коди, що відповідають повнофункціональній версії (можливо, деякі з них відключені, але все ж присутні у виконуваному файлі). Це пояснюється тим, що після продажу першої легальної (отже, повної) копії, покупець може її розтиражувати. Таким чином, у систему захисту слід вбудовувати якісь заходи, що протидіятимуть використанню однієї і тієї ж копії ПЗ різними користувачами на різних машинах.

Незалежно від використаного алгоритму запобігання несанкціонованому тиражуванню ПЗ, кінцево такі заходи зводяться до одного підсумкового умовного оператора (команди асемблера), що видає вирок: дана запущена копія працює законно чи ні. Місце такої перевірки в загальному випадку знаходиться зломщиком через невеликий проміжок часу, і може бути скориговане на логічно протилежну дію. Відбувається модифікація і реалізується «фізичний» злом ПЗ, докладніше про це див. [2].

Уникнути такого злomu можливо лише завдяки закриттю окремих ділянок коду програми, які відповідають системі захисту. Без використання якихось апаратних засобів таке закриття можна організувати лише шляхом шифрування коду програми. Таким чином, твердження про необхідність шифрування критичних ділянок виконуваних файлів доведено.

При виконанні програми зашифровані ділянки файлів для перетворення у змістовний код повинні бути розшифровані, причому, оскільки процес відбувається на комп'ютері користувача, алгоритм та необхідні дані для такого перетворення є потенційно відомим зловмиснику. Якщо застосувати симетричний криптоалгоритм, то зломщик може розшифрувати закриті ділянки, виправити необхідні йому команди, та знову зашифрувати уже виправлену версію програми. Після таких операцій програма працюватиме за тими ж принципами, що передбачив розробник системи захисту (тобто розшифровуватиме необхідні ділянки коду, і т.д.), але видаватиме невірний

висновок щодо законності використання даної копії.

Якщо ж застосувати якийсь асиметричний алгоритм, то розшифровуватися закриті ділянки будуть відкритим ключем, отже будь-хто зможе їх прочитати. Але, зважаючи на секретність другого ключа шифрування, ніхто, окрім виробника ПЗ, не зможе виправити код у зашифрованих ділянках вихідного файлу.

В загальному випадку така схема захисту могла б бути «ідеально» стійкою, тобто її стійкість дорівнювала б стійкості застосованого двохключового алгоритму. Але існує ще одна принципова можливість злому системи захисту, що полягає у зборі виконуваних команд з оперативної пам'яті ПК. Це пов'язано з важливою особливістю сучасних процесорів архітектури фон Неймана, які потребують розміщення коду їх програм перед виконанням у оперативній пам'яті (причому у відкритому, придатному до виконання виді). Таким чином, якщо навіть у файлі присутня ділянка із зашифрованим кодом, перед її виконанням процесором вона повинна бути розшифрована у деякій області пам'яті (нажаль, розшифровуватися у самому процесорі, поза областю доступності інших процесів, код не може).

Вказана можливість реалізується завдяки можливостям архітектури процесора P6, розробленої корпорацією Intel. Наявність спеціального режиму роботи процесора типу System Management Mode (DEBUG), що використовується для покрокового виконання програми, дозволяє зупинити її виконання у потрібному місці, і проаналізувати внутрішній стан таких компонентів ПК, як реєстри процесора та оперативна пам'ять. При цьому відбуваються в цілому несанкціоновані дії, адже одна прикладна програма (налагоджувач) отримує доступ до адресного простору іншого процесу, що має бути неприпустимим без відповідної санкції.

Таким чином, виникає наступна загроза. Оскільки замінити у вихідному файлі зашифровані ділянки зломщик не може, йому залишається спробувати відтворити змістовні оператори у новому виконуваному файлі без системи захисту. Отже, вихідний файл аналізується, зашифровані ділянки, що відповідають системі захисту відкидаються, а з необхідних операторів, що відповідають за функціональність ПЗ, формується новий виконуваний файл. Необхідні поля заголовків та загальний опис внутрішньої структури формуються самостійно, згідно до формату, притаманного даній платформі. В ОС Windows, наприклад, таким форматом є Portable Executable (PE) файли.

Якщо ж у виконуваному файлі буде зашифрованою не лише сама система захисту, а й деяка частина функціональних операторів, то тут можливі два варіанти. По-перше, можна розшифрувати зашифровані функціональні оператори, читаючи відповідні області прямо з файлу (бо алгоритм та ключ є потенційно відомими). По-друге, можна використати спеціальний режим роботи процесора DEBUG, дозволивши системі захисту самій розшифровувати код із файлу та розміщувати його у оперативній пам'яті, а потім зчитуючи і збираючи розшифрований код звідти.

Ще одним можливим типом захисту ПЗ від модифікацій є вміщення всіх, або частини програмних кодів у недорогий зовнішній запам'ятовуючий пристрій (наприклад, HASP). Внутрішній вміст такого пристрою є недоступним для модифікацій, але, зважаючи на вказану особливість сучасних процесорів, що не можуть розшифровувати код «власне у собі», ці команди мають бути попередньо розміщені в оперативній пам'яті ПК. Тут стає принциповою можливість доступу одного процесу до пам'яті, що належить іншому процесу. Таким чином, будь-які виконувані команди, які зберігаються в HASP'і можна зібрати, проаналізувати і відтворити у програмному емуляторі апаратного пристрою (які на сьогоднішній день є єдиним засобом для злому захищених апаратно програм).

Можливі і інші види захисту (наприклад, шифрування всього тіла даної копії програми розміщенням збійних секторів на гнучкому диску або CD-ROM), які на перший погляд не дозволяють використати дану копію без санкції виробника. Але використання даної копії і не обов'язкове – можливе відтворення усіх функціональних блоків коду у окремому виконуваному файлі, що збирається по частинах вручну (тому, звичайно, в ньому відсутні будь-які обмеження).

Звичайно, така методика є одним із найскладніших методів злому, до якого можна вдатися, лише якщо інші більш прості не дають відповідного результату, але принциповість такої можливості показує, що ідеального захисту від несанкціонованого копіювання програмного забезпечення створити *неможливо* (під ідеальним мається на увазі захист, стійкість якого рівна стійкості застосованих криптоалгоритмів).

Отже, побудувати систему захисту типу Кірхгофа (наприклад, шляхом запровадження якогось нового криптографічного протоколу чи застосування взаємодії з якимсь апаратним засобом) неможливо. Формалізація задачі не може забезпечити стійкий захист від відтворення коду програмного забезпечення. Це пояснюється тим, що незалежно від створеної системи захисту, програму все ж може бути вкрадено шляхом механічного збору усіх функціональних ділянок коду даної програми при повному їх переборі та виконанні.

Практично описаний тип злому реалізується найпростіше у консольних пакетних утилітах, які мають лінійну послідовність виконання, але звичайно ці програми не є комерційними і їх злом малоімовірний в цілому. Платне ж програмне забезпечення звичайно має гарний графічний інтерфейс, отже є таким, що управляється подіями. Відповідно програма не є лінійно виконуваною і окремі її функціональні гілки (можливо зашифровані) запускаються лише після приходу відповідних їм повідомлень. Повідомлення, які залежать від користувача, типу WM_COMMAND, WM_USER, WM_CHAR, та ін., можуть надсилатися зломщиком безпосередньо, отже і розшифровка та вміщення у оперативну пам'ять відповідних гілок відбувається негайно. Отже, зашифровані гілки коду, що є реакцією програми на повідомлення

про події користувача, легко відстежуються і можуть бути одразу зібрані.

Дещо іншою виглядає ситуація, при якій програма реагує на повідомлення системи або якісь внутрішні «події» (що не є подіями операційної системи, а лише характеризують досягнення програмою якогось заданого стану). Ці явища в цілому є випадковими для зломщика (тому що логічний аналіз усього асемблерного лістингу програми є все ж нереальним), тому можливо їх буде важко відстежити. Тоді гілки, що відповідають цим подіям відповідно важко локалізуються у розшифрованому вигляді у оперативній пам'яті.

Отже, можливим методом захисту від механічного збору коду для відтворення виконуваного файлу є запуск окремих функціональних операцій по випадковим внутрішнім подіям. Наприклад, користувач хоче запустити на виконання процедуру обробки фільтром якогось зображення. Замість негайного переходу до відповідної гілки, програма (можливо в окремому потоці, чи навіть процесі) запускає додаткову маскувальну процедуру, що виконує якісь незмістовні дії, а через деякий випадковий інтервал часу запускає функціональну гілку.

Таким чином, практично у власному програмному забезпеченні виробники можуть лише збільшувати трудовитрати зломщика шляхом включення до *обов'язкової* загальної системи захисту від тиражування та модифікацій ще й блоків, які ускладнюватимуть роботу програми, запускаючи окремі гілки не одразу за повідомленнями користувача, а за якимись внутрішніми власними алгоритмами (через невеликий проміжок часу).

Ще одним серйозним ускладненням для довготривалого процесу збору усіх кодів є випадок запуску повнофункціональної програми, що працює деякий (порядку десятків секунд) проміжок часу, а лише потім перевіряє свою законність за якоюсь випадковою подією (причому не обов'язково внутрішньою). Наприклад, при старті запускається окремий потік, який деякий випадковий час «спить», а потім запускає систему захисту. Або, наприклад, перевірка законності роботи текстового редактора відбувається лише тоді, коли користувач набирає на клавіатурі якусь випадкову послідовність символів. Або користувач пересунув мишу з заданою швидкістю. Усі ці випадки об'єднує те, що вони є нібито випадковими для зломщика, тому йому важко буде локалізувати механізми захисту, що весь час відключатимуть програму, не даючи проводити збір коду.

В-цілому ж слід констатувати, що можливість відтворення змісту виконуваних файлів (із зашифрованими ділянками), принципово присутня і базується на необхідності розшифровки перед виконанням процесором зашифрованого коду і розміщення його у оперативній пам'яті. Така можливість не дозволяє побудувати формальну модель системи захисту, стійкість якої буде рівною стійкості застосованих криптографічних алгоритмів, і змушує програмістів застосовувати, на перший погляд, непрофесійні прийоми закриття внутрішньої структури програми та утримування секретності алго-

ритмів захисту [3].

Література:

1. Гальчевський Ю.Л., Гайша О.О. Аналіз можливих шляхів боротьби з комп'ютерним піратством // Захист інформації: Сборник научних трудов. – К.: НАУ, 2004. – С. 169-175.

2. Гальчевський Ю.Л., Гайша О.О. «Логічні» та «фізичні» захисти програмного забезпечення від несанкціонованого копіювання // Захист інформації: Науково-технічний журнал. – 2005. – №2(23). – С. 34-40.

3. Гайша О.О. Секретність ключа чи алгоритму? // Електротехніка і електромеханіка: Матер. міжн. наук.-техн. конф. студ., аспір. і молод. наук. робітн. – Миколаїв: НУК, 2005. – С. 146-149.

МЕТОДИ ЗАХИСТУ ПРОЦЕСУ ПЕРЕДАЧІ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ ТА МЕРЕЖАХ

Ю.В. Хмельницький

м. Хмельницький, Хмельницький національний університет

cheshun_v@mail.ru

Вступ. В сучасних умовах комп'ютерні технології використовуються у всіх сферах життєдіяльності. Як засіб передачі інформації в цих технологіях використовують комп'ютерні системи та мережі. Широка розгалуженість, складність систем створюють проблему забезпечення надійності та вірогідності їх функціонування. Для забезпечення ефективної роботи систем та мереж використовують системи пошуку та ідентифікації несправностей.

До особливостей комп'ютерних систем та мереж віднесемо:

- неоднозначність інформації;
- наявність в компонентах мережі засобів вбудованого контролю;
- наявність несправностей визначеного типу в залежності від видів використовуваних протоколів і конфігурації мережі;
- значна схильність до перекручування програмного забезпечення і діагностичної інформації від “вірусів”;
- залежність прояву несправностей від щільності мережного трафіка і кількості підключених абонентів.

Для забезпечення більш ефективної та достовірної передачі інформації в комп'ютерних мережах слід провести аналіз типів перекручувань інформації та методи захисту процесу передачі і використовувати систему, яка може провести перевірку з урахуванням особливостей функціонування.

Аналіз існуючих методів. Для ефективної подачі тестових впливів система має працювати в реальному масштабі часу, тому необхідно оперативне виявлення перекручувань інформації та її наслідків, а також оперативне й автоматичне вживання заходів щодо ліквідації чи зменшенню можливих відхилень процесу від нормального режиму без його зупинки чи тривалого переривання. При цьому повинна враховуватися тривалість прояву наслідків перекручування в результатах функціонування системи і застосовуватися корегування ходу процесу, що забезпечує максимальне скорочення тривалості прояву цих наслідків. Для забезпечення захисту діагностичного процесу й інформації використовується інформаційна і часова надмірність. Тимчасова надмірність чи резерв часу використовується для контролю і виявлення перекручувань, на його діагностику й ухвалення рішення по відновленню діагностичного процесу і на реалізацію операцій відновлення. Виявлені перекручування за їхніми наслідками можна розподілити на три групи: не знецінюють, що частково знецінюють і цілком знецінюють всі отримані результати [1]. Якщо після відновлення процесу його можна продовжити без повторення з місця, де виявлене перекручування, то така від-

мова не знецінює результат. При відмові, що цілком знецінює, необхідно повторювати всі операції, виконані до моменту відмови. У проміжному випадку при відмові, що частково знецінює результат, проте зберігає цінність, деякі проміжні результати, що звичайно відповідають моменту попереднього контролю працездатності можуть використовуватись для аналізу.

Типи перекручувань інформації і методи захисту процесу передачі.

Незалежно від джерел, будь-які перекручування зрештою виявляються в результуючих даних системи. Тому у всіх випадках критерієм якості передачі є вірогідність і точність обробленої інформації і виданих тестових впливів. Розподіл методів захисту зводиться до захисту процесу і до захисту інформації. При цьому методи контролю і захисту вихідних даних, що складають інформацію і масиви накопичених оброблених даних можуть розглядатися спільно. Інша група методів захисту спрямована на зниження наслідків помилок у тестах. Проведемо класифікацію наслідків перекручувань інформації й процесу і хоча б наближену оцінку їхнього впливу на загальний критерій функціонування системи по аналогії [1]. З погляду побудови засобів захисту і розподіл ресурсів системи, які доцільно виділити захист від перекручувань, помилки результатів можна розділити по їх наслідках на три типи [1]. До першого типу належать:

- перекручування результатів, що приводять до припинення виконання системою основних функцій чи їхньої частини на тривалий чи невизначений час (цілком знецінюють);
- перекручування, що короткочасно, але значно спотворюють величину чи значеннєвий зміст окремих результатів (частково знецінюють);
- перекручування, що мало і короткочасно впливають на результати, видавані системою керуючих алгоритмів (не знецінюють).

Такий якісний розподіл перекручувань визначає розподіл ресурсів по захисту. Найбільші ресурси доцільно виділяти для захисту від перекручувань інформації й процесу першого типу, наслідки яких можуть виявлятися в системі у наступному виді:

- зациклення, тобто послідовна повторювана реалізація визначеної групи тестів, що не припиняється без зовнішнього втручання;
- зупинка і припинення рішення задач;
- перекручування процесів взаємного переривання тестів, що приводить до блокування можливості деяких типів переривань;
- припинення чи значне зниження темпу рішення деяких задач унаслідок перевантаження системи по пропускній здатності;
- значне перекручування чи втрата накопиченої інформації про поточний стан процесу передачі.

Перекручування інформації й процесу передачі другого типу також дуже небезпечні і для захисту від них варто застосовувати досить ефективні заходи. Ці перекручування можуть виявлятися в наступному виді:

- пропуск тестів чи їхніх істотних частин;

- вихід на тест чи їхні частини, що різко спотворюють результати;
- обробка помилкових чи сильно переключених повідомлень.

Третій тип переключувань характерний, в основному, для квазібезперервних величин. Ці помилки мало спотворюють загальні результати, однак окремі викиди можуть сильно впливати на процес і потрібен досить ефективний захист від рідких значних відхилень результатів.

Перераховані типи переключувань істотно розрізняються не тільки по величині зміни результатів, але і по тривалості прояву їхніх наслідків. Методи, застосовувані для ліквідації наслідків помилок, також розрізняються по тривалості їхньої дії. У залежності від ступеня прояву і причин виявлених переключувань, можливі наступні оперативні міри для ліквідації їхніх наслідків:

- відновлення інформації і збереження стійкості процесів;
- ігнорування виявленого переключування внаслідок його слабого впливу на весь процес і на вихідні результати;
- повторення функціонального алгоритму при тих же вихідних даних;
- виключення тесту з обробки внаслідок переключеності чи труднощів відновлення процесу;
- короткочасне припинення рішення задач функціонального алгоритму до відновлення вихідних даних;
- перебудова режиму роботи алгоритму для зниження впливу переважання в зв'язку з втратою інформації про хід процесу.

Ці методи можуть істотно розрізнятися не тільки по величині переключувань результатів, що виправляються, але і по тривалості прояву наслідків помилки. Застосування цих методів вимагає аналізу характеру прояву помилки, стану інформації про хід процесу передачі, інформації в повідомленнях, що надходять і т.д. При цьому схема прийняття рішень на застосування методів захисту повинна забезпечувати оперативне нагромадження й індикацію виявлених переключувань, а також довгострокове нагромадження результатів контролю для більш детального аналізу джерела помилок і виявлення можливості їхньої локалізації й усунення шляхом коректування тесту чи виправлення роботи системи. Оперативний захист повинний також передбачати реєстрацію й індикацію переключувань для аналізу. Помилки, що виявляються систематично і досить часто, варто аналізувати на можливість виправлення тесту.

Існує досить багато методів захисту процесу передачі інформації [2]. Розглянемо захист від за циклювання тестів. У тестах широко використовуються однотипні операції, що утворюють цикли для пошуку, упорядкування й однотипного перетворення інформації. Причиною зациклення можуть бути не тільки помилки в тесті й переключування вихідної інформації, але і збої в апаратурі. Тому при виявленні первинного зациклення доцільно повторити включення тестів при тих же вихідних даних. Якщо зациклення не повторюється, то, швидше за все, воно відбулося в результаті випадкового збою.

Повторне зациклення при однакових вихідних даних може бути обумовлено помилкою в тесті при правильній вихідній інформації, перекручуванням вихідної чи інформації частковим відмовленням апаратури. При багаторазових зацикленнях з різними вихідними даними причиною є часткове відмовлення в апаратурі чи перекручування інформації про процес передачі.

Захист від зупинок методично близька до захисту від зациклення. У цьому випадку також припиняється рішення тестових задач і система зупиняється на деякій довільній команді. Зупинка може відбутися або через помилку при формуванні тесту, або через помилку в тесті, що приводить до виходу на ділянку тесту, що містить команду зупинка. Автоматичне виявлення зупинок аналогічно виявленню зациклення, наприклад, періодичним порівнянням часу завершення підпрограми тесту чи функціонального алгоритму з поточним часом лічильника в систему. При виявленні зупинок логіка відновлення функціонування аналогічна логіці захисту від зациклення, тобто послідовно перевіряється можливість повторення даного алгоритму, можливість його пропуску і необхідність переходу на справну систему.

Захист від перекручувань взаємного переривання підпрограм, що приводять до блокування можливості деяких типів переривань, а отже, до блокування прийому і видачі інформації, здійснюється, в основному, апаратними методами. При цьому контролюється правильність виконання операцій переривання, переходу до програми, що перериває, і наступного повернення до перерваної програми. Для захисту від таких помилок, а також від апаратних збоїв при перериваннях повинний передбачатися контроль виконання переривань і періодичний контроль взаємодії. Для виконання цих функцій спеціальна періодична програма повинна підготовляти контрольні тести і видавати. При порушенні періодичного обміну, що може бути наслідком як тестової помилки, так і безлічі інших причин, порушення обміну усувається позапрограмами методами.

Захист від помилок, що приводять до пропуску тесту чи їхніх частин, виробляється, в основному, методом контролю ключових кодів, що визначають перелік підпрограм, що повинні бути включені. Захист від пропуску підпрограм тестів застосовується переважно при включенні періодичних підпрограм, а також для окремих функціональних тестів, де порушення послідовності їхньої роботи може істотно відбитися на функціонуванні всієї системи.

Захист від помилок, що приводять до виходу на тести чи їхні частини, що різко спотворюють результати, методично близький до захисту від пропуску підпрограм тестів. Однак у даному випадку може бути більше варіантів, тому що внаслідок помилкових сполучень чи формувань команд у принципі можливий помилковий перехід, з будь-якої підпрограми на будь-яку іншу підпрограму тесту. Тому цілком захиститися від помилкових підключень підпрограм неможливо, але деякі особливо небезпечні, неприпустимі сполучення підпрограм іноді має сенс заблокувати.

Захист від перевантаження системи по пропускній здатності припускає виявлення і зниження впливу наслідків швидкого алгоритмічного розподілу ресурсів, обумовлених неправильним визначенням необхідної пропускної здатності системи для роботи в реальному масштабі часу. Крім цього, перевантаження можуть бути наслідком неправильного функціонування джерел інформації і перевищення нормального (розрахункового) рівня інтенсивності потоків повідомлень. Для виявлення перевантаження по пропускній здатності можна використовувати:

- контроль тривалості збереження повідомлень, що підлягають обробці по нижчих пріоритетах;
- контроль заповнення буферних нагромаджувачів повідомленнями низьких пріоритетів;
- контроль часу останнього включення диспетчером алгоритму нижчого пріоритету при наявності заявок на включення;
- контроль частоти включення самого диспетчера з повним послідовним аналізом усіх пріоритетних рівнів без рішення задач.

При виявленні перевантаження бажана також оцінка її величини і причини появи. Контроль завантаження буферних нагромаджувачів організується дуже просто шляхом перевірки записуваних адрес, підлягаючих обробці повідомлень і корисний не тільки при прийомі, але й при видачі повідомлень. Це дозволяє, зокрема, виявляти помилки в тестах, що ведуть до порушення нормального темпу формування тестів.

Застосовуючи алгоритмічні методи захисту, можна істотно знизити шкідливий вплив перевантажень ресурсів системи і адаптацію алгоритмів на рівень припустимого завантаження [2]. Зокрема, така адаптація істотно знижує вплив помилок у структурі побудови алгоритмів, що ведуть до невеликих короткочасних перевантажень.

Висновки. Таким чином, оперативний захист від перекручувань інформації й обчислювального процесу може використовуватися як засіб виявлення помилок, що складно виявляються. Це особливо необхідно на етапах передачі інформації й у процесі експлуатації даної системи. Головна задача забезпечення стійкості та оперативного захисту від різних перекручувань складається в забезпеченні безперервності процесу керування системою при припустимих помилках у вихідних повідомленнях.

Література:

1. Липаев В.В. Проектирование математического обеспечения АСУ. М.: Советское радио, 1977. – 400 с.
2. Хмельницький Ю.В. Метод адаптивного діагностування комп'ютерних мереж // Вісник ТУП. – Хмельницький, 2003. – №3. – С. 43-48.

ОРГАНІЗАЦІЯ ЗАХИСТУ WEB-РЕСУРСІВ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ ASP.NET

А.А. Кулик

м. Хмельницький, Хмельницький національний університет
cheshun_v@mail.ru

На сучасному етапі розвитку інформаційних технологій багато уваги приділяється розподіленим системам обробки інформації. До таких систем відносяться будь-які комплекси, які об'єднуються у мережу для отримання та надання певних необхідних сервісів. Такі комплекси можуть бути як локальними, так і глобальними. Локальні комплекси (або локальні мережі) сьогодні дуже широко розповсюджені і використовуються у багатьох сферах. Вони часто надають лише деякий спектр послуг обмеженому колу користувачів. Глобальні мережі (наприклад, Internet) призначені для загального користування будь-якими абонентами і надання загально необхідних сервісів (електронна комерція, доступ до інформаційних ресурсів, розважальні ресурси тощо). На практиці часто виникає ситуація, коли певні інформаційні ресурси повинні бути доступними з будь-якої точки підключення, але лише конкретно визначеним особам. Наприклад, комерційні ресурси, до яких можуть отримати доступ лише авторизовані користувачі. Такі ресурси повинні використовувати механізми захисту, які забезпечать необхідний рівень безпеки інформації, що міститься на цих ресурсах. Компанія Microsoft створила спеціальну технологію ASP.NET, яка використовується для побудови ієрархічних Internet-ресурсів і включає в себе спеціальні механізми захисту інформації. Далі ми розглянемо такі механізми забезпечення захисту, як: захист даних в операційній системі Windows, існуючі форми ідентифікації користувачів, виконання контролю доступу до ресурсів web-серверів.

Використання ASP.NET надає функціональність лише у зв'язці із web-сервером IIS (Internet Information Server) фірми Microsoft. Захист інформації в такій системі проходить 2 етапи. Загальна структура цих етапів зображена на рис. 1 [2]. Захист даних може виконуватись за допомогою двох основних методів. Перший метод полягає у використанні спеціальних механізмів захисту, вбудованих в операційну систему Windows. Другий метод полягає у безпосередньому використанні засобів ASP.NET. Перший метод є більш простим при створенні web-ресурсів, оскільки вимагає менших затрат на програмування логіки web-сторінок. При цьому даний метод забезпечує лише необхідний мінімум для захисту інформації на web-сервері. Отже, даний підхід рекомендується використовувати лише при захисті некритичної інформації. Другий метод базується на системі ASP.NET, а отже дозволяє використовувати будь-які способи ідентифікації та авторизації користувачів: починаючи від зберігання інформації про реєстрацію в звичайному

файлі до використання механізмів баз даних та шифрування. Звичайно, такий підхід вимагає знань у відповідних областях захисту інформації та витрат часу на програмування спеціальної логіки додатків, що залежить від конкретних потреб [1].

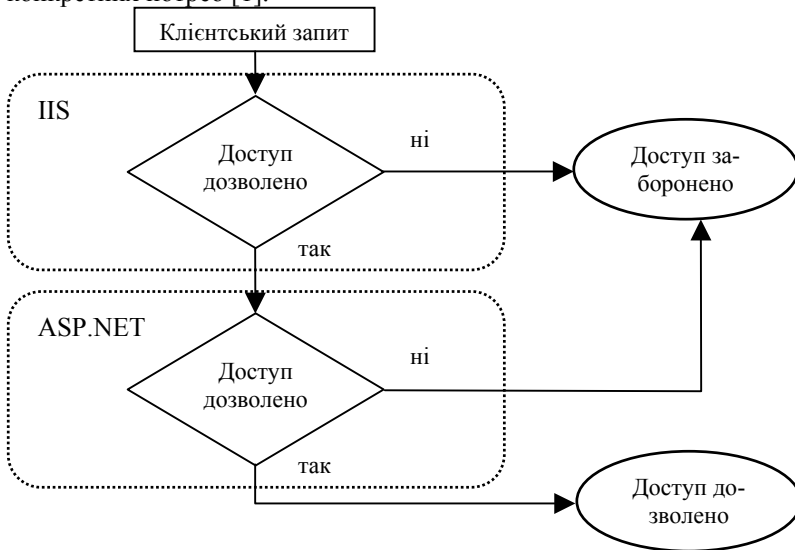


Рис. 1. Етапи захисту інформації в системі IIS та ASP.NET

Процес доступу до захищених даних складається з наступних етапів: ідентифікація користувача, авторизація користувача, імперсоналізація та делегування. Розглянемо детально кожен із цих етапів та їх реалізацію в ASP.NET.

Ідентифікація – це процес визначення користувача, який отримує доступ до інформаційного ресурсу. Ідентифікація здійснюється за допомогою списків облікових записів користувачів або інших подібних механізмів. Етап ідентифікації не є обов'язковим у процесі отримання доступу до веб-ресурсу. У випадку відсутності цього етапу користувач ідентифікується автоматично завдяки обліковому запису так званого *анонімного користувача*. У операційній системі Windows анонімному користувачу відповідає обліковий запис користувача `IUSR_НазваМашини`, де `НазваМашини` – це NETBIOS-назва комп'ютера. Такий тип ідентифікації за замовчуванням ввімкнений на кожному веб-сервері IIS. Для веб-ресурсів загального призначення така ідентифікація значно спрощує процес налаштування веб-серверу, оскільки анонімний користувач володіє обмеженнями в доступі, які часто використовуються на таких ресурсах. Якщо ж ідентифікація необхідна, то її можна виконати з використанням наступних засобів: з використанням систем ідентифікації Windows 2000/XP/2003, з використанням сховищ даних (файл налаштувань `web.config`, реляційні БД або файли XML).

В ASP.NET ідентифікація забезпечується за допомогою спеціальних модулів, які називаються провайдерами ідентифікації. Використання провайдерів ідентифікації вказується у файлах конфігурації ASP.NET machine.config та web.config, які мають формат XML. Наступний код демонструє як налаштувати ідентифікацію:

```
<configuration>
  <system.web>
    <authentication mode="НазваРежиму" />
  </system.web>
</configuration>
```

Атрибут mode використовується для вказівки, який тип ідентифікації потрібно використовувати. Даний параметр може приймати 4 значення: Windows (для ідентифікації використовуються засоби, вбудовані в операційну систему), Forms (ідентифікація користувачів виконується на спеціальних web-формах реєстрації), Passport (ідентифікація відбувається на спеціальному централізованому web-ресурсі фірми Microsoft за допомогою електронних паспортів) та None (ідентифікація не використовується) [2]. Розглянемо більш детально кожен тип ідентифікації.

Ідентифікація типу Windows. Даний тип ідентифікації ґрунтується на використанні списків користувачів Windows. Для використання такої ідентифікації на web-сервері, необхідно внести в його настройки деякі корективи, а саме вимкнути можливість анонічного доступу у діалоговому вікні Authentication Methods. Дане вікно викликається для окремих web-сайтів та директорій всередині web-сайтів, але не для всього серверу IIS. Після цього стануть доступними для вибору 3 режими Windows-ідентифікації: базова ідентифікація, дайджест-ідентифікація, вбудована Windows-ідентифікація. В режимі базової ідентифікації для отримання сервером імені користувача та його паролі використовується стандартне вікно web-реєстрації Windows. Потрібно зазначити, що інформація користувача (ім'я та пароль) передаються по мережі в звичайному вигляді. Таку інформацію дуже легко перехопити спеціальними засобами, а отже використовувати базову ідентифікацію потрібно дуже обережно [1]. Дайджест-ідентифікація може використовуватись тільки у випадку наявного домену мережі. При використанні цього режиму ім'я користувача та його пароль підлягають шифруванню перед відправкою на сервер. Насправді, використовується не шифрування, а хешування інформації. Після передачі цієї інформації сервер перевіряє отриману хеш-інформацію із своєю базою даних. Якщо відповідність знайдено, то користувач отримує доступ до інформаційного ресурсу. При використанні останнього режиму (так звана NTLM-ідентифікація) запит на введення інформації користувача не виконується. Замість цього, браузер використовує ім'я користувача та його пароль, які використовувались при реєстрації на комп'ютері клієнта. при цьому браузер непомітно шифрує необхідні дані та передає їх на сервер. Сервер, в свою чергу, оцінює отриману інформацію і

вирішує питання надання користувачу доступ до ресурсу. Необхідно відмітити, що список користувачів на web-сервері повинен містити облікові записи користувачів із клієнтських комп'ютерів для виконання порівняння. Додатково накладається обмеження на типи браузерів, які можуть використовувати клієнти, а саме: описані можливості підтримує лише браузер Internet Explorer.

Ідентифікація по паспорту. Даний тип ідентифікації підтримується фірмою Microsoft як приватна розробка, а отже являється платною послугою. Даний тип ідентифікації працює наступним чином. Коли користувач виконує запит до захищеного ресурсу, web-сервер перевіряє наявність коректної паспортної інформації у запиті. Якщо така інформація відсутня, то сервер перенаправляє клієнта на web-вузол служби паспортної ідентифікації. Web-вузол проводить ідентифікацію користувача і, у випадку її успішності, перенаправляє його до сервера реєстрації, використовуючи протокол безпечної передачі SSL. Якщо сервер реєстрації успішно ідентифікує користувача, то він автоматично повертається на початковий web-ресурс разом із передачею захищених паспортних даних. Для підвищення захищеності паспортних даних використовується шифрування за алгоритмом Triple-DES [2]. Для використання паспортної ідентифікації в ASP.NET потрібно змінити режим ідентифікації у файлі налаштувань.

Ідентифікація за допомогою web-форм. Даний вид ідентифікації використовується в тих випадках, коли потрібно організувати специфічний та гнучкий доступ до ресурсів на web-вузлі. В цьому випадку задача ідентифікації клієнта покладається на розробника сценаріїв. При цьому розробник може передбачити шифрування параметрів користувача, які передаються в запиті; виконувати збереження інформації про клієнта та пов'язані із ним права доступу в захищеній базі даних тощо. При використанні такого підходу користувачі направляються на спеціальну форму реєстрації, що зберігається на сервері і розробляється окремо. Форма реєстрації запитує в користувача необхідну інформацію і вирішує питання про прийняття чи відхилення реєстрації. Якщо додаток приймає дані користувача, то ASP.NET створює на стороні клієнта спеціальну характеристику (cookie). Дана характеристика використовується для подальшої авторизації. Створення характеристики необхідне для того, щоб реєстрація користувача в інших розділах сайту відбувалась автоматично. Дана характеристика в основному зберігається до моменту закриття браузера або завершення сесії. Але кожна характеристика містить спеціальну властивість Expires, яка визначає термін знищення окремої характеристики. Змінюючи цю властивість можна дати вказівку серверу завжди ідентифікувати користувача автоматично. Для використання ідентифікації на основі web-форм необхідно задати наступні налаштування у файлі web.config:

```
...  
<authentication mode="Forms">
```

```
<forms name="назва" loginURL="форма реєстрації"></forms>  
</authentication>
```

...

Атрибут name дозволяє вказати назву характеристики, яка зберігатиме інформацію про користувача. Атрибут loginURL вказує, яка сторінка буде використовуватись для реєстрації користувача [1]. Форма реєстрації може містити будь-який код для перевірки коректності імені користувача або його паролю: починаючи з простої перевірки із визначеними статичними даними, які задаються в програмному кодї, до виконання складних перевірок на наявність такої інформації у базї даних.

Існує ще один спосіб ідентифікувати користувача за допомогою web-форм. Цей спосіб ґрунтується на синтаксисі конфігураційного файлу web.config і дозволяє задавати необхідну для інформацію про користувачів за допомогою тегу credentials. Синтаксис задання ідентифікації в цьому випадку наступний:

```
...  
<forms name="назва" loginURL="форма реєстрації">  
  <credentials passwordFormat="режим передачі паролю">  
    <user name="user1" password="password1"></user>  
  </credentials>  
</forms>
```

...

Як видно, список користувачів ведеться безпосередньо в файлі конфігурації. В такий спосіб можна задати необмежену кількість користувачів. Параметр passwordFormat може приймати 3 значення: Clear (пароль не підлягає шифруванню), MD5 та SHA1 (браузер отримує хеш-значення пароля користувача по заданому алгоритму). Недоліком методу є зберігання в файлі значення паролю для кожного користувача, що іноді буває неприпустимо з точки зору безпеки. Наступним кроком після ідентифікації є авторизація.

Авторизація – це процес визначення ресурсів, до яких може отримати доступ той чи інший користувач. Авторизація в ASP.NET може відбуватися 2 шляхами:

1. Авторизація за допомогою операційної системи, яка відома ще під назвою *файлової авторизації*.
2. Авторизація по посиланню, яка використовує обмеження доступу на основі посилання на ресурс.

При використанні першого способу авторизації на операційну систему покладається відповідальність за надання доступу до ресурсів, що запитуються. При цьому використовуються спеціальні списки керування доступом (ACL, Access Control Lists). Оскільки кожен ресурс (web-сторінка, зображення тощо) на рівні операційної системи представляє собою простий файл, то до нього можна застосувати систему керування доступом до файлів, вбудовану в Windows. Потрібно зауважити, що такий спосіб авторизації мож-

ливий лише при використанні файлової системи NTFS, яка дозволяє зберігати інформацію про доступ на рівні файлової системи. Таким чином обмеження неявно стосується і операційних систем, оскільки не всі Windows-системи забезпечують підтримку даної файлової системи. З погляду ASP.NET файлова авторизація відбувається автоматично і повністю прозоро для користувача. До найбільш суттєвих недоліків файлової авторизації відносяться наступні: 1) кожен користувач, який отримуватиме доступ до ресурсів web-сервера повинен мати відповідний обліковий запис користувача, 2) при розробці великого web-сайту дуже складно налаштувати необхідні права доступу кожного окремого користувача до кожного окремого ресурсу. Таким чином файлову авторизацію зручно використовувати для невеликих web-сайтів.

Другий спосіб – авторизація по посиланню – співставляє дані користувача із каталогами, які містяться у посиланні [1]. Для включення авторизації по посиланню необхідно налаштувати відповідним чином конфігураційний файл web.config. Наступний синтаксис представляє приклад налаштування:

```
<authorization>  
  <allow users="список користувачів" roles="список груп користувачів" />  
  <deny users="список користувачів" />  
  <deny users="?" />  
</authorization>
```

Атрибути allow та deny містять список користувачів або їхніх груп, яким відповідно дозволено та заборонено доступ до конкретного ресурсу. ASP.NET використовує спеціальні позначення для спрощення ведення списків користувачів. Так для анонімного користувача передбачено символ «?», а для «*» - для всіх користувачів.

Потрібно пам'ятати, що кожен каталог на web-сервері може мати свій власний конфігураційний файл. Якщо ж розмістити один файл конфігурації в кореневому каталозі, то його установки будуть впливати на всі файли та підкаталоги, які містяться в кореневому.

Література:

1. Пейн Крис. Освой самостоятельно ASP.NET за 21 день: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 832 с.
2. Dan Hurwitz, Jesse Liberty. Programming ASP.NET, 2nd edition: O'Reilly, 2003. – 1004 p.

К ВОПРОСУ О МЕТОДАХ БОРЬБЫ СО СПАМОМ

А.Н. Бакал

г. Киев, Национальный педагогический университет

имени М.П. Драгоманова

vd34@ukr.net

Спам давно уже перестал быть просто навязчивой рекламой. Технологии, которые используют спамеры для рассылки почты, небезопасны для корпоративных информационных систем. Убытки от спама, на первый взгляд незначительные для отдельного пользователя, в масштабах всей индустрии и даже отдельной крупной компании выглядят впечатляюще. По разным оценкам, на спаме компании теряют от \$50 до \$200 в год в расчете на одного офисного сотрудника. В результате в 2005 году ущерб от спама по порядку величины стал сравним с потерями, которые нанесли мировому сообществу компьютерные вирусы и хакеры. По данным из европейских источников, ущерб от спама во всем мире составляет \$10 миллиардов ежегодно.

Методы борьбы со спамом.

Фильтрация спама осуществляется исходя из признаков и особенностей электронных писем. Она производится автоматически с помощью специализированных технических средств. Как правило, это программные средства, которые выделяют спам из общего потока сообщений и обеспечивают определенные действия над ним (блокировку, архивирование, дополнительную обработку и т.п.). В настоящее время существует множество различных технических средств борьбы со спамом. Они различаются по технологиям, которые они применяют для выделения спама. Применение той или иной технологии фильтрации спама зависит от разных факторов, однако определяющим является то, в каком месте сети применяются анти-спам фильтры. Исходя из наиболее общих подходов, можно выделить три места расположения таких фильтров:

1. Фильтрация на стороне провайдера.
2. Фильтрация на корпоративном сервере.
3. Фильтрация на стороне клиента.

Фильтрация спама на стороне провайдера.

Провайдеры могут фильтровать спам для клиентов, которые держат у них свои почтовые ящики. Обычно это домашние пользователи, использующие доступ по телефонной линии, либо пользователи выделенных линий. Среди них также есть некоторое количество корпоративных пользователей. Это характерно только для компаний, у которых не создана собственная почтовая система, и они держат почту исключительно у провайдера. В некоторых случаях это достаточно удобно и не требует больших затрат. Однако для компаний, у которых создана своя почтовая система, такой спо-

соб фильтрации не применим по следующим причинам:

- Конфиденциальность электронной почты. Эффективная фильтрация почты требует как минимум контроля текстовой составляющей письма, а это означает, что провайдер будет осведомлен о содержании электронной переписки компании.

- Невозможность построения гибкой политики использования электронной почты. Компании, как правило, имеют сложную структуру, в которой различные группы пользователей могут получать определенные категории писем. При этом одно и то же письмо может относиться одновременно к нескольким категориям (письмо может быть спамом для одной категории пользователей и деловым письмом для другой, к примеру, рекламное письмо о выставке профильной продукции для отдела маркетинга будет деловым, а для отдела информационных технологий – спамом).

- Методы и технологии фильтрации на стороне провайдера не применимы для корпоративного пользователя.

Для фильтрации спама провайдеры используют следующие методы фильтрации спама:

- с использованием RBL-сервисов (по почтовым адресам);
- распределенные методы обнаружения спама.

Каждый из способов имеет свои преимущества и недостатки. Попробуем показать, почему эти методы не применимы для корпоративного пользователя.

Фильтрация спама с использованием сервисов RBL.

Фильтрация по RBL-спискам является наиболее стандартным и легко реализуемым методом обнаружения спама, и с учетом этих обстоятельств этот метод в настоящий момент доминирует среди провайдеров. Сервисы RBL (Realtime Blackhole List) были первым эффективным средством борьбы со спамом. Эти сервисы устроены одинаково – имеются списки почтовых адресов известных спамеров, адресов открытых почтовых пересылок (open relay), используемых спамерами эпизодически или регулярно, и списки диапазонов адресов тех сетей, которые не борются со спамерами или слишком к ним либеральны. Доступ к данным спискам осуществляется в реальном времени по протоколу DNS. Почтовые серверы, использующие RBL, в момент приема очередного сообщения запрашивают сервис (или несколько RBL-сервисов) о том, является ли почтовый адрес отправителя письма «плохим», и на основании ответа RBL либо принимают, либо отвергают письмо. Простота идеи имеет и очевидный недостаток – сообщение принимается или отвергается только на основании адреса посылающей стороны (пользователя или другого почтового сервера). В результате, если какой-то почтовый сервер попал в RBL-список, то вся почта (как спам, так и «не спам») с этого сервера уже приниматься не будет. А это не всегда «плохие» серверы. В эти списки могут быть по ошибке внесены и «хорошие» серверы, например, дружественных вам провайдеров. RBL-сервисы в настоящее

время широко используются интернет-провайдером, почтовыми службами и организациями. Во многих случаях качество RBL оценивается по единственному параметру – количеству спама, который проходит через почтовый сервер. Если количество спама удастся уменьшить, данный RBL-сервис считается «хорошим». В то же время есть и другая, не менее важная характеристика – сколько «нормальных» писем не попало к получателям. Здесь речь идет о проблеме ложных срабатываний. Ложным срабатыванием (False Positive) принято считать тот случай, когда «нормальное» письмо (которое получатель не посчитал бы спамом) до получателя не дошло. Сам получатель об этом обычно не узнает, либо узнает по другим каналам связи («я тебе писал» – «а я ничего не получил»), поэтому проблема во многих случаях остается незамеченной.

Однако практика показала, что несмотря на отмеченные недостатки, метод фильтрации спама с использованием RBL-списков обязательно должен применяться. Да, он один не в состоянии решить проблему, но применение его в комплексе с другими решениями, обеспечивающими блокировку спама, дает положительные результаты. Как правило, проверка IP-адреса по RBL-спискам проводится на начальном этапе фильтрации спама и позволяет отсечь почту (20-30%), относящуюся к «сто процентному» спаму. Очень важно понять это, поскольку многие провайдеры, использующие только этот способ фильтрации, «подают» его как панацею, объясняя, что списки они составляют сами, а проверки на ошибки проводятся регулярно.

Распределенные методы обнаружения спама.

Распределенные методы обнаружения спама используют в основном провайдеры и то только крупные, поскольку анализ и принятие решения осуществляется на основе информации, получаемой из крупных почтовых систем с миллионами пользователей. Смысл распределенных методов обнаружения спама заключается в сборе данных о спам-почте из максимально возможного количества точек сети. Эти данные обрабатываются и делаются доступными для всех заинтересованных участников информационного обмена в сети.

В настоящее время реализованы следующие способы сбора данных о рассылках спама:

- Прием спама в специальные «ловушки» (honeypot).
- Голосование пользователей – пользователь, получивший спам, информирует об этом систему сбора данных, предоставляя образец спама. Одним из сервисов такого рода является так называемая «бритва Вайпула» (Vipul Razor). Основная идея сервиса заключается в создании сигнатур спамерских писем, причем письма присылают сами пользователи, а в базу сигнатуры спамерских писем заносятся по принципу голосования (если приславших данное письмо много или они достаточно авторитетны).
- Анализ всей проходящей через почтовую систему почты с целью определения контрольных сумм спам-сообщений и передачи их на цен-

тральный сервер, установленный в сети.

На основании собранных данных, которые выглядят как «такое-то письмо принято в мире столько-то раз», либо «на такое-то письмо пожаловались столько-то раз», строятся списки массовых на данный момент времени рассылок, которые становятся доступными участникам системы в реальном времени. Почтовые системы, приняв письмо, могут узнать его статус и либо отвергнуть (уничтожить, перенаправить в архив или карантин) как спам, либо передать получателю.

К недостаткам распределенных методов фильтрации спама относится, прежде всего, возможность компрометации данных систем. В качестве примера можно привести случай, когда в руки спамеров попадает часть списков «ловушек». В результате они «заваливают» ловушки легитимной почтой, что приводит к увеличению количества ложных срабатываний. Снижение качества работы системы, как правило, происходит в случаях, когда спам в эти «ловушки» перестает поступать.

Качество работы систем с голосованием пользователей напрямую зависит от активности пользователей. Влиять на такую активность практически невозможно, а скомпрометировать систему легко. Достаточно спамерам стать голосующими участниками и голосовать «против» легитимных рассылок.

Кроме того, серьезной проблемой для описанных методов детектирования массовых рассылок является уникальность каждого отдельно взятого спам-сообщения – каждое современное спамерское письмо существует в огромном количестве вариантов с незначительными отличиями в тексте. На сегодняшний день ни одна из распределенных систем полностью разрешить данную проблему не способна.

Вторая проблема связана с ложными определениями легальных рассылок как спама. Эта проблема характерна как для методов, анализирующих всю почту, так и для систем с голосованием пользователей.

Почтовый сервер провайдера характеризуется большим потоком писем. На нем можно обеспечить гарантированную производительность, на нем есть постоянная связь с другими серверами в сети. Однако именно из-за массового характера поступления почты на стороне провайдера практически не применимы в чистом виде алгоритмы, осуществляющие фильтрацию по смысловому содержанию текста письма. Массовость предполагает, кроме того, использование неперсонализированных анти-спам продуктов. Ведь в неперсонализированной анти-спам системе, которой известны предпочтения только усредненного пользователя, высокий показатель определения спама (как заявляют представители провайдеров, качество фильтрации таких систем составляет 98%) теоретически недостижим.

У корпоративного клиента совершенно иная картина. Почтовый поток не такой массовый, как у провайдера. Кроме того, невозможно или слишком дорого постоянно «закачивать» массивы контрольных сумм писем или IP

«черных дыр». Зато очень точно можно отличить чужие письма, они всегда не похожи на ваши по смыслу; стиль одного пользователя (группы пользователей) выявить несложно. Корпоративному пользователю необходимы системы, которые имеют возможность работать с текстом письма и не просто определять его содержимое, а уметь относить данное письмо к определенной категории, предназначенной для той или иной группы пользователей. Именно алгоритмы фильтрации, основанные на разборе и анализе текста, способны сегодня обеспечить эффективное определение спама. Они имеют возможность проводить более гибкую фильтрацию и персонализировать процесс обработки почты. Технически осуществить решение данной задачи возможно путем обеспечения фильтрации спама на корпоративном сервере.

Фильтрация спама на корпоративном сервере.

Большинство средних и крупных компаний имеют свой корпоративный почтовый сервер, установленный в офисе компании. Это значит, что средства фильтрации провайдера в данном случае не применимы. Для таких компаний существует категория специального серверного программного обеспечения – продуктов, позволяющих фильтровать спам на корпоративном почтовом сервере до рассылки его по рабочим местам сотрудников.

Такие почтовые серверы, как Microsoft Exchange, Sendmail, Postfix, обычно включают средства для обеспечения фильтрации содержания почтовых сообщений (спама и вирусов), однако эти средства обычно довольно примитивны и представляют собой «пустые рамки» для правил, то есть предлагают администратору почтовой системы самостоятельно создавать и настраивать правила фильтрации. Этот подход работает не очень хорошо, так как для фильтрации спама нужна гибкая политика, множество правил, которые постоянно обновляются и корректируются.

Данная проблема решается за счет того, что почти все почтовые серверы имеют возможность встраивать или интегрировать системы третьих производителей. Современный рынок информационной безопасности предлагает много продуктов, обеспечивающих фильтрацию спама на корпоративном сервере. Это могут быть как коммерческие, так и бесплатные продукты, распространяемые на условиях лицензии GPL (General Public License) или подобных ей.

Бесплатные фильтры. Наиболее известный бесплатный фильтр – SpamAssassin. Это весьма эффективная программа, фильтрующая 90-95% спама. SpamAssassin поставляется с постоянно обновляемой базой правил фильтрации, как по формальным признакам письма, так и по содержанию (ключевым словам). Недостатки этого фильтра заключаются в том, что он не имеет локальной привязки к языкам и регионам, ориентируясь в основном на англоязычный спам. Набор правил SpamAssassin очень велик и непрозрачен (понять, какое правило сработало можно, но трудно предугадать, к чему приведет его отмена), что очень затрудняет настройку.

Коммерческие продукты. Средства фильтрации спама, реализуемой на корпоративном сервере, предлагают многие производители. Сложность заключается в выборе продукта, который наиболее подходит для контроля использования электронной почты компании.

В настоящее время на рынке анти-спам систем представлены два основных типа фильтров:

- фильтры, работа которых основана на поиске в электронных письмах определенных признаков (так называемые, традиционные фильтры);
- фильтры, применяющие статистические (вероятностные) методы для обеспечения фильтрации спама.

И те, и другие осуществляют контекстную фильтрацию электронной почты, то есть содержание письма для них является одним из важнейших критериев, по которому его можно отнести к спаму. Однако традиционные фильтры обладают довольно серьезными недостатками.

Некачественное разделение спама и обычных писем обусловлено некоторой «однобокостью» традиционных фильтров. При отбраковке писем учитываются «плохие» признаки и не учитываются «хорошие», характерные для деловой переписки.

Этих недостатков лишен метод построения анти-спам фильтров, предложенный американским программистом и предпринимателем Полом Грэмом. Метод Грэма позволяет автоматически настроить фильтры согласно особенностям индивидуальной переписки, а при обработке учитывает признаки как «плохих», так и «хороших» писем. Такой метод фильтрации спама называют статистическим или вероятностным.

Статистические (вероятностные) методы фильтрации спама.

Статистический метод основывается на теории вероятностей и используется для фильтрации спама статистический алгоритм Байеса. Каждому встречающемуся в электронной переписке слову или тегу присваивается два значения: вероятность его наличия в спаме и вероятность его присутствия в письмах, разрешенных для прохождения. Баланс этих двух значений и определяет вероятность того, что письмо, в котором встречаются данные слова и теги, является спамом.

Для вычисления вероятности спама используются таблицы вероятности (принадлежности слов из письма, относящегося к категории «спам»), созданные в процессе обучения фильтра. А именно: берутся как минимум два списка слов различных категорий писем (например, «разрешенных» и «запрещенных») и передаются на обработку программе обучения. Она вычисляет частотные словари для каждой категории сообщений – сколько раз какое слово встречалось в письмах этой категории (в данном случае спама). Когда словари заполнены, вычисление вероятности принадлежности конкретного нового письма к тому или иному типу производится по формуле Байеса для каждого слова этого нового письма. Суммированием и нормализацией вероятностей слов получают вероятности для всего письма. Как пра-

вило, вероятность принадлежности электронного письма к одной из категорий на порядок выше, чем к другим. К данной категории и следует относить сообщение.

Сразу после начального «обучения» фильтра точность определения спама этим методом достигает значительной величины – 97–99% и продолжает уверенно двигаться к 100% после проведения дальнейших корректировок фильтра.

Корректировка фильтра заключается в обработке случаев неправильной классификации писем – фильтру указывается, к какой категории следует впредь относить эти письма, и он добавляет слова из этих писем в соответствующие таблицы вероятностей. Обратим внимание на то, что администратору не приходится вручную анализировать письмо и пополнять на основе проведенного анализа списки правил фильтрации, как это делается в традиционных фильтрах. Достаточно добавить письмо в архив писем данной категории, заново запустить процесс «обучения» фильтра и статистический «портрет» письма меняется полностью и автоматически. Практически байесовский фильтр заменяет все те лингвистические лаборатории, которые осуществляют анализ вновь поступающего спама. Ведь они осуществляют корректировку анти-спам фильтров тем же способом («впредь считай такие письма спамом»).

Приведем основные отличия статистической технологии фильтрации от технологии фильтрации на основе признаков, присущих спаму:

1. Особенность статистической технологии заключается в возможности индивидуальной автоматической настройки фильтра, что является важным преимуществом, поскольку разные люди или же компании (если фильтр устанавливается на корпоративном почтовом сервере) используют в электронной переписке разную лексику. Настройка фильтра производится по результатам статистического анализа имеющегося архива электронной почты или выборки, полученной за определенный период времени. Такой анализ дает возможность накопить достаточно информации для эффективной фильтрации электронной почты.

2. И в том, и в другом случае результатом оценки является так называемый «вес» письма. Однако при применении метода с использованием признаков спама «вес» письма вычисляется только на основе «плохих» признаков, что приводит к «обвинительному уклону» фильтра, и, как следствие, появляются ложные срабатывания.

3. В алгоритме Байеса наборы признаков определяются не субъективно, а в результате статистического анализа реальных подборок писем. Получающиеся наборы признаков оказываются весьма нетривиальными и эффективными. Например, в качестве «плохого» признака может появиться строка «0Xffff» – ярко красный цвет; а в качестве «хорошего» признака – Ваш номер телефона. И действительно, письмо, содержащее Ваши персональные данные, в любом случае следует прочесть.

По имеющимся оценкам, статистический метод борьбы со спамом является весьма эффективным. Так, в процессе испытания через фильтр были пропущены 8 000 писем, половина из которых являлась спамом. В результате система не смогла распознать лишь 0,5% спам-сообщений, а количество ошибочных срабатываний фильтра оказалось нулевым.

Самое важное преимущество байесовского фильтра заключается в том, что он надежно исключает ложные срабатывания. Ведь процесс принятия решения (относится письмо к спаму или нет) осуществляется в соответствии с особенностями индивидуальной переписки, а при обработке учитываются признаки как «плохих», так и «хороших» писем. Именно за счет баланса этих признаков и удается свести к минимуму количество ложных срабатываний фильтра.

Другим преимуществом теоремы Байеса является возможность ее использования для классификации любых текстов письма по любым категориям, и поэтому он имеет более широкое применение, чем тривиальная фильтрация спама.

Таким образом, в настоящее время наиболее эффективным и оптимальным для корпоративных пользователей являются системы, основанные на статистических (вероятностных) методах фильтрации спама.

Проблемы спама могут быть успешно решены только в контексте общей политики информационной безопасности, поскольку комплексное решение задач позволяет бороться с многочисленными угрозами корпоративной информационной системе, которые несет в себе спам.

ІНФОРМАЦІЙНА БЕЗПЕКА ЯК ОСНОВА УПРАВЛІННЯ ГОСПОДАРСЬКОЮ ДІЯЛЬНІСТЮ

Л.М. Петренко

м. Київ, Київський національний економічний університет

імені Вадима Гетьмана

plnkneu@ukr.net

Необхідністю розвитку економічного успіху будь-якого підприємства є розумне використання інформаційного середовища та захист його інформаційного поля, тобто забезпечення інформаційної безпеки. В процесі прийняття ефективних управлінських рішень, якісній організації роботи з інформаційними потоками для плідного функціонування господарюючого суб'єкту саме інформація відіграє найважливішу роль.

Інформація – це відомості, які отримують з різних джерел, котрі необхідно добути або отримати, вивчити та дослідити, прийняти або ні, перевірити, перетворити в ресурс, прийняти рішення по його використанню та розрахувати (спрогнозувати) загальну вигоду. Інформацію необхідно вміти використовувати як ресурс та як ланцюг, що пов'язує процеси діяльності людей та організацій [1]. Тому вона має особливо вагоме значення в управлінській діяльності як основний засіб управління процесами на підприємстві – як внутрішніми, так і зовнішніми.

Ефективне інформаційно-аналітичне забезпечення господарської діяльності підприємства передбачає проведення всебічного аналізу та обробки отриманих даних.

Методами збору інформації є:

– отримання відкритої інформації з різноманітних джерел: баз даних господарюючого суб'єкту, архівів, державних органів, організацій;

– отримання відкритої та закритої інформації завдяки особистим контактам співробітників підприємства з представниками інших державних та комерційних структур;

– збір інформації за всіма аспектами діяльності підприємства.

Інформаційне забезпечення – одна з найважливіших функцій процесу прийняття рішень, якість якого є визначальним фактором обґрунтованості рішення, що приймається та ефективності функціонування підприємства. Його основу складає інформація, що характеризує, по-перше – стан зовнішнього середовища, по-друге – стан підприємства.

В таблиці 1 наведена класифікація інформації за різними ознаками:

Табл. 1. Класифікація інформації за певними ознаками

Класифікаційні ознаки	Види інформації
Відношення до предмету дослідження	Основна, допоміжна

Класифікаційні ознаки	Види інформації
Зміст	Науково-технічна, економічна, соціальна, правова
Відношення до об'єкту аналізу	Внутрішня, зовнішня
Канали проходження	Вхідна, вихідна
Періодичність надходження	Регулярна, епізодична
Ступінь постійності	Постійна, умовно-постійна, змінна
Джерела документування	Первинний облік, звітність
Відношення до процесу обробки	Первинна, вторинна, похідна
Ступінь обхвату об'єкту	Загальна, локальна
Види вимірників	Натуральні, трудові, вартісні
Відношення до часу прийняття рішення	Попередня, оперативна, підсумкова
Призначення	Довідкова, рекомендаційна, нормативна, сигнальна, регулююча

Інформація повинна мати певну мету в момент передачі певній особі, в протилежному випадку вона є просто даними або шумами [1]. В процесі обміну інформацією виділяють чотири базових елементи:

- 1) *відправник* – особа, яка генерує ідею або збирає, кодує та передає інформацію;
- 2) *повідомлення* – предмет обміну, власне інформація, закодована з використанням символів;
- 3) *канал* – мова, листи, електронні засоби зв'язку;
- 4) *отримувач* – особа, яка декодує інформацію, кодує її відгук.

В процесі обміну інформацією відправник та отримувач міняються ролями завдяки зворотному зв'язку, наявність якого дозволяє підвищувати якість інформації шляхом усунення шумів (рис. 1).

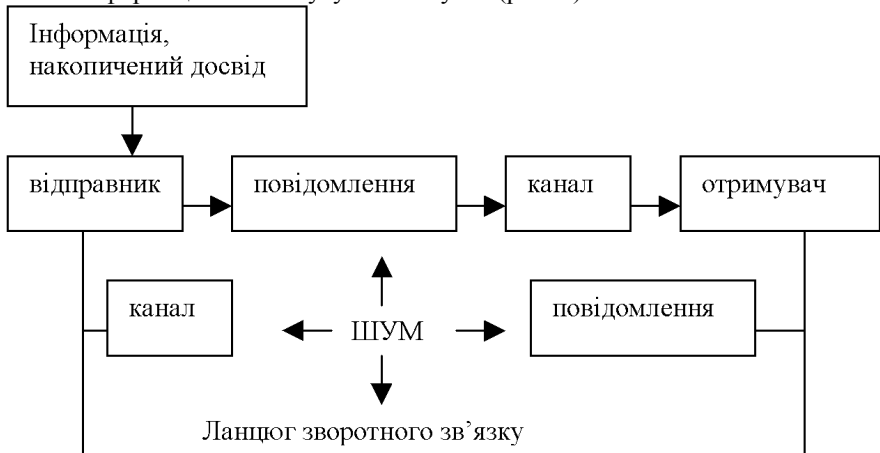


Рис. 1. Процес обміну інформацією

Динамічність ринкових ситуацій вимагає прискорення процесів збору, обробки інформації та створення нових інформаційних технологій на підприємстві. Кардинальна зміна інформаційно-комунікаційних процесів відбулася з впровадженням комп'ютерних технологій. Для виконання рішень необхідним є наявність наступних складових інформаційних технологій:

- комунікаційних засобів;
- баз вхідних даних про стан підприємства та зовнішнього середовища;
- банка методів обробки та аналізу даних, необхідних для оцінки показників результатів кожного рішення за різних факторів впливу на інформаційну безпеку, результатів рішень та можливих втрат;
- банка методів обґрунтування найбільш альтернативних рішень;
- банка методів управління інформаційною безпекою підприємства;
- менеджерів по управлінню інформаційною безпекою.

Джерела інформації можуть бути різними: від засобів масової інформації до проведення спеціальних самостійних досліджень або досліджень з залученням відповідних фірм. Витрати на отримання інформації можуть досягати значних сум, тому об'єм необхідної інформації для ефективного функціонування системи управління інформаційною безпекою підприємства повинен оцінюватися додатково попередньо. При цьому необхідно враховувати, щоб вигода від отримання інформації перевищувала витрати на її отримання.

Для кожного підприємства надто важливе значення набуває стан його інформаційної інфраструктури. Вона має дві сторони. Перша пов'язана з технічним оснащенням та супроводом інформації, яка включає комп'ютери для збору, обробки інформації, програмне забезпечення, зберігання інформації, формування інформаційних банків даних, банків методів обробки інформації, засоби зв'язку з банками інших організацій, засоби зв'язку підприємства (телефон, телефакс, комп'ютерний зв'язок).

Негативний вплив низької якості інформації на господарські рішення проявляється через два діючі фактори [2]. Перший з них пов'язаний з прийняттям рішення в умовах невірної уявлення про стан об'єкту, що призводить до помилок. Інший фактор полягає, з одного боку, в тому, що розробка методів прийняття рішень або їх використання в умовах інформаційного незабезпечення не має сенсу, а з іншого – в нерациональності розвитку інформаційної інфраструктури. Один з способів огляду якості інформації полягає в оцінці її характеристик з точки зору їх значення для підприємства. В число значимих відмінних властивостей інформації відноситься час, зміст та форма. Характеристики перерахованих властивостей високоякісної інформації наведено на рис. 2.

Недостовірна інформація може з'являтися в результаті різного роду помилок, таких як помилки при зборі даних, а також в результаті перекручення під час передачі по комунікаційним каналам і в результаті некорект-

ного використання методів обробки та аналізу.



Рис. 2. Характеристики високоякісної інформації

Проблема надійності вимірювання вирішується шляхом виявлення правильності виміру, стійкості та обґрунтованості виміру [3]. При вивченні правильності встановлюється загальна сприйнятливність даного способу виміру (шкали або системи). Безпосередньо поняття правильності пов'язане з можливістю обліку в результаті вимірювання різного роду систематичних помилок. Стійкість характеризує ступінь збігу результатів вимірювання при повторних вимірах та описується випадковою величиною помилки. Вона визначається постійністю до відповідей на однакові або подібні питання. Найбільш складне питання надійності виміру – його обґрунтованості. Обґрунтованість пов'язана з доведенням того, що виміряна визначена задана властивість об'єкту, а не інша, більш чи менш схожа на неї. Можливі різноманітні формальні підходи до виявлення рівня обґрунтованості методики дослідження. Їх можна поділити на три групи: конструювання топології у відповідності до мети дослідження на базі декількох ознак, використання паралельних даних та процедури обговорення [3].

Нерідко причинами інформаційної недостатності виступають часові обмеження: більшість господарських рішень повинно бути прийнято протя-

гом короткого інтервалу часу. У цьому зв'язку особа, що приймає рішення (ОПР), володіє неповною або недостовірною інформацією. Додаткова інформація може бути зібрана, недостовірною – перевірена, але для цього потрібен час. Разом з тим затримка в прийнятті рішення може призвести до ще гірших наслідків, ніж прийняття неповної інформації.

Отже, управлінці на підприємстві повинні оцінити, яка інформація потрібна для виконання відповідних управлінських функцій для того, щоб обґрунтувати доцільні інформаційні потоки, визначити необхідні засоби зв'язку, прийому, зберігання, обробки та передачі інформації, забезпечити необхідний рівень інформаційної безпеки.

Література:

1. Соснин А.С., Прыгунов П.Я. Менеджмент безопасности предпринимательства: Учеб. пособие. – К.: Изд-во Европ. ун-та, 2004. – 357 с.
2. Голошубов О. Информационное обеспечение принятия маркетинговых решений // Бизнесинформ. – 1997.
3. Голубков Е.П. Маркетинговые исследования: теория, методология и практика. – М.: Финпресс, 1998. – 416 с.

Наші автори

Алпатова Юлія Сергіївна, студент Дніпропетровського національного університету (*фундаментальні дисципліни*)

Аль Маді Мудар Кадрі, аспірант Черкаського державного технологічного університету

Андрієнко Володимир Олександрович, аспірант Черкаського державного технологічного університету

Бабич Олександр Вікторович, старший викладач кафедри математичного аналізу та інформатики, викладач спецдисциплін Полтавського державного педагогічного університету ім. В.Г. Короленка (*ОС Linux, UML, автоматизація обчислень, програмування, БД*)

Бакал Анатолій Миколайович, завідувач лабораторії спеціального фізичного практикуму Національного педагогічного університету ім. М.П. Драгоманова (*Web-програмування, астрономія, психологія*)

Балабай Руслана Михайлівна, к.ф.-м.н., доцент кафедри фізики Криворізького державного педагогічного університету (*комп'ютерне моделювання, фізичні основи мікроелектроніки*)

Баранов Володимир Юрійович, викладач кафедри інформаційних технологій Одеської національної академії зв'язку імені О.С. Попова (*системи комп'ютерного тестування знань, тестологія, методи оцінювання відкритих відповідей, контроль знань та тестування з інформатики та програмування, з математики та з іноземних мов (французька, англійська)*)

Бендес Юрій Петрович, к.ф.-м.н., завідувач кафедрою фізики, математики та хімії Полтавського військового інституту зв'язку (*дидактика фізики*)

Боско Ольга Миколаївна, старший викладач Інституту ділового адміністрування (*вивчення методів викладання прикладного програмного забезпечення у вищих економічних і технічних закладах та пов'язаних з цим проблем міждисциплінарних зв'язків*)

Бугайова Наталія Михайлівна, к.психол.н., старший науковий співробітник лабораторії нових інформаційних технологій навчання Інституту психології ім. Г.С. Костюка АПН України (*адикції, суїцид*)

Васильєв Костянтин Юрійович, к.т.н., доцент Дніпропетровського національного університету (*точні науки*)

Величко Ігор Миколайович, асистент кафедри вищої математики і фізики Керченського морського технологічного інституту (*математика*)

Волкова Тетяна Василівна, к.пед.н., доцент Бердянського державного педагогічного університету імені П.Д. Осипенко

Воронов Валерій Олександрович, старший викладач кафедри інформаційних технологій Хмельницького економічного університету (*інформаційні технології в навчанні, самоосвіта студентів та викладачів, взаємозв'язок шкільної освіти та вищої школи*)

Гайша Олександр Олександрович, аспірант кафедри фізики Національ-

ного університету кораблебудування імені адмірала Макарова (*проблеми захисту інформації у комп'ютерних системах та мережах, математичне моделювання, дослідження процесів тепло- та масопереносу*)

Гарбич Ольга Романівна, викладач кафедри інформатики та обчислювальної математики Дрогобицького державного педагогічного університету імені Івана Франка (*робота з обдарованими дітьми*)

Говаленков Сергій Валентинович, к.т.н., доцент Академії цивільного захисту України (*дистанційна освіта*)

Гризун Людмила Едуардівна, к.пед.н., доцент кафедри інформаційних технологій Харківського національного педагогічного університету ім. Г.С. Сковороди (*інформаційні технології в освіті, проектування навчальних дисциплін на засадах інтеграції наукових знань*)

Гринь Наталя Володимирівна, старший викладач Інституту ділового адміністрування (*вивчення методів викладання прикладного програмного забезпечення у вищих економічних і технічних закладах та пов'язаних з цим проблем міждисциплінарних зв'язків*)

Губенко Наталя Євгенівна, к.т.н., доцент Донецького національного технічного університету (*моделі та методи захисту інформації, моделі та системи Е-навчання, методи та способи КІТ*)

Гусева Любов Володимирівна, викладач кафедри інформатики та обчислювальної техніки Академії цивільного захисту України (*дистанційна освіта*)

Джулій Володимир Миколайович, к.т.н., доцент Хмельницького національного університету (*програмування, системи захисту інформації*)

Дорошенко Надія Іванівна, молодший науковий співробітник відділу дидактики Інституту педагогіки і психології професійної освіти АПН України (*методика навчання креслення та інженерної графіки*)

Дорошенко Юрій Олександрович, д.т.н., професор, завідувач лабораторії навчання інформатики Інституту педагогіки АПН України (*навчання інформатики та ІКТ; комп'ютерні графічно-інформаційні технології; геометричне моделювання об'єктів, процесів і явищ*)

Железко Борис Олександрович, к.т.н., доцент, професор кафедри інформаційних технологій Білоруського державного економічного університету (*інформаційні технології і системи автоматизації наукових досліджень, теорія і практика створення і підвищення ефективності адаптивних інформаційно-аналітичних систем підтримки прийняття рішень (ІА СППР) з комбінованим інтелектом, теорія моделювання і проектування ІА СППР, корпоративні інформаційні системи і персональні автоматизовані робочі місця спеціалістів, застосування нечітких множин в економіці і фінансах, прогнозування в нечіткій постановці*)

Жуков Микола Степанович, к.т.н., доцент Криворізького металургійного факультету Національної металургійної академії України (*системи автоматичного керування, цифрові системи, моделювання систем і процесів*)

Жукова Людмила Леонтівна, к.т.н., доцент, завідувач кафедри Криворізького навчально-консультаційного центру Запорізького національного університету (*інформаційні системи, моделювання систем і процесів*)

Забарна Алла Петрівна, вчитель інформатики, заступник директора Смілянського природничо-математичного ліцею (*профільне навчання інформатики в загальноосвітніх навчальних закладах*)

Кабанова Тетяна Віталіївна, учитель інформатики Санкт-Петербурзької середньої школи №58 (*дидактика інформатики середньої школи*)

Кірієнко Наталія Олексіївна, к.т.н., доцент, старший науковий співробітник Об'єднаного інституту проблем інформатики Національної академії наук Білоруси (*методика навчання сучасним засобам розробки програм*)

Клименко Ольга Федорівна, к.е.н., доцент кафедри інформатики Київського національного економічного університету імені Вадима Гетьмана (*інформаційні системи в економіці, інформатика*)

Ключко Оксана Віталіївна, к.пед.н., доцент кафедри інформаційних технологій в менеджменті Вінницького державного аграрний університету (*інформатика, прикладна математика*)

Кобильник Тарас Петрович, аспірант кафедри основ інформатики та обчислювальної техніки Національного педагогічного університету імені М.П. Драгоманова (*інформаційні технології навчання, системи комп'ютерної математики, методика викладання інформатики*)

Коломенська Валентина Володимирівна, к.ф.-м.н., доцент Донецького національного університету (*методика викладання фізики і інформатики*)

Кондратенко Світлана Вікторівна, асистент Криворізького державного педагогічного університету (*методика інформатики*)

Кононова Олена Олександрівна, студентка 5 курсу фізико-математичного факультету Криворізького державного педагогічного університету (*криптографія, програмування*)

Кравченко Володимир Володимирович, асистент кафедри інформатики та прикладної математики Криворізького державного педагогічного університету (*комп'ютерно-орієнтовані системи навчання, програмування*)

Красильников Сергій Романович, к.т.н., доцент, заст. декана факультету комп'ютерних систем та програмування Хмельницького національного університету (*комп'ютерні системи та мережі*)

Красильникова Ганна Володимирівна, к.т.н., доцент, начальник навчально-методичного управління Хмельницького національного університету (*інноваційні технології та методика викладання спеціальностей*)

Красюк Юлія Миколаївна, к.пед.н., старший викладач кафедри інформатики Київського національного економічного університету імені Вадима Гетьмана (*комп'ютерно-орієнтовані системи навчання, методика навчання інформатики студентів економічних спеціальностей*)

Кривель Ірина Анатоліївна, старший викладач Факультету ринкових, інформаційних та інноваційних технологій Київського національного уні-

верситету технологій та дизайну (*інформатика, математика*)

Кудрявцева Олена Петрівна, інженер-програміст інформаційно-комп'ютерного центру Хмельницького національного університету (*проекування баз даних, web-дизайн, web-програмування*)

Кузнецов Антон Юрійович, студент 3 курсу механіко-математичного факультету Харківського національного університету ім. В.Н. Каразіна (*комп'ютерне моделювання, математична фізика, теоретична механіка, інформатика*)

Кулик Андрій Анатолійович, асистент Хмельницького національного університету (*захист інформації*)

Леоненко Леонід Леонідович, к.філос.н., доцент кафедри інформаційних технологій Одеської національної академії зв'язку імені О.С. Попова (*некласична логіка, доведення за аналогіями, системи комп'ютерного тестування знань, методи оцінювання відкритих відповідей, контроль знань та тестування з математики, інформатики*)

Лісіна Людмила Олександрівна, старший викладач Криворізького державного педагогічного університету (*методика викладання інформатики*)

Ліхачов Роман Вікторович, асистент Криворізького металургійного факультету Національної металургійної академії України (*Web-дизайн, програмування, нові мови програмування, розробка промислових та програмних інтерфейсів, системи комп'ютерної математики*)

Логвіненко Вікторія Григорівна, к.пед.н., старший викладач кафедри кібернетики та інформатики Сумського національного аграрного університету (*психолого-педагогічні та організаційні аспекти викладання дисциплін комп'ютерного циклу*)

Ломаєва Тетяна Василівна, к.ф.-м.н., доцент кафедри вищої математики Національного педагогічного університету імені М.П. Драгоманова (*геометрія*)

Лукаш Ірина Миколаївна, к.пед.н., доцент Чернігівського державного педагогічного університету імені Т.Г. Шевченка (*методика викладання інформатики, ООП*)

Мазнюк Юлія Іванівна, методист навчального відділу Полтавського державного педагогічного університету ім. В.Г. Короленка (*методика викладання фізики*)

Матвієнко Юрій Сергійович, асистент кафедри математичного аналізу та інформатики Полтавського державного педагогічного університету імені В.Г. Короленка (*комп'ютерна техніка, програмне забезпечення навчального призначення, інформаційні технології*)

Мельников Олександр Юрійович, к.т.н., доцент кафедри економічної кібернетики Донбаської державної машинобудівної Академії (*програмування, проектування інформаційних систем (UML), комп'ютерна математика, web-дизайн*)

Миронова Олена Іванівна, асистент кафедри прикладної математики

Волинського державного університету імені Лесі Українки (*навчаючі системи, користувацький інтерфейс*)

Мінтій Ірина Сергіївна, асистент Криворізького державного педагогічного університету (*методика викладання інформатики*)

Моїсеєнко Наталя Володимирівна, к.ф.-м.н., доцент, завідувач кафедри інформатики та прикладної математики Криворізького державного педагогічного університету (*програмування, моделювання фізичних процесів, комп'ютерна графіка, криптографія*)

Моргун Олександр Миколайович, к.т.н., доцент, завідувач кафедри Черкаського інституту пожежної безпеки імені Героїв Чорнобиля (*інформатика, математика*)

Москаленко Артем Олексійович, ад'юнкт Полтавського військового інституту зв'язку (*методика викладання фізики та захисту інформації*)

Мясіщев Олександр Анатолійович, д.т.н., професор, завідувач кафедри комп'ютерних систем та мереж Хмельницького національного університету (*комп'ютерні системи та мережі*)

Настенко Ірина Василівна, аспірант кафедри інформатики Національного педагогічного університету імені М.П. Драгоманова

Онишко Оксана Григорівна, асистент кафедри вищої математики Хмельницького національного університету (*дидактика математики, інформатики вищої школи*)

Охременко Максим Григорович, студент Дніпропетровського національного університету (*точні науки*)

Очеретний Володимир Олександрович, аспірант Інституту педагогіки АПН України (*навчання інформатики та ІКТ; комп'ютерні графічно-інформаційні технології; геометричне моделювання об'єктів, процесів і явищ*)

Паніна Олена Олександрівна, викладач кафедри інформатики та обчислювальної техніки Академії цивільного захисту України (*дистанційна освіта*)

Пастухов Роман Сергійович, студент Донецького національного університету (*Internet-технології в навчанні*)

Петренко Людмила Миколаївна, асистент Київського національного економічного університету імені Вадима Гетьмана (*інформатика, економічна безпека підприємств*)

Петрушина Лідія Іванівна, вчитель початкових класів Криворізької загальноосвітньої школи №99 (*ігрова діяльність*)

Поліщук Олександр Павлович, к.т.н., с.н.с., доцент кафедри інформатики та прикладної математики Криворізького державного педагогічного університету (*об'єктно-орієнтоване моделювання динамічних систем, технології програмування*)

Полозова Валентина Михайлівна, к.е.н., доцент кафедри економіки Хмельницького національного університету (*комп'ютерні системи та ме-*

режі)

Протаскіна Юлія Вадимівна, студентка 5 курсу фізико-математичного факультету Криворізького державного педагогічного університету (*програмування складних систем*)

Румянцев Михайло Ігорович, старший викладач кафедри прикладної математики та інформатики Західнодонбаського приватного інституту економіки та управління (*автоматизовані системи управління підприємствами; теорія систем; імітаційне моделювання економічних систем; інформаційні технології у вищій школі*)

Рябцев Володимир Григорович, д.т.н., професор Черкаського державного технологічного університету

Рябенко Денис Володимирович, студент Донецького національного технічного університету (*моделі та методи захисту інформації, розробка складних програмних систем*)

Рябчиков Дмитро Віталійович, старший лаборант кафедри фізики Криворізького державного педагогічного університету (*схемотехніка, радіоелектроніка*)

Савчук Людмила Олександрівна, старший викладач кафедри інформаційних технологій Хмельницького економічного університету (*активізація пізнавальної діяльності студентів; методика викладання комп'ютерно-орієнтованих дисциплін*)

Семеріков Сергій Олексійович, к.пед.н., доцент кафедри інформатики та прикладної математики Криворізького державного педагогічного університету (*системне програмування, комп'ютерно-орієнтовані системи навчання, дидактика інформатики вищої школи*)

Смалько Олена Аркадіївна, к.пед.н., старший викладач кафедри інформатики та методики її викладання Кам'янець-Подільського державного університету (*методика викладання інформатики, новітні інформаційні технології, комп'ютеризована підтримка вивчення різних навчальних дисциплін, педагогічні програмні засоби, комп'ютерна графіка та дизайн, захист комп'ютерної інформації*)

Смілянecь Олена Геннадіївна, старший викладач кафедри інформаційних систем і мереж Вінницького торговельно-економічний інститут Київського національного торговельно-економічного університету (*застосування інформаційних технологій у вищій школі; методика навчання інформатики; розв'язання творчих економічних задач на персональному комп'ютері*)

Столяревська Алла Леонідівна, к.пед.н., доцент кафедри інформатики Харківського національного педагогічного університету імені Г.С. Сковороди (*мови програмування, дискретна математика, застосування нових інформаційних технологій у викладанні математики, дистанційна освіта, бази даних, інтелектуальні системи*)

Супрунова Юлія Анатолівна, старший викладач кафедри технічної кібернетики Криворізького інституту Кременчуцького університету економі-

ки, інформаційних технологій і управління (*інженерна та дизайнерська комп'ютерна графіка, створення тестових та навчальних програм, програмування, HTML*)

Сушенцева Лілія Леонідівна, к.пед.н., доц., завідувач кафедри інженерної педагогіки та мовної підготовки Криворізького технічного університету (*методична робота в навчальних закладах*)

Сушевський Дмитро Валерійович, студент Дніпропетровського національного університету (*фізика, програмування в об'єктно-орієнтованих середовищах, теоретичні основи мереж зв'язку*)

Триус Юрій Васильович, к.ф.-м.н., доцент, проректор з навчальної роботи Черкаського національного університету імені Богдана Хмельницького (*комп'ютерно-орієнтовані системи навчання математичних дисциплін*)

Туравініна Оксана Миколаївна, асистент Криворізького державного педагогічного університету (*вдосконалення управлінських функцій між суб'єктами навчально-виховного процесу*)

Учитель Марія Сергіївна, студентка 1-го курсу спеціальності «Фінанси» Національної металургійної академії України (*дистанційна освіта, фінансовий аналіз*)

Фільо Ірина Євгенівна, аспірант, асистент Національного університету водного господарства та природокористування (*проблеми комп'ютеризації вищої школи, математичне моделювання процесу навчання, методика викладання інформатики*)

Хмельницький Юрій Владиславович, к.т.н., доцент Хмельницького національного університету (*захист інформації*)

Холошня Тетяна Анатоліївна, викладач Західнодонбаського приватного інституту економіки та управління (*педагогіка*)

Чаплигіна Світлана Миколаївна, к.ф.-м.н., доцент кафедри прикладної математики Придніпровської державної академії будівництва і архітектури (*інформатика*)

Шаповалова Наталія Валентинівна, к.ф.-м.н., доцент кафедри вищої математики Національного педагогічного університету імені М.П. Драгоманова (*геометрія*)

Шматко Олександр Віталійович, к.т.н. Академії цивільного захисту України (*дистанційна освіта*)

Штиркова Ольга Юріївна, студент Криворізького технічного університету

Яковлева Ірина Олександрівна, к.т.н., доцент Академії цивільного захисту України (*дистанційна освіта*)

Зміст

<i>Ю.В. Триус.</i> Комп'ютерно-орієнтовані методичні системи навчання у вищих навчальних закладах	3
Розділ І. Дидактика інформатики вищої школи	7
<i>О.П. Поліщук, С.О. Семеріков.</i> Методичні та організаційні проблеми навчання комп'ютерного програмування у вищих навчальних закладах.....	8
<i>О.П. Поліщук, С.О. Семеріков.</i> Спецкурс з сучасних методів і технологій розробки програмних виробів компонентної архітектури	12
<i>Л.О. Савчук.</i> Сучасні методи і засоби навчання, що активізують пізнавальну діяльність студента при вивченні дисциплін комп'ютерної направленості	16
<i>В.Г. Логвіненко.</i> Цілі вивчення дисциплін комп'ютерного циклу та інформаційних технологій у вищій школі	20
<i>О.І. Миронова.</i> Організація самостійної роботи студентів спеціальності “Інформатика” з використанням інформаційно-навчального середовища.....	25
<i>В.О. Воронов.</i> З досвіду впровадження кредитно-модульної системи викладання дисципліни “Інформатика та комп'ютерна техніка”	29
<i>С.Р. Красильников, Г.В. Красильникова.</i> Особливості викладання навчальної дисципліни “Основи інформаційних технологій” за вимогами кредитно-модульної системи навчання	33
<i>С.В. Говаленков.</i> Впровадження модульних технологій навчання	37
<i>Н.А. Кириенко.</i> Тестирование как средство повышения эффективности лекционного процесса.....	40
<i>А.В. Шматко, Е.А. Панина.</i> Информационные технологии диагностики слушателей при преподавании информатики.....	45
<i>В.В. Кравченко.</i> Использование XML для организации и хранения базы тестовых заданий.....	50
<i>В.Ю. Баранов, Л.Л. Леоненко.</i> Методичні особливості використання системи “Control” на заняттях з інформатики у вищій школі.....	56
<i>А.Ю. Мельников.</i> Тестирование знаний студентов при помощи программного комплекса «MaTest-2.0»	61
<i>М.И. Румянцев.</i> Опыт разработки и внедрения программного комплекса LazyTest для автоматизированного тестирования уровня знаний студентов.....	66
<i>М.С. Учитель.</i> К вопросу о финансовой эффективности дистанционного образования.....	73
<i>В.В. Коломенська, Р.С. Пастухов.</i> Автоматизована навчальна система по розробці Web-сторінок	78
<i>Л.В. Гусева, А.В. Шматко.</i> Использование системы дистанционного интернет-обучения «МОДУС».....	83
<i>О.А. Мясцев, О.П. Кудрявцева.</i> Корпоративна мережа Хмельницького національного університету	87

<i>К.Ю. Васильев, Д.В. Суцевский.</i> Поиск решения проблемы перехода существующих телекоммуникационных сетей к поколению 3G	106
<i>А.В. Шматко.</i> Построение информационной системы кафедры с использованием терминальных систем.....	110
<i>Н.В. Моисеенко, Ю.В. Протаскина.</i> Забезпечення складського обліку (архітектура клієнт-сервер)	115
<i>А.А. Мясичев, В.М. Полозова.</i> Простой подсчет трафика через роутер Cisco	118
<i>А.А. Мясичев, В.М. Полозова.</i> Использование протокола STP для устранения петель и резервирования связей.....	123
<i>Б.А. Железко.</i> Инструментальные методы в бизнес-образовании	133
<i>О.В. Клочко.</i> Розвиток педагогічних здібностей у навчанні інформаційно-комунікаційним технологіям студентів економічних спеціальностей.....	137
<i>О.Г. Онишко.</i> Розвиток творчих здібностей студентів економічних спеціальностей на заняттях з інформатики.....	142
<i>О.Ф. Клименко, Ю.М. Красюк.</i> З досвіду використання методу проектів у процесі навчання інформатики студентів економічних спеціальностей.....	147
<i>О.Г. Смілянець.</i> Технологія розв'язування творчих фінансово-економічних задач на заняттях з інформатики	153
<i>О.М. Боско, Н.В. Гринь.</i> Дисципліна «Прикладне програмне забезпечення» як подолання розриву між навчанням та реаліями життя	158
<i>А.В. Бабич.</i> Создание методических материалов по Telelogic TAU G2 с помощью Microsoft Producer	161
<i>А.В. Шматко, И.А. Яковлева.</i> Компьютерные презентации в преподавании информатики в техническом вузе	166
<i>Н.С. Жуков, Л.Л. Жукова.</i> Подходы к разработке виртуальной лаборатории для дисциплины “Теория автоматического управления”	170
<i>Р.М. Балабай, Д.В. Рябчиков.</i> От ручного управления к программному	175
<i>М.К. Аль Мади, В.А. Андриенко, В.Г. Рябцев.</i> Этапы верификации конфигурируемых на кристалле цифровых систем	179
<i>К.Ю. Васильев, Д.В. Суцевский, Ю.С. Алпатова, М.Г. Охременко.</i> Рациональный подход к определению количества информации	183
<i>Т.А. Холошня.</i> К вопросу уровневой дифференциации изучения курса информатики в высшей школе.....	186
<i>С.Н. Чаплыгина.</i> Изложение раздела «Системология» дисциплины «Информатика и системология».....	190
<i>Н.В. Шаповалова, Т.В. Ломаева.</i> Математичне і комп'ютерне моделювання як фактор оптимізації навчального і науково-дослідного процесів..	193
<i>Л.Е. Гризун.</i> Використання комп'ютерного моделювання при викладанні курсу «Моделювання економічної динаміки».....	198
<i>Ю.С. Матвієнко.</i> Програмне забезпечення інтегрованого курсу математичного моделювання	202
<i>И.А. Кривель, А.Н. Моргун.</i> Изучение структур данных в курсе програм-	

мировання комп'ютерних спеціальностей вищої школи.....	205
<i>О.А. Смілько.</i> Сучасні тенденції в оформленні інтерфейсів прикладних програмних засобів	210
<i>Ю.О. Дорошенко, В.О. Очеретний.</i> Міжпредметні зв'язки та формування елементарних умінь роботи у середовищі AutoCAD.....	221
<i>Ю.А. Супрунова.</i> Изучение основ работы с графическими редакторами в курсе «Компьютерная графика»	231
<i>Р.В. Ліхачов.</i> До питання підвищення ефективності ознайомчого курсу роботи з текстовими процесорами	234
<i>І.Є. Фільо.</i> Методологічні підходи до застосування комп'ютера у навчальній діяльності студентів інженерних спеціальностей	237
Розділ II. Професійна підготовка вчителя інформатики.....	242
<i>Т.П. Кобильник.</i> На шляху інформатизації освіти	243
<i>Л.Л. Сушенцева, О.Ю. Штиркова.</i> Оптимальний вибір методів навчання з метою підвищення ефективності процесу навчання на уроках інформатики.....	248
<i>Т.В. Волкова.</i> Управління підготовкою майбутнього викладача спеціальних дисциплін.....	252
<i>І.М. Лукаш.</i> Пояснювально-ілюстративний метод в концепції електронного навчання інформатики	259
<i>С.В. Кондратенко.</i> Здобутки та проблеми впровадження програми «Intel® Навчання для майбутнього».....	265
<i>О.М. Туравініна.</i> «Хто володіє інформацією, той володіє світом»	268
<i>А.П. Забарна.</i> Інтерактивні технології навчання на уроках з основ алгоритмізації та програмування	271
<i>І.В. Настенко.</i> Використання інноваційних методів навчання в курсі програмування.....	280
<i>Т.В. Кабанова.</i> Межпредметные связи при обучении информатике в средней школе	287
<i>Л.І. Петрушина.</i> Використання комп'ютерних технологій на уроках у початкових класах.....	293
<i>О.Р. Гарбич.</i> Підготовка до олімпіад з програмування студентів педагогічних ВНЗ.....	296
<i>Д.М. Велчко.</i> Використання Е-технологій у сучасному освітньому просторі.....	299
<i>Н.І. Дорошенко.</i> Методика навчання креслення у професійно-технічному навчальному закладі: формування графічних понять	303
<i>А.Л. Столяревская, А.Ю. Кузнецов.</i> Редактор фракталов и его применение на занятиях по информатике	312
<i>Л.О. Лісіна, І.С. Мінтій.</i> Вивчення логічних основ ЕОМ в шкільному курсі інформатики.....	320
<i>Л.О. Лісіна, І.С. Мінтій.</i> Спрощення логічних виразів за допомогою карт	

Карно в шкільному курсі інформатики	327
<i>А.Л. Столяревская.</i> Использование новых информационных технологий в курсе «Основы искусственного интеллекта»	332
Розділ III. Захист інформації та інформаційна безпека	338
<i>Н.М. Бугайова.</i> Інтернет-адикція, як форма залежної поведінки	339
<i>А.О. Москаленко, Ю.П. Бендес, Ю.І. Мазнюк.</i> Реалізація міжпредметних зв'язків розділів «Геометрична оптика» та «Технічні канали витоку інформації»	343
<i>О.О. Кононова, Н.В. Моїсеєнко.</i> Навчання криптографічним методам захисту інформації	348
<i>В.М. Джулій.</i> Аналіз ненадійності криптосистем	353
<i>Н.Е. Губенко, Д.В. Рябченко.</i> Сравнительный анализ криптоалгоритмов DES и авторского алгоритма Epigma-R10	360
<i>И.А. Кривель, А.Н. Моргун.</i> Хэш-функция на основе двоичного дерева	364
<i>О.О. Гайша.</i> Дослідження можливості відтворення виконуваних файлів персональних ЕОМ	368
<i>Ю.В. Хмельницький.</i> Методи захисту процесу передачі інформації в комп'ютерних системах та мережах	373
<i>А.А. Кулик.</i> Організація захисту Web-ресурсів за допомогою технології ASP.NET	378
<i>А.Н. Бакал.</i> К вопросу о методах борьбы со спамом	384
<i>Л.М. Петренко.</i> Інформаційна безпека як основа управління господарською діяльністю	393
Наші автори	397

Наукове видання

**Теорія та методика навчання
математики, фізики, інформатики**

Випуск VI

В 3-х томах

Том 3

Підп. до друку 29.03.06

Папір офсетний №1

Ум. друк. арк. 23,72

Формат 80×84 1/16

Зам. №1-2903

Наклад 300 прим.

Жовтнева районна друкарня
50014, м. Кривий Ріг, вул. Електрична, 5
Тел. (0564) 664381

E-mail: cc@kpi.dp.ua