

Міністерство освіти та науки України
Криворізький державний педагогічний університет

Теорія та методика
навчання математики,
фізики, інформатики

Збірник наукових праць

Том 3

Кривий Ріг
Видавничий відділ КДПУ
2001

Теорія та методика навчання математики, фізики, інформатики: Збірник наукових праць: В 3-х томах. – Кривий Ріг: Видавничий відділ КДПУ, 2001. – Т. 3: Теорія та методика навчання інформатики. – 209 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 та впровадження нових інформаційних технологій у викладання базових дисциплін.

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

Редакційна колегія:

В.М. Соловійов, доктор фізико-математичних наук
Є.Я. Глушко, доктор фізико-математичних наук
О.І. Олейніков, доктор фізико-математичних наук
Я.В. Шрамко, доктор філософських наук, професор
В.І. Хорольський, доктор технічних наук, професор
О.А. Учитель, доктор технічних наук, професор
І.О. Теплицький, відповідальний редактор
С.О. Семеріков, відповідальний секретар

Рецензенти:

В.М. Назаренко – д-р техн. наук, професор, завідувач кафедри інформатики, автоматики та систем управління Криворізького технічного університету
А.Ю. Ків – д-р фіз.-мат. наук, професор, завідувач кафедри теоретичної фізики Південноукраїнського державного педагогічного університету (м. Одеса)

Затверджено Вченою радою Криворізького державного педагогічного університету (протокол №7 від 08.02.2001 р.)

ISBN 966-8302-42-6

К ВОПРОСУ О ПРИКЛАДНЫХ ЗАДАЧАХ В КУРСЕ СОБЫТИЙНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ В ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЯХ

С.В. Артюшенко, Н.В. Моисеенко
г. Кривой Рог, Криворожский государственный педагогический
университет

Стремительное развитие компьютерной техники, расширение сферы ее использования и круга пользователей несколько изменили понятие о хорошем программном обеспечении. В последнее время удобство пользования и внешняя привлекательность программы, прежде всего, определяются наличием удобного интерфейса взаимодействия с пользователями. Особенно это важно для интерактивных программ типа игровых, программ моделирования, тестирования и т. д. Во время работы таких программ пользователь активно влияет на ход их выполнения, выбирая команды, отвечая на вопросы. Поэтому в целях создания дружественного интерфейса желательно все действия по вводу информации свести к выбору одного из предлагаемых вариантов. Всегда быстрее и проще выбрать значение из списка, чем набирать его на клавиатуре. Кроме того, во время работы пользователю часто предоставляется контекстная помощь, возможность открытия и сохранения файлов и т. п. Все это приводит к тому, что зачастую создание пользовательского интерфейса отнимает у программиста больше времени и сил, чем содержательная часть проекта.

Несмотря на расширение возможностей традиционных языков программирования, только с появлением визуальных языков типа Visual Basic, Delphi, Visual C++ произошло качественное изменение в решении этого вопроса. Разработчик использует при создании своего проекта библиотеку объектов элементов интерфейса (окон, меню, кнопок, полос прокрутки), каждый из которых обладает совокупностью характеристик, которые можно изменять на разных этапах создания и выполнения программы. При этом нет необходимости писать сотни операторов вручную, программист описывает только реакцию на различные события, связанные с этими объектами. Создание интерфейса в визуаль-

ных средах программирования подобно работе с конструктором: из набора деталей собирается целая конструкция, но при этом можно изменять не только положение, но и цвет и размер деталей и даже создавать некоторые новые детали по своему вкусу. При этом разработчик все время может видеть, каков будет конечный результат его работы, что существенно экономит время, сокращая количество тестовых запусков проекта.

При изучении событийно-ориентированного программирования большое значение имеет приобретение навыков создания проектов. Предлагаемые в большинстве зарубежной литературы «игрушечные» примеры не вызывают энтузиазма у студентов и не дают возможности рассмотреть все тонкости программирования. Мы считаем, что в целях обучения для студентов значительно полезнее создать хотя бы не очень большой, но реальный проект. При этом будущие программисты могут прочувствовать все этапы разработки, возникающие при этом трудности и возможности их разрешения.

Приведем пример возможной прикладной задачи: *требуется создать программу, позволяющую определять уровень субъективного контроля с помощью некоторой психологической методики.* В данном случае методика тестирования представляет собой представление тестируемому специально подобранных утверждений, к которым он определяет свое отношение. Тест можно провести с помощью биполярной (возможен выбор из двух полярных ответов) или шестибалльной (предлагаются шесть вариантов ответов) шкалы. Первая шкала более грубая, вторая – более гибкая, точная.

Консервативно настроенная часть наших читателей может здесь заявить, что принципиально возможно решить поставленную задачу средствами языка С, не прибегая ни к объектно-ориентированному, ни к событийно-ориентированному программированию. Справедливости ради отметим, что это верно, но преимущества программы разработанной для операционной системы Windows вполне очевидны:

- удобство выбора вариантов ответа с помощью мыши;
- простота реализации многодокументного интерфейса;
- простота технологии сохранения документа;

- возможность работы с буфером Windows для обмена данными между документами и др.

В итоге на Visual C++ была создана программа **Locus**. После её запуска появляется изначально пустой список (документ, в котором затем будут храниться результаты тестирования). Чтобы получить информацию о методике тестирования предлагается команда главного меню *Помощь/Методика тестирования*.

При запуске тестирования появляется окно, в котором содержится утверждение, варианты ответа и несколько вспомогательных кнопок (Рис. 1).

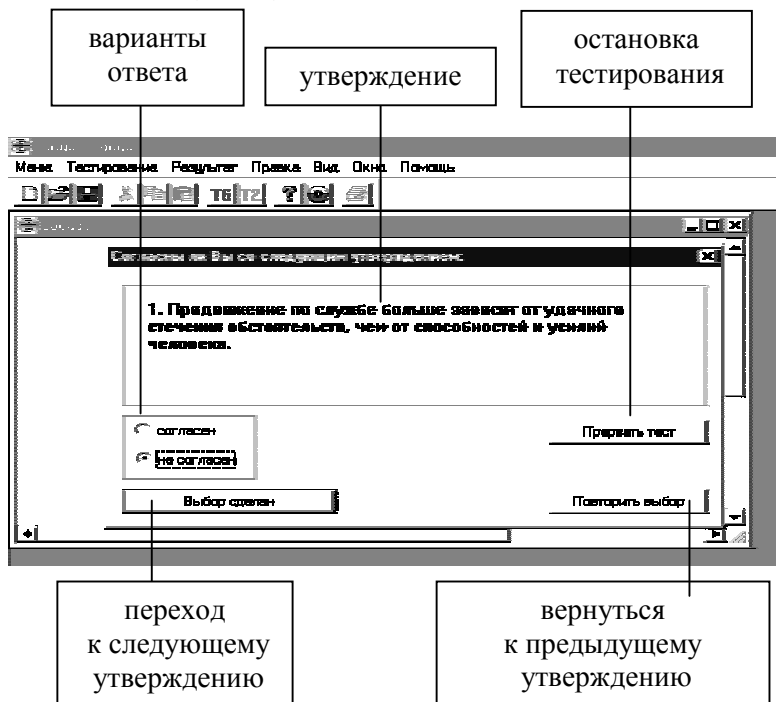


Рис. 1. Главное окно тестирующей программы.

После прохождения теста появляется окно «Информация о тестируемом», в поля которого необходимо внести данные об имени, возрасте, роде занятий испытуемого. После закрытия этого окна в конец активного списка добавится ещё один результат (Рис. 2).

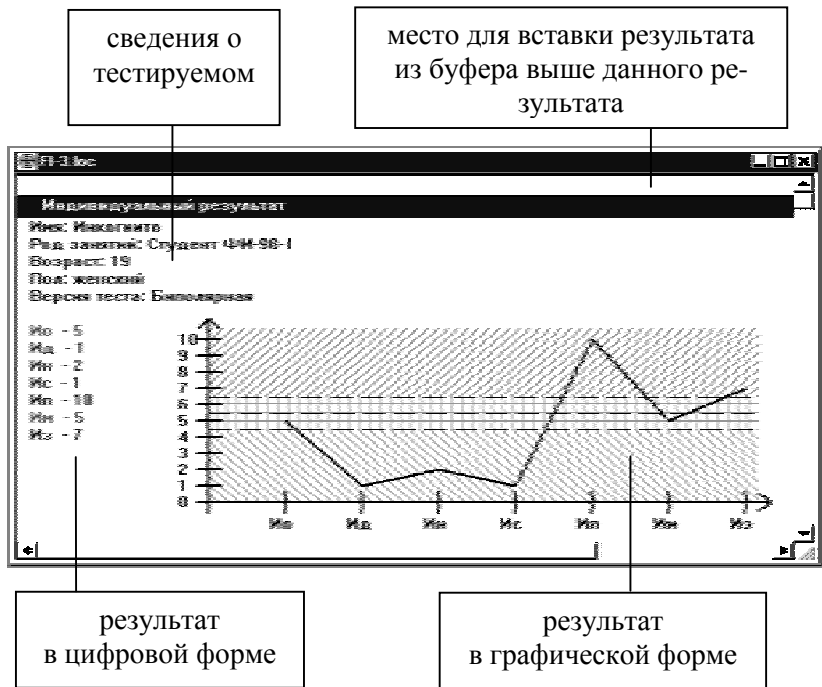


Рис. 2. Окно с результатами тестирования.

Проинтерпретировать результат можно, выбрав команду *Помощь/Помощь при интерпретации* или *Помощь/Интерпретация результатов* (для более детального толкования). Программа снабжена строкой-подсказкой, содержащей информацию о количестве индивидуальных и средних результатов в списке.

Программа поддерживает стандартные операции редактирования: копировать результат, вырезать, удалить, вставить и некоторые другие. Они представлены в пункте *Правка* главного меню. В пункте *Результат* содержатся команды, позволяющие находить средний результат по всему списку и отдельным параметрам, найти результат в списке и изменить данные, которые он содержит.

Также предусмотрена поддержка альтернативных способов вызова команд главного меню посредством горячих клавиш или кнопок на панели инструментов.

Для запуска тестирующей программы необходимо запустить файл **locus.exe**, а также она запускается автоматически при двойном щелчке на любом файле с расширением **.loc**.

Подытоживая все вышесказанное, отметим, что средства визуального проектирования существенно упрощают процесс разработки программного обеспечения. Visual C++ позволяет даже на начальном этапе изучения событийно-ориентированного программирования создать проект, содержащий практически все элементы полноценного Windows-приложения. При написании программы перед разработчиками возникает ряд сопутствующих задач, решение которых требует всё более и более глубокого изучения объектно-ориентированного программирования и демонстрирующих применение на практике тех или иных элементов ООП. Это делает обучение наглядным, насыщенным проблемными ситуациями, креативным, что положительно отражается на качестве усвоения знаний, их прочности, на развитии умений и навыков программирования.

ДО ПИТАННЯ ІНТЕНСИФІКАЦІЇ ПРОЦЕСУ НАВЧАННЯ КОМП'ЮТЕРНИМ ІНФОРМАЦІЙНИМ ТЕХНОЛОГІЯМ

О.П. Бесклінська, С.І. Міхневич

м. Київ, Київський державний лінгвістичний університет

У Київському державному лінгвістичному університеті вивчається цикл дисциплін “Інформаційні технології в менеджменті”. Метою даного курсу є навчити студентів застосовувати можливості сучасних інформаційних технологій, без котрих зараз неможливе успішне ведення бізнесу в невиробничій сфері. Оволодіння студентами основних навичок роботи зі стандартними пакетами прикладних програм MS Office значно полегшують їх майбутню роботу як менеджера при створенні та обробці ділової документації, допомагають в прийнятті рішень та ефективному плануванні роботи, надають можливість швидкого пошуку і обміну інформацією (використання електронної пошти та Інтернету) та її оформлення.

Одним з розділів даного курсу є вивчення текстового процесору Word. Згідно з аналізом діяльності персоналу офісів та бюро менеджери повинні опанувати певними практичними навичками, які б дозволяли сформуванню базовий рівень знань, що графічно представлені на рис. 1.

Хоч зараз виходить багато різноманітної літератури для роботи з процесором Word, не завжди студенти можуть дозволити собі її придбати. Як правило книжки, що розраховані на професійного користувача – а саме такий рівень передбачає ефективне використання інструментів Word – дорого коштують. Крім того, українською мовою виходить лише обмежена кількість видань і ці видання не враховують професійного напрямку. Щоб оволодіти певними навичками роботи треба мати доступ до комп'ютера. Нажаль, поки що лише окремі студенти мають комп'ютер вдома чи в гуртожитку.

Враховуючи викладені вище обставини, має бути розроблена методика викладання матеріалу, яка б дозволяла за конкретний проміжок часу отримати максимум стійких теоретичних знань та практичних навичок. Зрозуміло, якщо спланувати заняття у вигляді лекцій з конспектуванням матеріалу, то їх ефективність

буде незадовільною, не кажучи вже про повну відсутність практичних навичок. Низькою вона буде і тоді, якщо лекції проводити із застосуванням світлопроекційної та відеотелевізійної техніки. Тому за даним циклом дисциплін заплановані лише практичні заняття. Заняття проводяться у комп'ютерних класах КДЛУ, де кожний студент забезпечується окремим робочим місцем за комп'ютером.

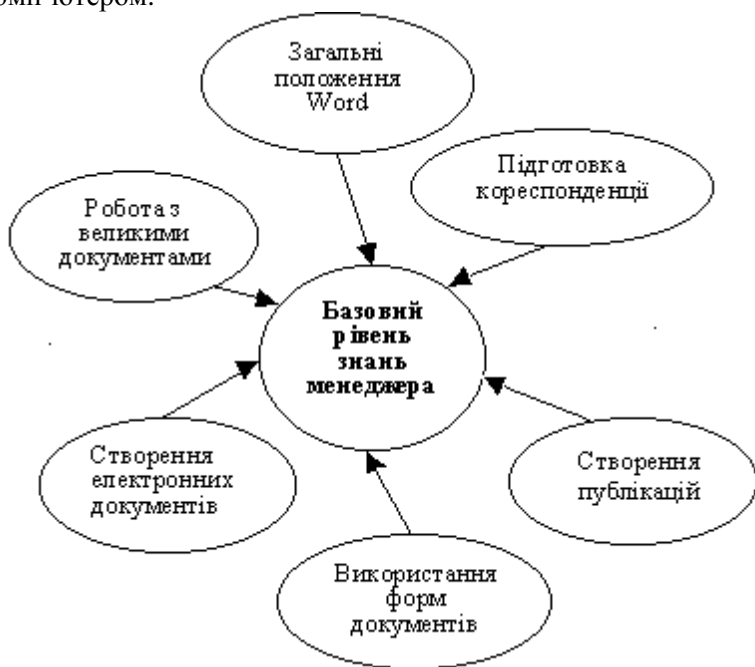


Рис. 1.

Починати заняття потрібно з викладання основного теоретичного матеріалу. Одночасно студентам видаються методичні посібники, в яких стисло подаються основи тематики, що вивчається. Завдання студента на цьому етапі – це розуміння та засвоєння наданої викладачем інформації. Подання навчального матеріалу здійснюється згідно з план-конспектом методичного посібника.

У план-конспекті теоретичний матеріал до теми, що розглядається на занятті, розбивається на невеликі питання згідно структури:

- назва питання;
- пояснення декількох способів виконання, що пропонує Word (з урахуванням принципу роботи Word, коли пропонується декілька способів виконання тієї чи іншої операції);
- стислий огляд дій;
- типовий приклад чи приклади.

Далі за теоретичною частиною план-конспекту містяться вправи для самостійного виконання. Ці вправи мають бути показовими, тобто такими, які краще сприяють засвоєнню матеріалу. Важливо також, щоб вони були не громіздкими, але змістовними. Тобто такими, щоб на їх виконання студент не витрачав багато часу, але зміг досконалим та поглиблено розібратися зі способами їх виконання. Крім того, вправи повинні відображати професійну орієнтацію майбутніх фахівців.

Основною, суттєвою частиною практичного заняття є самостійна робота студента. До активної самостійної роботи необхідно залучати кожного студента. Щоб самостійна робота студента була продуктивною, треба належним чином підготувати його теоретично. Тобто розробити методичні посібники для проведення занять таким чином, щоб студент зміг опанувати базовим теоретичним рівнем і далі вже набував знання і навички згідно своїх можливостей та рівнем загальної компетенції.

Процес навчання можна поділити на такі етапи:

I. *Оголошення теми заняття, його мети.*

II. *Викладання матеріалу викладачем.* Викладач подає теоретичний матеріал із застосуванням світлопроекційної техніки та/або засобів Multimedia (на базі CD-ROM), які дозволяють забезпечити комплексний вплив на учня завдяки статичній та динамічній наочності.

III. *Знайомство з план-конспектом.*

IV. *Осмислення матеріалу.* Студент може звернутись за допомогою до викладача, або до план – конспекту. На даному етапі навчання зворотній зв'язок викладач ↔ студент є перш за все характеристикою стану студента (готовність до заняття, увага, психологічний стан, загальний рівень компетенції), на підставі яких викладач вибирає тактику навчальної діяльності.

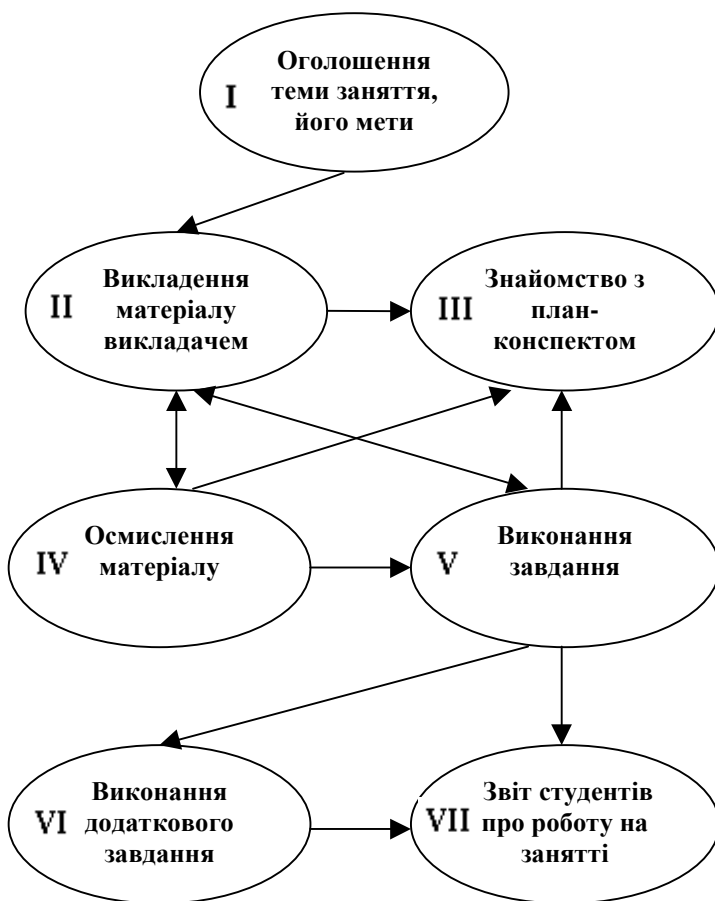
V. *Виконання завдання.* Студент може звернутись за до-

помогою до викладача, або до план – конспекту.

VI. Виконання додаткового завдання. На цьому етапі студент, який впорався з обов'язковою частиною, виконує додаткове завдання.

VII. Звіт студентів про роботу на занятті.

Схема проведення практичного заняття



Приклад план-конспекту до теми “Табулятори”

Тема: табулятори

Мета: навчитися застосовувати маркери табуляції для створення кореспонденції, публікацій, форм документів



Щоб встановити маркер табуляції необхідно:

1. Вибрати тип табуляції за допомогою кнопки на лінійці:



2. Клацнути лівою кнопкою миші в потрібному місці лінійки форматування.

При натисканні клавіші Tab курсор пересувається до наступної позиції табуляції.

Типи табуляції:

-  Left Tab
-  Center Tab
-  Right Tab
-  Decimal Tab
-  Bar Tab

Обрати команду **Формат (Format) ⇒ Табуляція (Tabs)**.



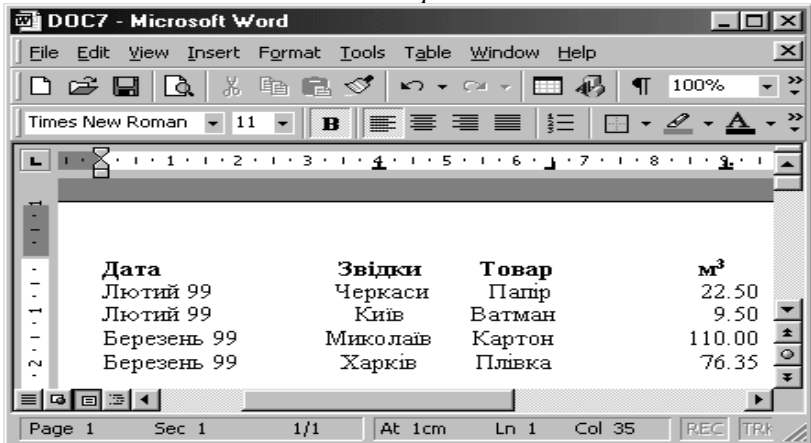
1. У вікні **Tab stop position** вказати відстань, на яку треба перенести курсор.

2. У вікнах **Alignment** вказати тип табулятора.

3. У вікнах **Leader** вказати вид заповнювача.

4. Натиснути клавішу **Set**. Натиснути **OK**.

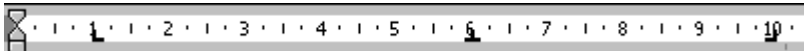
Типовий приклад:



Зауваження ! За замовчуванням задається табуляція по лівому краю

Вправи для самостійного виконання:

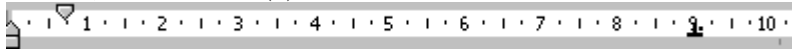
1. За допомогою лінійки встановити маркери табуляції та надрукувати текст. Tab Left на 1 см; Tab Center на 6 см; Tab Right на 10 см.



Курси іноземних мов:

Адреса	мова	термін навчання
пр. Перемоги, 32	англійська	1 рік
вул. Потапова, 12	німецька	2 роки
вул. Шевченко, 23	італійська	3 роки

2. За допомогою меню Tabs виконати завдання. Tab Decimal на 10 см, заповнювач (3):



Тарифи Мобільний Інтернет:

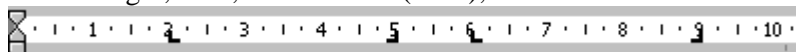
War-доступ до Інтернет-----0,15
Традиційний доступ до Інтернет-----0,25

3. Виконати завдання:

Tab Right, 2 см, заповнювач (2);

Tab Left, 5 см, заповнювач (none);

Tab Right, 6 см, заповнювач (2);
 Tab Right, 9 см, заповнювач (попе);



Αα.....Альфа Ββ Бета
 Γγ.....Гамма Δδ.....Дельта

4. Виконати завдання:

**Outgoing calls within Ukraine,
 all incoming calls and incoming message to voice mail
 “Starmail”**

Time period	Monday-Friday	Saturday,Sunday
From 8.00 till 18.00	0.48	0.24
From 18.00 till 22.00	0.24	0.24
From 22.00 till 8.00	0.15	0.15

З ДОСВІДУ ВИКЛАДАННЯ КУРСУ «ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ»

М.П. Білан

м. Кривий Ріг, Криворізький обласний ліцей-інтернат для сільської молоді

Багато понять шкільного курсу інформатики доцільно формувати емпіричним шляхом.

Так, одним з фундаментальних понять в курсі інформатики є поняття «алгоритм» і «виконавець алгоритму». Спочатку поняття алгоритму пояснюється на прикладах алгоритмів, виконувати які може людина. Такі приклади дозволяють спиратися на інтуїтивні поняття учнів, але при складанні програм для ЕОМ ті ж інтуїтивні поняття стають перешкодою, оскільки учні з трудом усвідомлюють обмеженість набору допустимих дій ЕОМ (і будь-якого конкретного виконавця). Щоб подолати цю перешкоду, доцільно розглянути виконавців з різними наборами допустимих дій (ця ідея йде від Г.А. Звенигородського, О.Г. Гейна, В.Г. Житомирського). Учні повинні чітко собі уявляти, чи може даний виконавець розв'язати запропоновану задачу. Тобто, говорячи про виконавця алгоритму, потрібно говорити про чітко окреслену систему команд виконавця. Для школярів такий підхід не новий. Наприклад, шкільні задачі на побудову за допомогою циркуля і лінійки чи однієї лінійки.

Для закріплення поняття виконавця використовуємо задачі з посібника [1]. Наприклад

1. Виконавець вміє: множити число на 2, збільшувати число на 1.
А. Складіть для цього виконавця алгоритм одержання числа 100 з одиниці.
Б. Скільки дій в найпростішому з таких алгоритмів?
2. Зловмисник поміняв місцями дії в алгоритмі обчислення середнього арифметичного трьох чисел:
 - 1) присвоїти a значення $(a+b+c)/3$;
 - 2) ввести a, b, c ;
 - 3) вивести a .

Встановити правильний порядок дій.

3. Дано три листки паперу. Виконавець бере лист паперу і розрізає його на чотири рівні частини і кладе їх назад. Яка кількість листків може бути одержана в результаті його роботи?
4. Яких допустимих дій ви надали б виконавцям:
 - 1) продавцю;
 - 2) регулювальнику дорожнього руху;
 - 3) клоуну в цирку.

(Ця задача потребує колективного обговорення, в ході якого будуть виявлені найбільш суттєві допустимі дії).

5. Чи можна створити виконавця, який все може?

6. Дано три сторони трикутника a, b, c .

А. Скласти для Обчислювача алгоритм, виконуючи який, він повідомить півпериметр трикутника, його площу, радіус вписаного кола.

Б. Скласти алгоритм розв'язку цієї задачі, в якому використовуються тільки чотири змінні.

В. Скласти алгоритм розв'язку цієї задачі, в якому використовуються тільки три змінні.

При вивченні базових структур алгоритмів використовуємо так звані «алгоритми від зловмисника», які містять логічні помилки. Робота над такими алгоритмами дає змогу учням усвідомити, що для перевірки правильності алгоритму потрібно перевіряти не тільки виконання умови, але й її не виконання. Прикладів таких алгоритмів достатньо приведено в [1]. Наприклад,

1) алг картопля	2) алг голод
поч	поч
якщо картопля почищена	поки не розв'язані всі задачі
то зварити картоплю	пц
інакше почистити картоплю	розв'язати наступну задачу
все	піти погуляти
з'їсти картоплю	кц
кін	кін

Не можна навчити алгоритмізації учнів, якщо тільки пояснювати алгоритми, а не навчити їх самостійно виконувати ці ал-

горитми. Звичайно, виконання алгоритмів потребує багато часу. Але набагато важливіше розв'язати хоч і не багато задач, але так, щоб вони були зрозумілі учням. Тому на початку вивчення таких базових понять як алгоритми розгалуження, циклів, допоміжних алгоритмів, опрацювання табличних величин складаємо протоколи ручного виконання алгоритмів. Ідея так званого ручного виконання алгоритмів йде ще від А.П. Єршова і В.М. Монахова і добре викладена в [2].

Перед написанням алгоритму потрібно попросити учнів залишити півсторінки справа вільною. Це місце знадобиться для запису кроків виконання. Рекомендується в одному рядку писати не більше однієї команди.

Приклад 1. Програма обчислення коренів квадратного рівняння і виконання її для рівнянь:

а) $x+2x+1=0$; б) $3x+2x-1=0$; в) $2x+x+1=0$.

Перед виконанням команди розгалуження треба підкреслити, що перевірка умови – це перший крок виконання, а виконання однієї із серій команд – другий крок. І цей другий крок може бути зовсім не простим, а містити декілька простих команд, а також і складених.

<code>#include<stdio.h></code>			
<code>main()</code>			
<code>{</code>			
<code>float a,b,c,d,x1,x2;</code>			
<code>scanf("%f",&a);</code>	a = 1	a=3	a=2
<code>scanf("%f",&b);</code>	b = 2	b=2	b=1
<code>scanf("%f",&c);</code>	c = 3	c=-1	c=-1
<code>d= b*b-4*a*c;</code>	d = 0	d = 16	d = -7
<code>if (d>=0)</code>	0>=0?	16>=0?	-7>=0? (ні)
<code>{</code>	(так) ↓	(так) ↓	↓
<code>x1=(b+sqrt(d))/(2*a)</code>	x1=1	x1=1/3	
<code>x2=(-b-sqrt(d))/(2*a);</code>	x2=1	x2=-1	
<code>printf("%f, %f",x1,x2);}</code>	1, 1	1/3, -1	
<code>else</code>			Коренів немає
<code>printf("Коренів немає");</code>			
<code>}</code>			

Приклад 2. Написати програму обчислення суми натуральних чисел від 1 до 10.

При виконанні циклічних алгоритмів, щоб розібратися в алгоритмі і перевірити його, достатньо виконати перші 2–3 повторення і останні проходи циклу.

```
#include<stdio.h>
main()
{int N,S;
```

N = 1;	N = 1				
S = 0;	S = 0				
while(N<=10)	1<=10? (так)	2<=10? (так)	...	10<= 10? (так)	11<=10 (ні)
{S = S + N;	S = 0 + 1	S=0+1+2	...	S=0+1+...+10	
N = N + 1;	N = 2	N = 3	...	N=11	
}					
printf(“%d”,S);					
}					

Виконання алгоритмів є також і ефективною формою контролю знань учнів, бо дозволяє виявити, чи розуміє учень весь алгоритм, чи хоча б окремі його команди. Виконання алгоритмів можна зробити і засобом активізації класу, для цього до виконання залучають якомога більше учнів. Наприклад, виконання алгоритмів розгалуження при різних вхідних даних, чи виконання окремих повторень циклу можна доручити різним учням. Виконання починає сам вчитель, а потім до цієї роботи залучаються і інші учні. Воно має і велике виховне значення, бо привчає учнів до акуратності, уважності, вміння доводити розпочату справу до кінця, розвиває логічне, раціональне мислення.

Література:

1. Гейн А.Г. и др. Основы информатики и вычислительной техники. – Свердловск: Изд-во Урал. ун-та, 1989. – 272 с.
2. Исаков В. Исполнение алгоритмов // Информатика и образование. – 1987. – №2. – С. 30–43.

ВИКОРИСТАННЯ СЦЕНАРІЇВ ЕЛЕКТРОННОЇ ТАБЛИЦІ EXCEL ДЛЯ АНАЛІЗУ ДАНИХ СТУДЕНТАМИ ВИЩОГО НАВЧАЛЬНОГО АГРАРНОГО ЗАКЛАДУ ТА КОНТРОЛЬ ЗНАНЬ

В.Г. Борищенко

м. Суми, Сумський державний аграрний університет.

Сучасні вимоги до знань та вмій сучасного спеціаліста будь-якої спеціальності – володіти навичками роботи на персональному комп'ютері для оформлення результатів роботи, проведення дослідницької роботи, підготовки проектів.

Курс “Інформатики” у вищих закладах передбачає вивчення основних програмних засобів: системного програмного забезпечення, засобів розробки програм та пакетів прикладних програм. В межах професійного напрямлення в цьому курсі необхідно акцентувати увагу студентів на застосування інформаційних технологій у професійній діяльності.

Студентам СумДАУ викладається ця дисципліна на 1-ому та 2-ому курсах в залежності від спеціальності. Далі на старших курсах, і тільки для студентів-економістів, вивчається окремо дисципліна “Інформаційні системи”. Тому на заняттях з “Інформатики” потрібно вивчати методи застосування різних програм та області їх застосування, показувати вирішення проблем, які можуть виникати при використанні персонального комп'ютера та при вирішенні професійних завдань.

Широкі можливості для цього існують при вивченні електронної таблиці EXCEL WINDOWS2000. Так, студентам 2-го курсу викладається питання аналізу даних за допомогою електронної таблиці EXCEL. Вони вже вміють виконувати розрахунки і мають основні навички роботи з електронною таблицею EXCEL. Для певної економічної задачі, на наш погляд, дуже важливим є показати, як впливає на кінцевий показник зміна деяких величин, як можна створити модель для відображення цих показників. Таким засобом в EXCEL є Диспетчер сценариев.

Вивчаючи цей матеріал, потрібно використовувати міжпредметні зв'язки, застосовувати поняття інших профільних дисциплін. Студенти набувають умінь та навички використання ін-

формаційних технологій, навчаються замислюватися над реально набутими знаннями, побачити великі можливості застосування електронної таблиці EXCEL.

Починати вивчення засобу Диспетчер сценариев потрібно з визначення основних термінів: сценарій, модель “что-если”, клітини (чарунки), що змінюються.

Пропонуємо проведення заняття зі студентами, виконуючи практичну роботу. Виходячи з досвіду викладання, найкраще виконати такий аналіз розв’язуючи конкретну задачу (з невеликою кількістю розрахунків).

Практична робота. Тема: Диспетчер сценариев електронної таблиці EXCEL

Студентам надається завдання: існує документ наступної форми – таблиця окремих видів продукції по господарству (рис.1).

Вид продукції	Виторг	Собівартість	Прибуток	Рівень рентабельності, %
Зерно	19180	9660	*	*
Цук.буряки	13100	7000	*	*
Картопля	6140	2660	*	*
Молоко	20922	20090	*	*
Овочі	9275	6325	*	*
Всього:			*	

Рис. 1. Таблиця окремих видів продукції по господарству.

Необхідно:

1) Визначити рівень рентабельності окремих видів продукції та прибуток по господарству в цілому, використовуючи формули:

$$R=P/C*100; P=B-C.$$

2) Виконати аналіз впливу зміни собівартості різних видів продукції та їх виторгу на загальний прибуток господарства, використовуючи Диспетчер сценариев електронної таблиці EXCEL. Створити звіт за сценаріями. За даними звіту побудувати, наприклад, гістограму зміни загального прибутку.

Приклад наведеного завдання тільки надає змогу студентам розібратися з тим, яка існує технологія виконання поставленої

задачі. При підготовці практичних робіт з викладачами інших профільних дисциплін можна чітко визначитися в тому, які основні економічні та сільськогосподарські завдання (що пов'язані з розрахунками) вирішують вони при вивченні своїх дисциплін.

І, звичайно, вивчення нового матеріалу потрібно закріпити невеличким контрольним опитуванням, до якого можна включити наступні запитання:

1. Яке основне призначення електронної таблиці EXCEL?
2. Які типи даних існують в EXCEL?
3. Яка адресація клітин використовується в EXCEL?
4. Як виконати розрахунки в EXCEL?
5. Для чого застосовується аналіз “что-если” в EXCEL?
6. Дати визначення сценарію в EXCEL.
7. Як в EXCEL створити окремий сценарій?
8. Як в EXCEL вивести окремий сценарій на екран монітора?
9. Як створити звіт за сценаріями в EXCEL? Для чого потрібен звіт за сценаріями?
10. Які клітини електронної таблиці EXCEL потрібно вибирати в якості даних?

Ці питання надають можливість усвідомлено сприймати новий матеріал. Але краще для засвоєння цього методу аналізу даних буде виконання індивідуального завдання. Виконавши це завдання можна протестувати кожного студента різними видами тестування. Кожне тестування реєструється в журналі оцінок з повною інформацією про студента: тема, прізвище, факультет, спеціальність, група, оцінка, дата.

На основі досвіду викладання “Інформатики” на прикладі методики, яка надана вище, можна зробити висновок, що використання цього методу аналізу даних в EXCEL та використання цієї методики формує у студентів навички застосування ПЕОМ у майбутній професійній діяльності, дозволяє вивчати складний матеріал, враховуючи індивідуальні можливості, та контролювати знання.

ТЕСТОВІ ЗАВДАННЯ ЯК ІНСТРУМЕНТ ДЛЯ ПОРІВНЯННЯ ЗНАНЬ УЧНІВ З ІНФОРМАТИКИ

Л.С. Возняк, Б.М. Дрінь, В.В. Дудка, О.М. Дудка
м. Івано-Франківськ, Прикарпатський університет імені
В. Стефаника

... В конструюванні
тесту поєднані наука і ми-
стецтво...

Л. Клейн

Характерною рисою сучасного періоду соціально-економічного і науково-технічного розвитку суспільства є зміна домінуючих видів людської діяльності. Центр ваги зміщується на ті з них, котрі пов'язані із сучасною інформаційною технологією, що розширює можливості людей в управлінні технологічними і соціальними процесами. Інформатизація освіти вносить істотні зміни в діяльність людей, педагогічний процес і охоплює всі ланки системи освіти, її установи та органи управління.

Загальними тенденціями у розвитку інформатизації освіти є:

- розширення сфери використання засобів нових інформаційних технологій в освітньому процесі;
- перехід від епізодичного до систематичного застосування засобів нових інформаційних технологій, їх постійного використання при вивченні освітніх предметів;
- поява принципово нових засобів навчання (навчальних і ігрових середовищ, інтелектуальних наставників, текстових редакторів, експертних навчаючих систем), інтелектуалізація навчаючих систем;
- використання засобів нових інформаційних технологій у позакласній та позашкільній роботі, що сприяє наближенню навчальної діяльності до дослідницької, пошукової діяльності та подоланню розриву між навчальною і професійною діяльністю;
- формування основ інформаційної культури при вивченні різних освітніх предметів.

Водночас мають місце і певні негативні прояви складових

процесу інформатизації освіти. Вони зумовлені передусім недостатньо розробленими теорією та технологією проектування навчання, слабким узагальненням практичного досвіду. Програмне забезпечення не завжди обґрунтоване потребами педагогічного процесу. Традиційні форми контролю знань далеко не в повній мірі реалізують свої функції.

Зупинимось більш детально на питанні контролю.

Контроль знань та вмінь учнів є важливим компонентом навчального процесу в будь-яких закладах освіти. Ефективність навчального процесу залежить від організації зворотного зв'язку, що можливо саме завдяки контролю знань учнів.

Десятиліттями використовувались та вдосконалювались такі традиційні форми контролю, як усне опитування та письмова перевірка знань, які зараз вже називають класичними. Той факт, що ці форми контролю знань проіснували такий довгий час і надалі продовжують використовуватись в навчальному процесі, безперечно говорить на їх користь. Все ж разом із вагомими перевагами ці форми мають і значні недоліки. Проведені дослідження переконливо вказують на недостатній рівень надійності, валідності і об'єктивності вимірювання та оцінювання знань при використанні класичних форм контролю.

Для прикладу розглянемо усне опитування. При такій формі контролю оцінювання проводилося за чотирибальною шкалою в інтервалі 2–5. З позиції умов, що забезпечують об'єктивність, при усній формі контролю знань не може бути забезпечена ні об'єктивність процесу вимірювання, ні об'єктивність інтерпретації результатів вимірювання. Одного й того ж опитуваного порізно оцінюють різні викладачі. Навіть тоді, коли опитування проводить група досвідчених педагогів і середні оцінки більшменш об'єктивні та підпорядковуються статистичним закономірностям, індивідуальні оцінки у різних викладачів можуть коливатись від максимальної до мінімальної, в той час як стандартна помилка цього методу вимірювання дорівнює ± 1 бал. Значні коливання між оцінками для однієї і тієї ж особи, знання якої контролюються, є наслідком недостатньої надійності методу оцінювання, що в свою чергу є результатом порушення критерію об'єктивності.

Крім вже названих суттєвих недоліків такої форми перевірки

знань, звертаємо також увагу на неможливість виконання критерію валідності, а саме, валідності змісту. В цій формі контролю знань інструментом вимірювання є запитання. Навіть, якщо використовуються запитання, що охоплюють весь зміст курсу, з якого проводиться опитування, то відповідь на обмежену кількість запитань не дає змоги проконтролювати знання за такими критеріями, як обсяг та повнота, системність та узагальнення.

Іншою класичною формою перевірки знань є письмова перевірка. Інструментом виміру такої перевірки є письмова робота. Мета її застосування – пошуки точніших методів вимірювання та оцінювання знань, ніж при усній формі контролю. Цей метод з точки зору критерію об'єктивності, а саме уніфікації вимірювання, де забезпечується більша об'єктивність процесу вимірювання та опрацювання даних, дійсно дає позитивні результати. Але процес оцінювання фактично залишився таким же, як і в усній формі контролю, а саме – далеким від об'єктивності при інтерпретації результатів. Тому метод письмової перевірки знань також не відповідає критеріям надійності і валідності. Більш того, використання цього методу як письмового оформлення фактично тих же завдань, які використовуються і при усній формі контролю, є не вдосконаленням методу вимірювання, а адміністративною боротьбою із зловживаннями, прояви яких є наслідком недосконалості методу вимірювання.

Однією із форм контролю, яку можна успішно використовувати при комп'ютерному контролі знань, є тестування. Нагадаємо, що тестування – це метод педагогічної діагностики, за допомогою якого вибір поведінки, яка репрезентує передумови та результати навчального процесу, повинен максимально відповідати принципам “співставлення”, об'єктивності, надійності і валідності вимірів.

Тест – це стандартизована система завдань, за результатами виконання яких можна в значній мірі об'єктивно оцінити рівень і якість засвоєння учнями навчального матеріалу, сформованість у них практичних навичок і вмінь.

Зазначимо, що слово “тест” в перекладі з англійської означає перевірку, іспит. Грамотно складений і правильно опрацьований тест має ряд переваг перед “класичною” контрольною роботою. Тестування дозволяє провести ширший і глибший контроль за-

своєння матеріалу на ту чи іншу тему. Це особливо важливо в тих випадках, коли необхідно виявити верхню межу знань в групі учнів, визначити серед них лідерів і відстаючих. Тестування виявляє загальну картину засвоєння матеріалу і дає можливість для індивідуальної роботи як із встигаючими, так і з відстаючими учнями.

Тестування може дуже допомогти учителеві. Перш за все педагогу легше виявити, якими конкретними елементарними діями учні ще не володіють. Крім того, тестування дозволяє охопити більший об'єм матеріалу, що дає педагогу ширші уявлення про знання учнів, що тестуються.

Дуже важливо, що, крім контролю, тестування реалізує і функцію навчання. Тому серед варіантів відповідей має бути правильна відповідь, щоб учень побачив її. В контрольних тестах це не обов'язково. В них можливий такий варіант відповіді: “правильної відповіді немає”.

Тестові завдання поділяються на задачі закритого типу та задачі відкритого типів. Задачі закритого типу – це тип тестових завдань, які передбачають різні варіанти відповідей на поставлене запитання чи завдання. Із списку запропонованих відповідей учень вибирає на його думку правильну.

Задачі закритого типу поділяються за видами:

- альтернативні завдання, які передбачають наявність двох варіантів відповідей: позитивну та негативну;
- завдання з множинним вибором. Такі тести передбачають вибір однієї, правильної відповіді з тих, що пропонуються;
- тести, побудовані за принципом відновлення відповідних частин. Це завдання на доповнення. Вони дістали широке використання у зошитах з друкованою основою;
- завдання з переструктуруванням даних. Це тести перехресного вибору або по співставленню. Вони розраховані на перевірку вмінь учнів зіставляти певні об'єкти, встановивши між ними відповідність. Різновид цих тестів – тести на ідентифікацію, коли замість словесних, числових або формульних відповідей пропонуються схеми, графіки, діаграми, малюнки тощо.

Задачі відкритого типу – це тип тестових завдань, що передбачають вільні відповіді учнів, тобто завдання без варіантів від-

повідей. Прикладом таких тестових завдань, є алгебраїчні чи геометричні задачі.

При тестуванні за допомогою комп'ютера використовують в основному тестові завдання закритого типу. Найбільшого поширення при комп'ютерному тестуванні набули альтернативні тести, тести з множинним вибором і тести перехресного вибору.

Як стверджує англійський психолог Клейн, в конструюванні тесту поєднані наука і мистецтво. Придумати завдання тесту – це мистецтво; перевірити створений тест за створеними правилами на надійність, валідність і стандартизувати його – наука.

Зазначимо, що тестові задачі повинні відповідати таким загальним вимогам:

- запитання мають бути легкими для розуміння; учень не повинен витратити час на те, щоб зрозуміти зміст запитання;
- тестові завдання повинні бути лаконічними, тобто, тексти запитання і відповіді повинні бути короткими, зберігаючи при цьому потрібний зміст;
- серед варіантів відповідей обов'язково має бути правильна, або варіант відповіді: “Правильної відповіді немає”;
- серед варіантів відповідей не повинно бути більше однієї правильної відповіді;
- серед варіантів відповідей не має бути абсолютно абсурдних відповідей.

Кількість тестових завдань визначає довжину тесту, яка може бути короткою (10–20 завдань), середньою (30–40 завдань) або довгою (50 і більше завдань). Найоптимальнішим є тест середньої довжини, що містить 30–40 завдань.

Про надійність тестів іноді роблять висновки за такою ознакою: якщо в усіх випадках перевірки тесту чи його варіантів виявиться, що учні в розподілі за показниками успішності займуть одні й ті ж місця, то тест можна вважати надійним. При повторних перевірках тесту за критерієм надійності натрапляють на певні труднощі: коли одному і тому ж учневі через деякий час запропонувати той самий тест, то немає жодної гарантії, що відповіді ґрунтуватимуться тільки на знаннях навчального матеріалу, а не на пригадуванні раніше даних відповідей. У зв'язку з цими труднощами виникає питання, чи не можна визначити коефіцієнт надійності на підставі лише однієї проби. Виявляється,

що це можна зробити. Весь тест ділять на дві частини, до однієї з них належать, наприклад, запитання з парними номерами, до другої – з непарними. Особливо доречно так робити, коли тест складається із завдань приблизно однакової складності. У протилежному випадку потрібно підібрати запитання паралельних варіантів у такий спосіб, щоб забезпечити однакову складність цих частин тесту. Якщо показники успішності обох частин співпадають, то тест можна вважати надійним. Опрацювання і аналіз результатів тестування – найважливіші моменти тестування. Якраз вони забезпечують якісний “зворотній зв’язок” навчання і показують напрям для індивідуальної роботи. Чим чіткіше визначені цілі і завдання тесту, тим легше скласти його і провести аналіз.

Результати виконання тестів, залежно від їх складності, оцінюють в балах. Наприклад, альтернативні тести оцінюються в один бал. Одним балом оцінюються і найпростіші тести з множинним вибором, тобто нескладні завдання і вправи. Складніші запитання можуть оцінюватись більшою кількістю балів. Сумарна кількість балів, набрана учнем за виконання всього тесту, складає його рейтинг. Для переведення балів у діючу дванадцятибальну систему оцінювання знань, використовують спеціальну шкалу.

З усього сказаного можна зробити висновок: переваги тестів полягають у можливості перевірити стан засвоєння значної частини навчального матеріалу, одночасно охопити опитуванням велику кількість учнів, забезпечити об’єктивність в оцінюванні знань. Тести економні в часі: учневі не доводиться витратити час на самостійне складання, формулювання відповідей і на поширений письмовий виклад їх, він зосереджує свою увагу на обдумуванні відповіді по суті. Тест не дає можливості відхилитися від висвітлення основних питань розділу чи теми, що цілком можливе у випадку вільної письмової чи навіть і усної відповіді. Перевірка тестових робіт не потребує значних зусиль і часу, односторонність правильних відповідей полегшує оцінювання. Учні через короткий час дізнаються про результати своїх відповідей, що запобігає запам’ятовуванню помилкових положень. Метод тестів дає змогу мати кількісні показники успішності, які можна математично опрацювати. Тестування дає досить точну картину засвоєння учнями навчального матеріалу.

Однак не потрібно ідеалізувати можливості тестів. Вони не в силі виявити всі особливості учнів. Звичайний тест не дає уявлення про метод розв'язку і не дозволяє вяснити, як розвинуті навички запису розв'язування задач, що є важливим моментом навчання. Тому тести не повинні повністю замінити «класичний» контроль. Лише розумна комбінація тестової з традиційними формами контролю дасть об'єктивний результат.

Авторським колективом кафедри прикладної математики та матаналізу Прикарпатського університету імені В. Стефаника розроблено тести з курсу інформатики відповідно до типових навчальних програм. Тестові завдання охопили такі теми: Операційна система MS-DOS, Операційна оболонка Norton Commander, Комп'ютерні віруси, Архівація файлів, Електронні таблиці SuperCalc, Системи управління базами даних FoxPro, Англійська мова для комп'ютера, Операційна оболонка Windows, Текстовий процесор Word, Електронні таблиці Excel, бази даних Access. З різних тем є різна кількість тестових завдань (від 1 до 3).

Тести занесені в комп'ютерну тестову оболонку. Тестова оболонка дозволяє вводити нові тести, редагувати вже існуючі, забезпечує тестування та перегляд результатів тестування. При тестуванні можна встановлювати параметри тестування, зокрема: кількість запитань, час на роздуми тощо. Програма вибирає запитання та виводить варіанти відповідей в довільному порядку. При введенні тесту до кожного запитання можна додавати текст допомоги, задавати складність запитання, визначати кількість варіантів відповідей (від 2 до 9). Окремі тести з однієї і тої ж теми можна об'єднувати в каталоги.

Оцінювання в кінці тестування дається в чотирибальній системі: "2", "3", "4", "5". Розроблена система тестів розрахована на середній критерій оцінювання, тобто при 12-бальній системі оцінювання, використовуємо таку шкалу: оцінка "2" – 2 бали, "3" – 5 балів, "4" – 7 балів, "5" – 10 балів. Для оцінювання за вищим критерієм рекомендується крім тестів давати учням ще й практичне завдання, тобто задачу відкритого типу.

Проводити тестування доцільно перед або замість традиційної контрольної чи самостійної роботи. Оскільки тестування займає небагато часу, то його можна проводити декілька раз, слід-

куючи за динамікою засвоєння знань учнями.

На завершення зазначимо, що розроблені тестові завдання дають змогу перевірити вміння і навички, отримані учнями на заняттях з інформатики. Їх можуть використати для самоконтролю студенти вищих навчальних закладів, слухачі курсів та всі, хто займається самоосвітою з інформатики.

Література:

1. Федоров Е. Контрольный тест-анализ. / Математика в школе. – 1991. – №3. – С. 27.
2. Халамайзер А. Тест для первокурсников Брауншвейского университета (ФРГ). / Математика в школе. – 1990. – №5. – С. 66–68.
3. Жолковський К. Тепер – тести. / Демократична Україна. – 20 березня 1994.
4. Човнюк Ю., Діктерук М. Методи тестування знань, здібностей і обдарованості студентів вузів: концепція біотехнічної системи та інформаційний аналіз. / Проблеми освіти: Науково-метод. зб. Вип. 12. – К., 1998.
5. Орлова Т.М. Тесты для всех. – К.: Фірма “Довіра”, 1993.
6. Мауковский М. Тесты, тесты, тесты... – М.: Молодая гвардия, 1990.
7. Розенберг Н.М. Тестова перевірка знань учнів. – К.: Радянська школа, 1973.
8. Глинський Я.М. Практикум з інформатики: Навчальний посібник. 2-е видання. – Львів: Підприємство Деол, 1999.
9. Фигурнов В.Э. IBM PC для пользователя. 6-е издание. – М.: Инфра-М, 1996.

ПРОБЛЕМИ НАВЧАННЯ ІНФОРМАТИКИ В ШКОЛІ

М.С. Головань

м. Суми, Українська академія банківської справи

На сьогоднішній день інформатика виділилася у науку, що вивчає фундаментальні властивості, структуру, функції автоматизованих інформаційних систем, а також їх проектування, створення, оцінки, використання і впливу на різноманітні галузі соціальної практики. Об'єктом вивчення інформатики є інформаційні процеси та інформаційні системи. Інформатику трактують в широкому розумінні як галузь людської діяльності, пов'язану з процесами опрацювання інформації за допомогою комп'ютерів і телекомунікаційних засобів зв'язку із середовищем її використання. У вузькому розумінні інформатику трактують також як науку про комп'ютерну техніку, яка складається з трьох взаємопов'язаних частин – технічних, програмних та алгоритмічних засобів.

У свою чергу, інформатику в цілому розглядають як фундаментальну науку, як прикладну дисципліну і як галузь народного господарства.

З'явилася відмінність між інформатикою як наукою з власною предметною областю та інформаційними технологіями.

В останні роки шкільний курс “Основи інформатики й обчислювальної техніки” вийшов на якісно новий етап свого розвитку. Змінився погляд на те, що розуміти під комп'ютерною грамотністю. Якщо на початку становлення інформатики як навчальної дисципліни основна увага була спрямована на розвиток алгоритмічного мислення, вивчення мов програмування (“програмування – друга грамотність”), то зараз уже практично усіма усвідомлено, що шкільна інформатика не повинна бути курсом програмування. На сьогодні мінімальним рівнем комп'ютерної грамотності є оволодіння засобами комп'ютерних інформаційних технологій і основним завданням курсу інформатики є формування інформаційної культури учнів.

Проте, помилково було б орієнтувати курс основ інформатики й обчислювальної техніки тільки на практичне освоєння засобів інформаційних технологій. Тоді інформатика втратила б зна-

чення як самостійної навчальної дисципліни.

Вивчення основи інформатики в школі повинно переслідувати дві мети: загальноосвітню і прагматичну. Загальноосвітня мета полягає в освоєнні фундаментальних понять сучасної інформатики. Прагматична – в одержанні практичних навичок роботи з апаратними і програмними засобами сучасних комп'ютерів. Курс шкільної інформатики змістовно і методично повинен бути побудований так, щоб обидва завдання – загальноосвітнє і прагматичне – вирішувалися паралельно.

Інформатика як освітня дисципліна швидко розвивається і її зміст постійно змінюється.

На наш погляд, в останні роки в розвитку інформатики як навчальної дисципліни спостерігається криза, викликана тим, що:

- завдання першого етапу введення шкільного предмету інформатики в основному виконано;
- усі учні знайомляться з основними комп'ютерними поняттями й елементами програмування, і доки вирішувалося це завдання, передній край наукової і практичної інформатики пішов вперед, і стало неясно, у якому напрямку рухатися далі;
- вичерпані можливості вчителів інформатики, які були, як правило, вчителями математики або фізики, або ж у минулому професійними програмістами, які пройшли лише короткострокову підготовку в інституті удосконалення вчителів;
- відсутні зважені, реалістичні підручники;
- школи оснащені як сучасними персональними комп'ютерами, так і персональними комп'ютерами першого покоління вітчизняного виробництва, або й зовсім не мають ніякої комп'ютерної техніки;
- внаслідок відмінності умов для викладання інформатики в різних школах (різноманітності типів комп'ютерної техніки) і відносної свободи шкіл у виборі профілів класів, навчальних планів і освітніх програм, з'явилися значні відмінності у змісті навчання інформатики.

У вищих навчальних закладах підготовка з інформатики має орієнтацію на формування навичок використання комп'ютера як інструмента діяльності, мало враховує ту підготовку школярів з

інформатики, яка здійснюється в середніх школах.

Істотно змінилася і парадигма досліджень у галузі інформаційних технологій і їхнього застосування на практиці. У початковий період свого існування шкільна інформатика черпала ідеї в основному з практики використання інформаційних технологій у наукових дослідженнях, технічній кібернетиці, АСУ і САПР. У зв'язку з кризою фінансування наукових установ і досліджень, фактичною зупинкою наукомістких виробництв і їхнього перепрофілювання, загальна наукова орієнтація курсу інформатики втратила актуальність. Значно знизилася початкова мотивація школярів щодо вивчення науково-орієнтованих предметів і знизилася успішність з цих предметів. Сьогодні явно виявляється соціальний запит, спрямований на бізнес-орієнтовані застосування інформаційних технологій, користувачькі навички використання персональних комп'ютерів для підготовки і друкування документів, бухгалтерських розрахунків, роботу з базами даних, використання комунікаційних можливостей комп'ютерних мереж і т.д. Щодо програмування, то зараз розрізняють процедурне і декларативне програмування, об'єктно-орієнтоване програмування, візуальне програмування. З'явилися нові мови програмування, змінилася сама технологія програмування, тому повинні бути і нові підходи до навчання учнів елементам програмування.

Проте, більшість загальноосвітніх навчальних закладів не готові до реалізації цього запиту в наслідок відсутності відповідної навчальної комп'ютерної техніки і недостатньої підготовки вчителів інформатики.

Згідно сучасних дидактичних уявлень про структуру загальної освіти, в середній школі вивчаються не “основи наук”, а певні області дійсності. Тому при побудові навчальних планів виходять саме з принципу вивчення в навчальних предметах саме областей дійсності (хоча їх вивчення здійснюється через вивчення основ наук, де систематизовані знання про ці області), а не безпосередньо відповідних їм галузей науки.

Стосовно інформатики, чия область включає в себе інформаційні процеси і інформаційну діяльність людини, то в навчальному предметі мають бути відображені основи наукових напрямків, об'єднаних спільним предметом дослідження – інформаційними процесами та інформаційними системами.

Структуру предметної галузі сучасної інформатики складають наступні розділи:

Теоретична інформатика;

Засоби інформатизації:

- технічні (опрацювання, подання, передавання даних);
- програмні (системні програмні засоби, програмні засоби реалізації універсальних та професійно орієнтованих технологій);

Інформаційні технології;

Соціальна інформатика.

Зміст виділених розділів може бути наступним.

Теоретична інформатика. Інформація як семантична властивість матерії. Інформація і еволюція в живій і неживій природі. Основи теорії інформації. Методи її вимірювання, принципи кодування інформації.

Математичні та інформаційні моделі. Теорія алгоритмів. Обчислювальний експеримент як методологія сучасного наукового пізнання.

Інформація і знання. Семантичні аспекти інтелектуалізації процесів та інформаційних систем. Інформаційні системи штучного інтелекту. Методи подання знань. Теорія і методи розробки та проектування інформаційних систем і технологій.

Засоби інформатизації.

Технічні засоби опрацювання, подання і передавання даних. Персональні комп'ютери. Робочі станції. Пристрої введення/виведення та відображення інформації. Аудио- відеосистеми, системи мультимедіа. Мережі ЕОМ. Засоби зв'язку та комп'ютерні телекомунікаційні системи.

Системні програмні засоби інформатизації. Операційні системи і середовища. Системи і мови програмування. Сервісні оболонки. Системи теледоступу, обчислювальні та інформаційні середовища.

Засоби реалізації універсальних технологій. Текстові процесори. Електронні таблиці. Системи управління базами даних. Засоби моделювання об'єктів, процесів та систем. Інформаційні мови і формати подання даних і знань; словники; класифікатори; тезауруси. Засоби захисту інформації від руйнування і несанкціонованого доступу.

Засоби професійно орієнтованих технологій. Видавничі системи. Системи реалізації технологій автоматизації розрахунків, проектування, опрацювання даних (обліку, планування, управління, аналізу, статистики і т.д.). Системи штучного інтелекту (бази знань, експертні системи, діагностичні, навчаючі та ін.).

Інформаційні технології.

Ведення/виведення, збирання, зберігання, передавання та опрацювання інформації.

Підготовки текстових і графічних документів, технічної документації.

Інтеграції та колективного використання різномірних інформаційних ресурсів.

Захисту інформації.

Програмування, проектування, моделювання, навчання, діагностики, управління (об'єктами, процесами, системами).

Соціальна інформатика. Інформаційні ресурси як фактор соціально-економічного і культурного розвитку суспільства.

Інформаційне суспільство – закономірності і проблеми становлення і розвитку. Інформаційна інфраструктура суспільства. Проблеми інформаційної безпеки. Нові можливості розвитку особистості в інформаційному суспільстві. Інформаційна культура і інформаційна безпека особистості.

Інформатика як наука перебуває у стадії розвитку, погляди на її предмет і надалі буде змінюватися, тому пошук шляхів удосконалення фундаментальної підготовки учнів з інформатики триватиме і далі, але на наш погляд, необхідно прагнути виділити у змісті навчального курсу її інваріантну складову, зміст якої б залишався тривалий час незмінним, незважаючи на швидкий розвиток комп'ютерної техніки, програмного забезпечення, появи нових інформаційних технологій. Ця інваріантна частина повинна бути теоретичною основою курсу, а її технологічна (змінна) складова буде основою практичних знань, умінь і навичок учнів з курсу інформатики.

ВИМОГИ ДО КОРИСТУВАЧІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПЕРСОНАЛЬНИХ КОМП'ЮТЕРІВ В УМОВАХ РОЗВИТКУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ІНФОРМАЦІЙНОГО СУСПІЛЬСТВА

Д.В. Гондюл

м. Київ, Міжнародний інститут лінгвістики і права

Тенденції автоматизації праці та розвитку мережі Інтернет призвели до зростання кількості автоматизованих робочих місць, а звідси і користувачів персональних комп'ютерів (ПК) у світі та в Україні, що, в свою чергу, призвело до зростання вимогливості щодо комп'ютерної грамотності користувачів ПК.

Обсяги і терміни виконання конкретних задач з використанням ПК випереджають темпи навчання користувачів професійній роботі з ПК, що суттєво збільшує кількість користувачів ПК з несистематизованими, неповними або вузько спрямованими знаннями інформаційних технологій, зокрема комп'ютерної техніки та програмного забезпечення. Така тенденція уповільнює процес перерозподілу робочої сили між галузями діяльності та зменшує якість професійної відповідності посаді, що займається.

Стрімкий розвиток апаратного та програмного забезпечення вимагає постійного поновлення комп'ютерної грамотності та періодичних витрат на модернізацію персональних комп'ютерів та периферійної техніки.

Переважає більшість видів діяльності, де вже широко застосовується комп'ютерна техніка, потребує знань певного чітко визначеного спектру програмного забезпечення, що дозволило визначити його як базове програмне забезпечення.

Насамперед це операційна система зі стандартними додатками, що входять до її складу, та інтегрований офісний пакет, який включає текстовий процесор, електронні таблиці, систему управління базами даних, систему створення презентацій та графічний редактор. Серед стандартних додатків операційної системи окреме місце посідають програми для управління файловою структурою та програми для роботи в мережі Інтернет.

Професійні знання базового програмного забезпечення дають можливість ефективного усвідомленого вивчення інших на-

прямків розвитку програмного забезпечення та забезпечують систематизований підхід до його освоєння.

Таким чином, головна вимога до користувача персонального комп'ютера набути якісних професійних знань базового програмного забезпечення. Для такої гнучкої динамічної системи як розвиток програмного забезпечення дуже важливо дотримуватись методик навчання комп'ютерній грамотності, щоб мати системне уявлення про еволюцію програмного забезпечення, його класифікацію та напрямки подальшого розвитку.

Поняття “комп'ютерна грамотність” (PC literacy) та “комп'ютерно-грамотний користувач” (PC-literate user) є досить гнучкими і не мають чітко визначених світових або національних стандартів, оскільки кожна особа або організація, яка вимагає “комп'ютерної грамотності”, вкладає в це поняття суб'єктивні вимоги, притаманні саме для своєї діяльності, мета яких максимально ефективно виконати з допомогою персонального комп'ютера поставлені завдання.

Єдиного вирішення проблеми стандартизації вимог до користувачів програмного забезпечення ПК немає, але науковці з усіх країн об'єднують свої зусилля для створення найбільш прийняттого комплексу вимог, і результатом їх зусиль є створення організацій, сертифікати або ліцензії яких про набуття користувачем комп'ютерної грамотності визнаються багатьма країнами світу та міжнародними компаніями чи організаціями.

Серед таких організацій найпоширенішими є організації двох типів:

1. Державні (національні).
2. Міжнародні.

Мета національних організацій полягає в тому, щоб забезпечити мінімальний рівень володіння персональним комп'ютером, достатній для виконання простих загальних завдань. Ліцензія або сертифікат таких організацій дозволяє соціально захистити людину від безробіття в межах держави, а отримані знання дають підстави для поглибленого вивчення базового програмного забезпечення та вивчення спеціалізованого апаратного та програмного забезпечення.

Головна мета міжнародних організацій, які, як правило, є недержавними, – уніфікувати вимоги до комп'ютерної грамот-

ності шляхом врахування інтересів і розбіжностей серед країн, в яких є представництва цих організацій. Між такими організаціями, як правило, існує більш високий рівень конкуренції та більша відповідальність, тому якість знань користувачів з сертифікатами міжнародних організацій не потребує додаткової перевірки і визнається більшістю роботодавців.

Яскравим прикладом такої організації є Європейський центр комп'ютерного ліцензування. Ліцензія цієї організації більш відома, ніж сама організація, і має назву Європейська ліцензія володіння комп'ютером (European Computer Driving Licence – ECDL). Ця ліцензія визнається практично всіма країнами Європи, а сама організація, головний офіс якої розташований в Ірландії, має представництва в 26 країнах світу.

Вивчення програмного забезпечення персонального комп'ютера можна порівняти із вивченням іноземної мови. Навчання складається з декілька етапів, і з кожним етапом збільшується складність, поглибленість та специфічність матеріалу. Як в іноземній мові спочатку вивчається абетка та звуки, так само і в вивченні програмного забезпечення спочатку формується загальне уявлення користувача про принципи роботи та складові персонального комп'ютера, вивчається термінологія, що є першою вимогою до комп'ютерної грамотності користувача ПК.

Наступним етапом в вивченні іноземної мови є формування навичок побудови речень. Аналогом у вивченні програмного забезпечення є організація інформації на дисках (пристроях збереження інформації) та файлова структура. Тому знання організації інформації в комп'ютері, файлової структури, класифікації файлів та вміння побудувати і модифікувати власну структуру файлів і каталогів, виконувати основні операції над ними є наступною вимогою до знань та вмінь користувача.

Як в мові поділяють члени речення на основні та другорядні, так і в програмному забезпеченні існує чіткий поділ програм на операційні системи та додатки. А як головні члени речення поділяють на підмет і присудок, так і операційні системи розрізняють як командні (DOS) та віконні (Windows), кожна з яких має власні ознаки. Таким чином, вивчення основних властивостей та можливостей операційної системи є теж не-

обхідною вимогою до комп'ютерної грамотності користувача ПК.

Роль та різноманітність другорядних членів речення по відношенню до речення в цілому дуже схожа з роллю та різноманітністю додатків по відношенню до програмного забезпечення в цілому. Саме додатки виконують основну та специфічну роботу, яка і дає практичний кінцевий результат, тому їхню роль не можна недооцінювати. А виділення всього різноманіття додатків в окремі групи дозволяє сформуванати вимоги до знань та умінь користувача для кожної групи окремо.

З досвіду Європейського центру комп'ютерного ліцензування доцільно виділити 7 основних груп вимог до знань та умінь користувачів програмного забезпечення ПК:

1. Загальна концепція інформаційних технологій (ІТ);

Користувач повинен знати:

- складові персонального комп'ютера (основні та периферійні);

- принципи роботи операційної системи;

- функціональні можливості комп'ютерних програм;

- принципи організації і збереження інформації в ПК;

- основні характеристики файлів;

- як створюється програмне забезпечення;

- принципи роботи та можливості графічного користувацького інтерфейсу;

- принципи роботи мультимедіа.

Користувач повинен вміти:

- визначати та розуміти важливість і доцільність використання комп'ютерів та можливості їх використання;

- визначати можливості використання комп'ютерних систем у бізнесі;

- визначати можливості використання комп'ютерних систем у промисловості;

- визначати можливості використання комп'ютерних систем в освіті;

- визначати можливості домашнього використання комп'ютерних систем у приватних інтересах;

- визначати можливості впровадження комп'ютерних систем у повсякденному житті;

- розрізняти операційні системи та додатки.

2. Використання персонального комп'ютера та управління файлами;

Користувач повинен знати:

- роль та послідовність етапів використання персонального комп'ютера;

- можливості використання операційної системи для коректного виконання будь-яких операцій.

Користувач повинен вміти:

- коректно завантажити персональний комп'ютер;
- створювати каталоги та підкаталоги;
- розуміти структуру дерева каталогів;
- копіювати та переносити файли;
- знищувати файли з одного або кількох каталогів;
- створювати файл за допомогою текстового редактору та зберігати його у каталозі;
- перейменовувати файли;
- визначати властивості каталогів (кількість файлів у них, розмір файлів у них, дату створення та поновлення файлів);
- форматовувати дискети (гнучкі магнітні диски);
- організувати резервні копії даних (копіювати файли з гнучких магнітних дисків на інші гнучкі та жорсткі магнітні диски або в інші каталоги на тому ж диску);
- зберігати файли на дискетах;
- вибрати із переліку встановлених драйвер потрібного принтеру;
- роздруковувати на встановленому принтері;
- коректно вимикати комп'ютер.

3. Текстовий процесор

Користувач повинен знати:

- як редагувати та створити новий документ;
- можливості оформлення документу.

Користувач повинен вміти:

- завантажувати текстовий процесор;
- відкривати існуючі документи (вводити, редагувати, вставляти та знищувати текст);
- створювати нові документи (вводити, редагувати та вставляти текст);

- зберігати документ на гнучкому та жорсткому магнітних дисках;

- використовувати стандартні функції текстового процесора (переміщати текст в документі, копіювати текст в інший документ, шукати та замінювати слова на інші);

- змінювати вигляд тексту (використовувати курсив, напівжирний текст, підкреслення, змінювати шрифт, горизонтальне вирівнювання та відступи першого рядка та всього абзацу, інтервал між рядками та абзацами);

- використовувати програми перевірки орфографії та вносити зміни;

- змінювати параметри сторінок документу;

- друкувати документи та частини документу;

- створювати колонтитули та зноски;

- задавати та змінювати нумерацію сторінок документу;

- використовувати довідкові функції текстового процесора;

- змінювати регістр та індекс тексту;

- додавати в поштові повідомлення текст документу та документ;

- імпортувати таблиці, малюнки, діаграми та інші об'єкти;

- створювати таблиці в документі;

- встановлювати табулятори;

- оформляти ділові документи і використовувати їх в роботі (організаціях);

- використовувати стилі, розділи та посилання;

- використовувати шаблони документів;

- використовувати програмне забезпечення, що інтегрує текстовий процесор з електронними таблицями, базами даних та графічними програмами.

4. Електронні таблиці.

Користувач повинен знати:

- властивості електронних таблиць;

- основні функції програмного забезпечення електронних таблиць;

- можливості створення електронних таблиць, здійснення розрахунків, побудови діаграм.

Користувач повинен вміти:

- завантажувати програмне забезпечення електронних таб-

лиць;

- відкривати файли електронних таблиць, вносити зміни, додавати клітини, розраховувати нові дані;

- вставляти стовпці та рядки, додавати діапазони стовпців та рядків у певне місце;

- створювати нову таблицю та вносити дані (числа, текст, формули);

- виділяти суміжні та окремі діапазони клітин;

- змінювати формат клітини та діапазонів клітин;

- змінювати ширину стовпців та висоту рядків;

- сортувати дані в таблиці;

- використовувати основні математичні та логічні функції електронних таблиць (знаходження суми, добутку тощо);

- друкувати та зберігати електронні таблиці;

- змінювати параметри сторінок таблиці;

- використовувати довідкову систему електронних таблиць;

- використовувати абсолютні та відносні посилання в формулах;

- використовувати різні типи діаграм в залежності від змісту даних;

- імпортувати діаграми з дискети чи диску;

- друкувати діаграму з назвою та легендою;

- додавати, знищувати та змінювати порядок листів електронної таблиці;

- змінювати формат клітин у групі листів;

- створювати формули зі посиланнями на дані на інших листах електронної таблиці в інших електронних таблицях;

- переносити дані з однієї таблиці в іншу;

- використовувати програмне забезпечення, що інтегрує електронні таблиці з текстовим процесором, базами даних та графічними програмами.

5. Бази даних.

Користувач повинен знати:

- принципи створення простої бази даних, використовуючи стандартну систему управління базами даних;

- основні можливості баз даних.

Користувач повинен вміти:

- створювати нову базу даних:

- побудувати форму простої бази даних;
- створювати структуру записів;
- вводити дані в базу даних;
- редагувати дані;
- додавати записи;
- знищувати записи;
- визначати ключові поля;
- зберігати базу даних на диску;
- шукати, вибирати та сортувати дані за певними критеріями;
- виводити вибірку даних в певній послідовності на екрані та в звіті;
- використовувати довідкові функції;
- використовувати існуючу базу даних;
- завантажувати та відкривати існуючу базу даних;
- вводити дані в нову базу даних;
- редагувати дані;
- додавати записи;
- шукати, вибирати та сортувати дані за певними критеріями;
- виводити вибірку даних в певній послідовності на екрані та в звіті;
- змінювати структуру бази даних;
- використовувати довідкові функції.

6. Презентації.

Користувач повинен знати:

- загальні можливості програмного забезпечення створення презентацій;
- практичний підхід до створення різноманітних презентацій;

Користувач повинен вміти:

- побудувати та оформити кожний коментар;
- вставляти малюнки з бібліотеки малюнків;
- малювати прості елементи (прямокутники, коло, текст, лінії);
- додавати та змінювати колір, тінь та границі об'єктів презентації;
- використовувати різні шрифти та змінювати шрифти;

- центрувати та виділяти жирним текст;
- копіювати та змінювати розмір графічних об'єктів;
- використовувати організаційні діаграми;
- використовувати та налаштовувати показ слайдів;
- використовувати довідкові функції.

7. Інформаційні служби мережі.

Користувач повинен знати:

- основні принципи роботи в мережі;
- як користуватись електронною поштою;
- як використовувати загальнодоступні інформаційні служ-

би мережі;

- основні можливості ресурсів Інтернет.

Користувач повинен вміти:

- використовувати електронну пошту:
 - o створювати повідомлення;
 - o відправляти повідомлення;
 - o створювати прикріплення до повідомлень;
 - o зберігати повідомлення у файл;
 - o друкувати повідомлення;
 - o копіювати повідомлення;
 - o відповідати на повідомлення;
 - o переадресувати повідомлення;
 - o переміщати повідомлення в певний каталог;
 - o знищувати повідомлення.
- використовувати ресурси мережі:
 - o підключатись до мережі Інтернет / інформаційної служби / корпоративної мережі;
 - o знайти та вилучити потрібну інформацію;
 - o підтвердити отримання інформації (зберегти у файлі, показати на екрані, роздрукувати або переслати електронною поштою).

Таким чином, систематизовані знання сучасних інформаційних технологій, складових і принципів роботи персонального комп'ютера, можливостей та ресурсів мережі Інтернет, впевнене володіння операційною системою та інтегрованим офісним пакетом є необхідною і достатньою вимогою до користувачів персонального комп'ютера, здатних його використовувати як для професійної діяльності, так і для власних потреб.

ВИКОРИСТАННЯ ЖИТТЄВОГО ДОСВІДУ УЧНІВ ЯК ЕЛЕМЕНТ ОСОБИСТІСНО-ЗОРІЄНТОВАНОГО НАВЧАННЯ НА УРОКАХ ІНФОРМАТИКИ

Г.А. Горшкова

м. Кривий Ріг, Криворізький обласний ліцей-інтернат для сільської молоді

В сучасних умовах активізується розвиток педагогіки в різних напрямках: гуманістична, соціальна, діагностична, корекційна, експериментальна, педагогіка ненасильства та інші.

Предметом гуманістичної педагогіки є виховання ЛЮДИНИ, одним із провідних напрямів гуманізації є впровадження особистісно зорієнтованого навчання, мета якого не тільки навчати і виховувати учня, а й розвивати в ньому особистість.

Особистісно-зорієнтоване навчання дає можливість кожному учню реалізувати себе в пізнанні, в навчальній діяльності з опорою на інтереси та нахили, можливості та здібності, ціннісні орієнтації та суб'єктний досвід.

Особистісно-зорієнтоване навчання ґрунтується на особистісному підході. Саме він визначає місце дитини у виховному процесі, робить учня активним суб'єктом виховання, а отже, встановлює суб'єкт-суб'єктні взаємовідносини.

Суб'єкт-суб'єктні взаємовідносини дозволяють створювати на навчальних заняттях атмосферу партнерства і співробітництва, передбачають позитивне сприймання і гуманне ставлення до протилежної думки, погляду, позиції, пошук спільної узгодженої основи.

Змістовні, цікаві уроки, насичені самостійними творчими завданнями, ще не є уроками особистісно-зорієнтованими. Окрім створення вчителем доброзичливої творчої атмосфери, необхідно постійне звертання до суб'єктного досвіду школярів, тобто, до досвіду їх власної життєдіяльності.

Звернемося до шкільного курсу інформатики. Світ інформатики – це світ комп'ютерів та алгоритмів. Зупинимося на одній з найважливіших частин курсу – алгоритмізації.

На етапі формування основних понять в учнів виникають певні труднощі, які ведуть до зменшення зацікавленості предме-

том. Так, поняття алгоритму є фундаментальним поняттям алгоритмізації, а отже і програмування. Від правильності засвоєння залежить подальше вивчення всього курсу. Підводячи учнів до поняття алгоритму, вчитель розглядає ряд життєвих прикладів (режим дня, рецепт торта та інші), де обов'язковим є дотримання певної послідовності, а також пропонує неправильний порядок дій, що приводить до цікавих, але невірних результатів.

Приклад. Деякий зловмисник видав таку послідовність дій за алгоритм отримання кип'ятку:

- 1) налити воду у чайник;
- 2) відкрити кран газового пальника;
- 3) поставити чайник на плиту;
- 4) чекати, доки вода закипить;
- 5) піднести сірник до пальника;
- 6) запалити сірника;
- 7) виключити газ.

Виправте помилки в цьому порядку дій для того, щоб не стався нещасний випадок.

Учні знаходять помилки, виправляють їх, після чого їм дається означення алгоритму як точної системи вказівок для виконання послідовності дій, що направлені на досягнення поставленої мети чи на розв'язування задачі.

Завданням вчителя на особистісно-зорієнтованому уроці є не лише виявлення суб'єктного досвіду учнів, а й переведення його у соціально значущий зміст («окультурення»), тим самим сприяння особистісному засвоєнню цього змісту.

При вивченні теми «Цикли» виникає проблема розуміння учнями відмінностей в роботі різноманітних видів циклів при їх побудові. Вчитель використовує такі приклади.

Ви прийшли у гості:

1. Господиня пропонує з'їсти одне тістечко. Якщо ви не зголодніли – відмовитесь, якщо ви зголодніли – ви з'їсте, не вгамувавши голоду, візьмете ще одне;

Цикл ПОКИ:

поки голодний

пц

візьми одне тістечко;
з'їж тістечко;

кц

2. Друга господиня відразу пропонує вам з'їсти тістечко, не запитавши-чи є у вас бажання їсти, чи подобається вам таке тістечко.

Цикл ДО:

Повторювати

пц

взяти тістечка;
з'їсти тістечко;

кц

До вгамував голод;

3. Третя господиня настійно змушує вас з'їсти 25 тістечок

Для тістечко від 1 до 25

пц

взяти тістечка;
з'їсти тістечко;

кц

Питання, на яке необхідно відповісти учням: яка господиня «найдемократичніша» (найгостинніша)?

Одночасно зі словесним наведенням прикладів, вчитель малює на дошці блок-схеми і допомагає учням знайти відмінності у роботі циклів, робить запис алгоритмічною мовою.

Такі методи та прийоми можна використовувати на різних етапах уроку. Вони допомагають вчителю формувати «колективне» знання як результат «окультурення» індивідуальних знань учнів, а не просто добиватися від класу відтворення готових зразків, підготовлених ним для засвоєння.

КОНСТРУИРОВАНИЕ ВЕРБАЛЬНЫХ ТРЕНАЖЕРОВ

В.Н. Евтеев, О.М. Ткаченко, Г.Е. Черный
г. Кривой Рог, Криворожский государственный педагогический
университет

В настоящее время образование в нашей стране пребывает в глубочайшем кризисе, граничащим с катастрофой. Основная причина столь печального состояния системы обучения состоит в невостребованности обществом квалифицированных кадров. Социальная ситуация такова, что оплата труда очень слабо связана со сложностью и качеством выполнения работ. Указанная причина является внешней по отношению к самой системе. Но существует и внутренняя причина, усугубляющая и без того напряженную ситуацию. Речь идет о жестком директивном управлении образованием в сочетании с нелепым набором нормативных показателей, принятых для характеристики состояния системы. Так, например, качество знаний проверяют по диаграмме распределения оценок, нисколько не заботясь об объективности самих оценок.

Наиболее важной характеристикой любой системы является устойчивость ее функционирования. Потеря устойчивости и переход к режиму неконтролируемых возрастающих самовозбуждающихся колебаний, как правило, приводит к разрушению. К катастрофической потере устойчивости может привести оптимизация и интенсификация. Однако устойчивость не теряется, если ввести обратную связь: жесткий план заменить величиной, пропорциональной фактически имеющимся ресурсам [1]. Попытка постепенного перевода системы из текущего в более благоприятное состояние сопровождается рядом ухудшающих обстоятельств. Мы не будем останавливаться на их описании, укажем лишь, что слабо развитая система может перейти в лучшее состояние почти без предварительного ухудшения, в то время как развитая система, в силу своей устойчивости, на такое постепенное, непрерывное улучшение неспособна [1]. Поэтому слаборазвитая система народного образования в первые годы советской власти управлялась жесткими директивами достаточно успешно. Но продолжение управления такими методами уже достаточно

развитой, многоуровневой и стабилизовавшейся образовательной системы привело к нарастанию нестабильности.

Наиболее важной задачей системы образования в настоящее время, на наш взгляд, является не столько перестройка, сколько стабилизация качества подготовки специалистов на уровне, сравнимом с началом семидесятых годов. Без этого шага всяческие инновации приведут лишь к разрушению системы. Поскольку система образования имеет ступенчатую иерархическую структуру, особо важное значение приобретают критерии отбора в элитную группу верхней ступени из нижней. Из математической теории элит известно, что, во-первых, критерий отбора должен быть внешним по отношению к элите, то есть, независимым от ее текущего состояния, иначе элита обречена на вырождение, и, во-вторых, понижение среднего уровня генеральной совокупности неизбежно приведет к понижению уровня элиты. Падение уровня образования находит свое отражение в ослаблении требований к выпускникам средних школ. Так, нормативное количество слов в изложении упало за 12 лет на 20%: с 450-550 до 350-450 [2, 3].

Остановка падения среднего уровня подготовки специалистов возможна лишь при наличии обратной связи в процессе обучения, а именно, объективной информации о степени обученности. Говоря другими словами, необходимо употреблять только те критерии оценки, которые удовлетворяют следующим требованиям:

Независимость критерия от процесса обучения и протекающего параллельно воспитания, то есть критерий оценивания обученности не должен зависеть от особенностей применяемых методов обучения, общественной деятельности учащегося, не должен содержать поощрительных или осуждающих аспектов. Критерий должен характеризовать только результат обучения, и никоим образом не зависеть от процесса.

Неизменность критерия в процессе использования. Нельзя менять классификационное правило во время классификации. Критерий не должен учитывать региональных, национальных, религиозных и прочих особенностей, способных исказить проверку.

Формальность критерия необходима для исключения влия-

ния субъективного человеческого фактора. Преподаватель, подготовивший ученика, или любой другой человек, заинтересованный в результатах отбора, не должен участвовать в проведении классификационного тестирования.

Таким требованиям должны удовлетворять лишь критерии, используемые при отборочной проверке перехода на следующий уровень иерархии образования. В самом же процессе обучения преподаватель волен использовать любые критерии, которые он считает эффективными. В Российской Федерации предпринимается попытка введения единого государственного экзамена (ЕГЭ) для выпускников средних школ [4]. При этом игнорируется третье требование к критериям оценки, а именно, ЕГЭ может только увеличить оценку. Такая ошибка приведет к провалу эксперимента и, в конечном счете, к дискредитации самой идеи единого госэкзамена.

Мы считаем, что введение жесткого независимого тестирования при переходе на следующий уровень образования продиктовано суровой необходимостью нынешней ситуации в образовании Украины. Возражений, связанных с тем, что, мол, вся система образования превратится в подготовку к тестам, не стоит опасаться. Во-первых, потому, что все равно она таковой уже является. Достаточно упомянуть олимпиадное движение, подготовку к контрольным мероприятиям отделов образования и т.п. Во-вторых, потому, что использование многоуровневых по сложности тестов позволит охватить все многообразие требований к результатам обучения.

При урочно-классной организации обучения преподаватель не имеет физиологической возможности полностью контролировать процесс тренировки каждого ученика по выработке нормативных навыков. Ликвидировать это узкое место в процессе обучения призваны компьютерные тренажеры.

Мы рассмотрим тренажеры, предназначенные для выработки навыков вербально логического оперирования понятиями. Кроме общих требований к дидактическому материалу, таким как тщательность отбора, структурированность информации, безукоризненная стилистика и т.д. Необходимо также учитывать специфические требования к компьютерным тренажерам: формализацию иерархического дерева вопросов (заданий); порцион-

ность подачи материала; по возможности, исчерпывающее объяснение ошибок, допускаемых учащимся; а также психологические и физиологические требования к работе с клавиатурой и дисплеем.

Мы реализовали достаточно простую схему взаимодействия учащегося с тренажером (рис. 1).

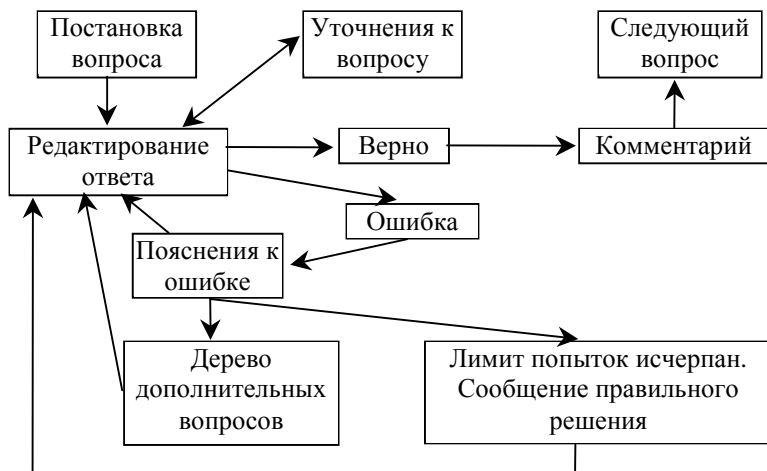


Рис. 1. Схема сценария взаимодействия ученика с тренажером. Стрелками указаны возможные переходы.

Прохождение дополнительных вопросов осуществляется тоже по этой схеме, но переход к следующему вопросу заменяется на переход к предыдущему заданию. В режиме контроля остаются только этапы постановки вопроса, редактирования ответа и переход к следующему заданию.

Возможны несколько типов заданий: а) выбрать один из предложенных ответов б) их совокупность, в) упорядоченная совокупность. Варианты ответов при выводе на экран перетасовываются. В случае (а) разработчик может потребовать от системы произвольно заменять во время работы любой ответ на фразу: «Правильный ответ отсутствует». В любом случае для ответа ученику необходимо отредактировать маску текста, которая при проверке сравнивается с эталоном. Хорошо подобранная маска позволяет таким простым способом интерпретировать ответы с многовариантной структурой (например, математические фор-

мулы), оставляя ученику всего лишь один, но нестандартно реализованный вариант записи.

Дидактический материал набирается в текстовом редакторе и записывается в RTF формате. Затем интерпретатор системы переводит его во внутренний формат. На рис. 2 представлена структура данных.

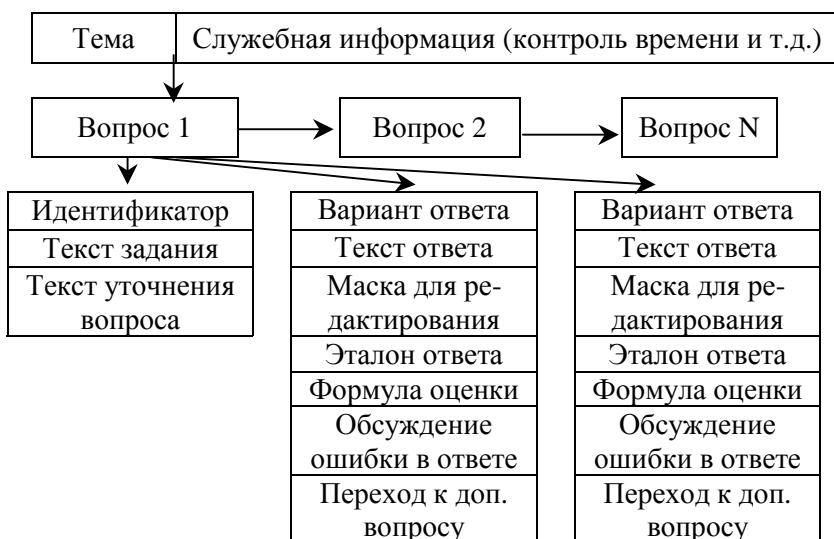


Рис. 2. Схема структуры данных. Исходные тексты могут сопровождаться картинкой.

Для работы с созданной структурой данных написан соответствующий интерпретатор, позволяющий использовать широкие возможности, заложенные в структуре данных. В частности, интерпретатор позволяет: применять ссылки на дополнительные вопросы (при неправильном ответе на вопрос, программа предлагает учащемуся ответить на вопросы, которые имеют смысловую связь с текущим заданием); создавать развитую систему помощи и подсказок; использовать в режиме обучения рисунки, анимацию, видеофрагменты и звуковые эффекты; строить гибкую оценочную систему, осуществлять возможность контроля времени прохождения всех заданий, или же времени на каждое задание.

Простая линейная схема интерпретатора (без рекурсий) позволяет очень просто расширять его возможности, вводя дополнительные команды и соответствующие процедуры исполнения.

Программа накапливает статистику ошибочных и правильных ответов, что позволяет преподавателю корректировать процесс обучения.

Набор структуры данных максимально упрощен для разработчика и выполняется в текстовом редакторе MSWORD. Для этого создан макрос, который вставляет блоками ключевые слова и выводит на экран подсказки по написанию текстового материала. Интерфейс программы выполнен в стандарте Windows 9x, что позволяет использовать программу без специальной подготовки ученика. Интерфейс помощи и подсказок ученику организован в стандарте WinHelp.

Требования к оборудованию и ОС: операционная система Windows 9x, Windows NT, процессор 486/40 и выше, 4 Мб ОЗУ, 690 Кб на жестком диске плюс текстовый и графический материал. При использовании анимации и звука: процессор Pentium 100 и выше, 16 Мб ОЗУ.

При локальном использовании программы существует возможность уменьшить размер как текстовых файлов (используя особенности формата RTF), так и графических файлов (путем применения формата JPEG с нужной величиной компрессии).

Литература:

1. Арнольд В.И. Теория катастроф. – М: Наука, 1990. – 128 с.
2. Машбиц Е.И. Психолого-педагогические проблемы компьютеризации обучения. – М.: Педагогика, 1988. – 192 с.
3. Освіта в школі. – №40. – 05.10.2000 р. Критерії оцінювання досягнень учнів у системі загальної середньої освіти.
4. Народная газета «Круг жизни». – № 05 (58). – 16.03.2001.

ДИНАМІЧНА КОМП'ЮТЕРНА МОДЕЛЬ ЦИФРОВИХ СХЕМ ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

М.С. Жуков, Ю.В. Темнюк, Р.О. Постоєнко, С.В. Діордіца
м. Кривий Ріг, Криворізький державний педагогічний університет

Сьогодні активно йде процес переведення всієї інформаційної техніки (телебачення, радіо, фото- і кінотехніки, систем зв'язку) на цифрову елементну базу. Здійснюється і комп'ютеризація побутової техніки, а також багатьох інших видів промислової продукції (від наручних годинників до автомобіля і літака). Є аргументовані прогнози про те, що ХХІ століття буде століттям глобальної цифрової техніки. Наприклад, зараз у Сінгапурі реалізується цілий ряд великомасштабних національних програм (у тому числі – у сфері освіти), метою яких є зробити країну до 2002 р. першою у світі державою цифрових інформаційних технологій [1].

Перед школою стоїть задача – підготувати молоду людину до життя в сучасному інформаційному суспільстві. Одними теоретичними заняттями проблему цю не вирішити. У школах і вузах практично цілком відсутня матеріальна база для проведення лабораторних занять з вивчення елементної бази цифрової техніки. Моделювання роботи окремих логічних, цифрових і багатофункціональних елементів, а також електричних схем на їхній основі не тільки заповнює зазначений недолік, але і дозволяє набагато швидше і дешевше організувати навчальний процес.

Візуалізація роботи цих елементів не тільки полегшує введення і сприйняття нових понять, створюючи відповідний графічний образ, але і значно розширює набір методичних засобів навчання, дозволяє внести динаміку в досліджуваний матеріал.

На першому етапі розробки моделюючої програми використовувався текстовий режим відеоадаптера. Це було зумовлено тим, що вона орієнтована для використання в школах і вузах, а вони, як правило, оснащені слабкою комп'ютерною технікою. Відомо, що при роботі в графічному режимі спостерігається деяке уповільнення швидкості роботи комп'ютера і використовується більше пам'яті, ніж у текстовому режимі. Зараз на завер-

шальному етапі знаходиться програма, що працює в операційній системі Windows 9x/NT.

Візуалізація і зв'язок об'єктів у моделі.

Об'єкти на екрані (прості і складні логічні елементи, тригери, лічильники, шифратори, дешифратори й ін.) відображаються в загальноприйнятому вигляді (рис. 1).

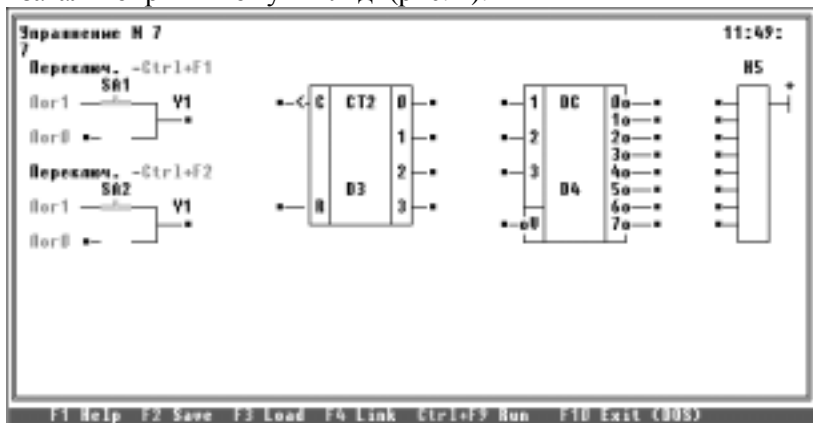


Рис. 1. Розміщення елементів до з'єднання.

Для з'єднання контактів двох мікросхем потрібно підвести курсор до відповідного контакту однієї мікросхеми і натиснути комбінацію клавіш Ctrl+Home (ознака початку з'єднання). Потім підвести курсор до контакту іншої мікросхеми і натиснути комбінацію клавіш Ctrl+End (ознака кінця з'єднання). Переміщення курсору здійснюється двома способами (рис. 2):

- без відображення лінії з'єднання – стандартними клавішами управління курсором;
- для малювання видимих ліній – з використанням так званого WORDSTAR-івського хреста.

Обмін інформацією між об'єктами моделі.

Кожен відображуваний об'єкт реалізований певною процедурою. Їй як параметр передається його порядковий номер і ліві верхні координати розташування на екрані. Процедура обчислює координати усіх контактів мікросхеми, зчитує з масиву станів значення в елементах з координатами входів, обробляє ці значення відповідно до функції даного об'єкта і записує результати в елементи масиву з координатами виходів об'єкта.

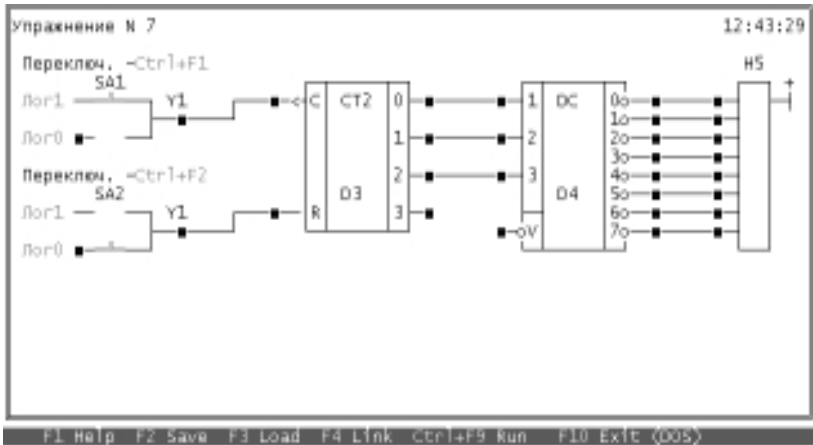


Рис. 2. Розміщення з'єднаних елементів.

Для передачі інформації з одного об'єкта в іншій реалізована функція зв'язку між об'єктами, яка переадресовує дані з елементів з координатами виходів однієї мікросхеми в елементи з координатами входів мікросхеми, з яким зв'язано вихід першої.

Перша версія програми розроблена для IBM-сумісних персональних ЕОМ серії PC/AT. Мова програмування – Турбо Паскаль. Для її роботи потрібна операційна система MS DOS, 181К оперативної пам'яті та близько 20К вільного простору на диску. Друга версія розробляється в середовищі Delphi 4. У ній, звичайно, значно більше зручностей і можливостей.

В програмі практичні завдання органічно зв'язані з відповідними теоретичними розділами. Таким чином, вивчення досить складного теоретичного матеріалу отримує практичне обґрунтування.

Програма являє собою навчальну систему, при розробці якої виходили з того, що кожен учень, який має мінімальні навички роботи з ПЕОМ, у змозі з нею працювати.

Література:

1. Колин К. К. Информатика на пороге XXI столетия / Сб. н. тр. «Системы и средства информатики». – М.: Наука, 1999. Вып. 9.

ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ПРИ ИЗУЧЕНИИ АБСТРАКТНЫХ ТИПОВ ДАННЫХ В ВУЗАХ

Н.С. Жуков, М.В. Швец, С.А. Томилин

г. Кривой Рог, Криворожский государственный педагогический университет

Задачи, решаемые на компьютере, являются математическими моделями процессов или явлений реальной жизни. Модели объектов вместе с присущими им связями образуют структуры данных. Языки программирования, ориентированные на решение различных классов задач, используют различные способы представления и обработки структур данных. Использование таких структур не представляет сложности для студентов.

Однако на практике широко используются такие абстрактные типами данных (АТД), как списки, очередь, стеки, графы, деревья и др. Представление внутренней организации данных в таких структурах способствует пониманию алгоритмов работы с ними. В связи с этим целесообразно создать визуальные компьютерные модели сложных структур данных, с помощью которых можно исследовать различные алгоритмы.

Для представления АТД используются структуры данных, которые представляют собой набор переменных, возможно, различных типов данных, объединенных определенным образом. Базовым строительным блоком любой структуры данных является ячейка, которая предназначена для хранения значения определенного базового или составного типа данных. Структуры данных создаются путем задания имен совокупностям ячеек и интерпретации значения некоторых ячеек как представителей (т.е. указателей) других ячеек. Таким образом, для реализации основных абстрактных типов данных используются указатели, связывающие последовательные элементы АТД. При работе с ними и возникают основные сложности, связанные с трудностью абстрактного представления таких структур данных и отсутствием визуальной модели, которая отображает работу динамической структуры. Поэтому для решения задачи и была создана обучающая программа по работе с такими типами данных, как оче-

редь, стек и список. Одним из видов АДД является очередь. Очередь – это тип данных, элементы которого могут быть одним из базовых типов данных, а основными операциями являются занесение элемента в очередь и извлечение из нее, правила которых указываются в самом определении. Для этого используются два соответствующих указателя. Для стека характерны аналогичные действия. При этом, однако, используется один указатель – вершина стека. Разница состоит в том, что очередь организована по принципу «первый зашел, первый вышел», а для стека – «первый зашел, последний вышел».

Предлагаемая разработка включает в себя обучающую программу, которая наглядно демонстрирует работу с вышеназванными структурами данных. Кроме этого, пользователь программы может получить теоретические сведения по выбранной тематике и программной реализации структуры на языке программирования Turbo Pascal.

Работа начинается из главного окна (рис. 1). Выбор интересующей темы (очередь, стек, список) производится через меню.

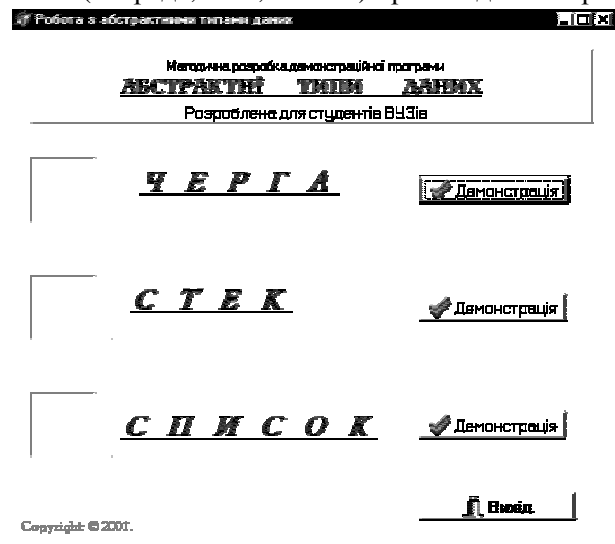


Рис. 1. Главное окно программы

Активизация соответствующей кнопки обеспечивает демонстрацию выбранной структуры. Управление последующими опе-

рациями производится посредством стандартных элементов Windows: кнопки, опции, селекторы и др. Для примера показан фрагмент работы программы с очередью (рис. 2).

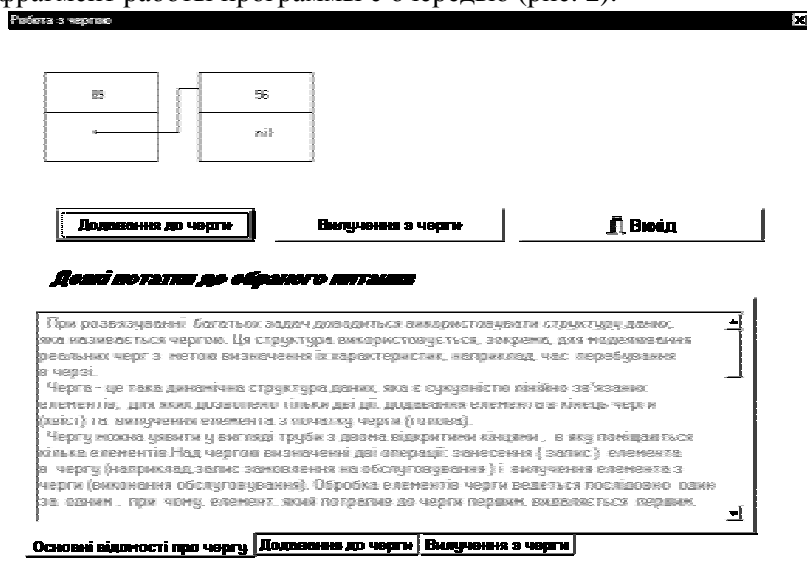


Рис.2. Формирование динамической очереди.

Здесь отображается графическая схема структуры выбранного типа данных, общие теоретические сведения и справка по организации соответствующих операций над выбранными типами данных.

В целом программа обеспечивает выполнение таких функций:

- 1) предоставление пользователю интересующих теоретических данных по выбранной теме, необходимых для построения одного из трех абстрактных типов данных;
- 2) обеспечение наглядности организации структуры соответствующих типов данных.

Обучающая программа разработана в среде Delphi 5 и рассчитана для работы в ОС Windows. В настоящее время в ней демонстрируется только три абстрактных типа данных. В стадии разработки находятся составляющие, которые будут демонстрировать более сложные структуры, такие как деревья, графы и др.

СИСТЕМА КОМПЬЮТЕРНОГО ТЕСТИРОВАНИЯ ЗНАНИЙ

О.А. Жульковский, А.В. Лисовец, И.А. Жульковский
г. Днепродзержинск, Днепродзержинский государственный тех-
нический университет

В учебных заведениях различного уровня для контроля знаний широко применяются тесты. Обычно тесты оформлены в виде карточек, которые, как известно, требуют периодического обновления. Поэтому очевидна актуальность создания системы компьютерного тестирования знаний. Поскольку подобного рода программы ориентированы, в основном, на функционирование в устаревшей операционной системе MS-DOS, а современный уровень развития вычислительной техники требует кардинального обновления программного обеспечения, то резонна разработка Windows-ориентированной системы тестового контроля.

Представляемая программа написана на языке Borland Pascal with Objects 7.0 с использованием редактора ресурсов Windows – Resource Workshop – для разработки элементов пользовательского интерфейса. Минимальные требования к аппаратуре: IBM-совместимый персональный компьютер с процессором 80386, VGA-монитор, манипулятор типа мышь.

Система снабжена встроенной утилитой для компиляции теста на базе предварительно подготовленных специальным образом текстовых файлов и позволяет генерировать необходимый перечень вопросов в зависимости от вида контроля (контрольная работа, зачет, экзамен и т.п.). В программе предусмотрен скроллинг поля теста с возможностью произвольного выбора очередности ответов на вопросы и корректировки принятого решения. Система снабжена таймером и полностью защищена от несанкционированного доступа со стороны пользователя.

Система опробована во время экзамена по дисциплине «Организация и функционирование ЭВМ и систем» в Днепродзержинском государственном техническом университете у студентов специальности «Программное обеспечение автоматизированных систем» и может быть использована в учебных заведениях любого уровня для компьютерного контроля знаний.

НАВЧАННЯ З ТОЧКИ ЗОРУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

М.І. Задорожній

с. Новоюлівка, Новоюлівська середня загальноосвітня школа

Характеристика інформаційних технологій навчання

Інформаційні технології навчання – це організація навчання учнів, яка

- ґрунтується на повному та глибокому вивченні структури навчальної інформації;
- використовує найбільш ефективні алгоритми навчальної діяльності учнів;
- враховує всі позитивні та негативні фактори, що впливають на результативність навчання;
- гарантовано забезпечує досягнення позитивного результату навчання при виконанні всіх необхідних умов;
- не передбачає обов'язкового використання комп'ютерної техніки на уроках, вона використовується лише там, де машина швидше, точніше, ефективніше ніж людина виконує обчислення, побудову графіків, розв'язування інших завдань;
- в навчально-виховній діяльності вчителя та учнів широко використовує поняття, закони та принципи інформатики, комп'ютерного програмування, інформаційних систем;
- формує в учнів аналітичний тип мислення, тобто вміння порівнювати, виділяти головне, складати план чи алгоритм, визначати найкращий з кількох правильних розв'язків, швидко і точно приймати правильні рішення в різних галузях діяльності людини.
- розвиває в учнів вміння і навички працювати з великими обсягами різноманітної інформації різного рівня складності;
- включає в себе поняття комп'ютерних технологій навчання, які повністю пов'язані з використанням ПК, їх ще називають нові інформаційні технології навчання.

Що таке навчання?

Навчання – це інтелектуальний труд учня. Цей труд – важкий, складний, неперервний, довготривалий, самостійний та доб-

ровільний.

Якщо учень на уроці чи вдома не працює самостійно, а лише виконує окремі завдання з допомогою вчителя, батьків, інших учнів, тоді навчання для нього перетворюється на гру.

Якщо учень зовсім не виконує ніяких завдань, тоді навчання для нього стає театральною виставою.

Якщо учень заважає навчатись іншим учням та працювати вчителю, тоді навчання стає злочином.

Завдання всіх інших учасників навчального процесу – створити такі умови для навчання учнів, щоб вони досягали найвищих результатів при мінімальних затратах сил та часу.

Учень та учитель в умовах гуманізації освіти

В умовах гуманізації освіти, коли вчитель і учень виступають як рівноправні особистості, результати навчання залежать від учня в значно більшій мірі, ніж раніше.

Учень самостійно обирає профіль навчання, рівень складності та об'єм теоретичних знань, вправ, задач, домашніх завдань.

Завдання вчителя:

- опрацювати навчальну інформацію: впорядкувати її, вибрати головне, подати її учням в зручному вигляді;
- підібрати таку послідовність завдань для учнів, яка забезпечує неперервність та повноту навчального процесу і дає можливість досягти позитивного результату;
- об'єктивно оцінити навчальні досягнення учнів.

Труд учителя – це теж важкий та складний інтелектуальний труд, який потребує значних затрат сил та робочого часу, тому це обов'язково треба враховувати при управлінні навчальним процесом.

Структура навчального процесу

Найменшою структурною одиницею навчального процесу, що містить повний функціональний цикл, є тема. Всередині теми планується кілька циклів

ВИВЧЕННЯ...– ЗАСВОЕННЯ... – ФОРМУВАННЯ...

В кінці теми обов'язковий цикл

ПОВТОРЕННЯ... – КОНТРОЛЬ...

Вивчення нових знань здійснюється не більше ніж на половині уроків, що відводяться на вивчення теми. Тому новий матеріал об'єднується у досить великі блоки навколо кількох найважливіших питань. Пояснення вчителя містить лише найважливіші елементи знань, а не довільний переказ матеріалу підручника.

Засвоєння нових знань здійснюється за допомогою запитань та вправ розвиваючого характеру. Ці запитання та вправи носять практичний характер, доступний та зрозумілий більшості учнів.

Формування умінь і навичок здійснюється за допомогою виконання вправ та розв'язування задач. Починати необхідно з найпростіших, поступово збільшуючи їх складність. Вчитель пропонує учням список вправ або задач, а учні самостійно обирають рівень складності та кількість задач, що ними виконуються. В умовах загальноосвітньої школи по іншому діяти і не можливо, тому що в одному класі навчаються учні трьох-чотирьох рівнів навчальних досягнень і для підбору завдань кожній групі учнів треба дуже багато часу, якого не має.

Повторення та систематизація знань є обов'язковим етапом перед контролем знань. Проведення такого контролю без повторення є некоректним по відношенню до учнів. Цей етап є не механічним повторенням вивченого матеріалу, а має на меті виділити головне, об'єднати основні знання в єдине ціле, проаналізувати зв'язки між окремими елементами знань.

За змістом навчального матеріалу його систематизація здійснюється окремо для теоретичних знань, вправ та задач, практичних робіт. При недостатній кількості навчального часу перевага надається тому виду навчального матеріалу, за яким буде проводитись тематичний контроль.

Контроль навчальних досягнень учнів

Контроль знань поділяється на кілька видів у залежності від обсягу навчального матеріалу, який підлягає контролю та часу протягом якого він вивчався.

Поточний контроль – матеріал одного або кількох уроків, що вивчався не більше тижня.

Тематичний контроль – матеріал однієї теми, що вивчався не більше місяця.

Семестровий контроль – матеріал одного розділу або кількох тем і вивчається не більше одного семестру.

Річний контроль – матеріал предмету, що вивчався не більше одного року.

Підсумковий контроль – матеріал за весь курс навчального предмету, що вивчався протягом кількох років.

За змістом навчального матеріалу контроль поділяється на три види:

- контроль теоретичних знань;
- контроль виконання вправ та розв’язування задач;
- контроль практичних робіт.

Всі ці види контролю можуть здійснюватись як в усній, так і в письмовій формі з обов’язковою рівневою диференціацією складності завдань.

Диференціація знань учнів та система їх оцінювання

Диференціація знань учнів і відповідно їх оцінювання здійснюється за чотирма рівнями.

ТВОРЧИЙ РІВЕНЬ – самостійне одержання нових знань.

Усно – повна правильна відповідь на запитання з елементами знань, одержаних самостійно з інших джерел.

Вправи та практичні роботи – правильні виконання та результат завдання третього рівня складності.

АНАЛІТИЧНИЙ РІВЕНЬ – перетворення готових знань.

Усно – виділення головного, порівняння, систематизація, складання плану або алгоритму.

Вправи та практичні роботи – правильні виконання та результат завдання другого рівня складності.

РЕПРОДУКТИВНИЙ РІВЕНЬ – відтворення готових знань.

Усно – самостійна правильна відповідь на запитання одним або кількома повними реченнями.

Вправи та практичні роботи – правильні виконання та результат завдання першого рівня складності.

ФРАГМЕНТАРНИЙ РІВЕНЬ – окремі частини готових знань.

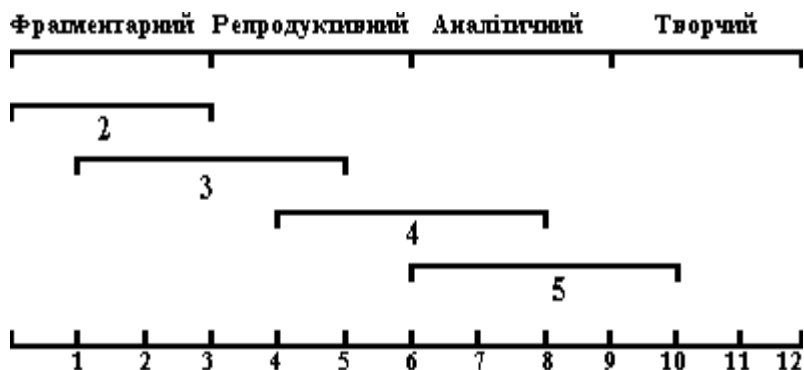
Усно – відповідь неповними реченнями або за допомогою зошита, підручника, вчителя, інших учнів.

Вправи та практичні роботи – виконання та результат з по-

милками хоча б одного завдання.

Порівняння старої 4-бальної та нової 12-бальної системи оцінювання подано в таблиці. Оцінка старої системи

– це великий проміжок, який не має чіткої межі. Бал нової системи – це чітко визначена точка на шкалі знань.



Література:

1. Задорожній М.І. Інформаційні технології навчання в загальноосвітній школі // Комп'ютерне моделювання та інформаційні технології в освітній діяльності. – Кривий Ріг: Видавничий відділ КДПУ, 1999. – 249 с. – С. 158–168.
2. Задорожній М.І. Алгоритм розв'язування фізичних задач для комп'ютера та учнів // Комп'ютерне моделювання та інформаційні технології в природничих науках: збірник наукових праць. – Кривий Ріг: Видавничий відділ КДПУ, 2000.– 462 с. – С. 350–356.

ВИВЧЕННЯ ЧИСЕЛЬНИХ МЕТОДІВ РОЗВ'ЯЗУВАННЯ АЛГЕБРАІЧНИХ РІВНЯНЬ У ШКОЛІ

М.М. Задорожній

м. Кривий Ріг, Криворізький обласний ліцей-інтернат для сільської молоді

Існує багато способів чисельних методів розв'язування рівнянь n -го степеня виду $F(x)=0$, тобто за допомогою обчислення значень $F(x)$ для великої кількості значень x , що змінюються з невеликим приростом dx , та перевіркою чи належить цьому інтервалу нульове значення $F(x)$. Якщо ця умова виконується, то це значення x є коренем рівняння з точністю dx .

Ці методи відрізняються способом поділу інтервалу значень аргументу $[a; b]$ на частини, способом визначення довжини інтервалу dx , та умовами, які необхідно виконати для одержання коренів.

Перший метод, який ми розглянемо, – це метод половинного поділу. Цей метод застосовується для невеликого інтервалу $[a; b]$, на якому існує лише один дійсний корінь. Функція $F(x)$ на цьому інтервалі повинна бути неперервною і задовольняти умову:

$$F(a) \cdot F(b) < 0.$$

Для знаходження кореня відрізок $[a; b]$ ділиться пополам і визначається та його частина, на кінцях якої зафіксована зміна знаку. Процес поділу продовжується доти, поки довжина інтервалу не стане меншою за вказану точність dx коренів.

Другим методом є метод дотичних. Цей метод для обчислення дійсних коренів потребує вказувати початкове наближення кожного кореня, що задовольняє умову:

$$|F(x_i) \cdot F''(x_i)| < |F'(x_i)|^2,$$

а приріст аргументу визначається за формулою:

$$dx = F(x_k) / F'(x_k).$$

При цьому ітераційний процес збігається до точного значення кореня і припиняється при $dx < 0$.

Наступний метод – це метод хорд. Він подібний до методу дотичних і використовує замість обчислення похідної її оцінку за формулою:

$$F'(x) = (F(x+dx) - F(x)) / dx.$$

Метод січних-хорд застосовується, коли на проміжку $[a; b]$ неперервна функція $F(x)$ задовольняє умову:

$$F(a) * F(b) < 0.$$

При цьому інтервал $[a; b]$ ділиться на дві частини точкою

$$x = a - F(a) * (a - b) / (F(a) - F(b)).$$

Далі для поділу вибирається та частина проміжку, на кінцях якої функція $F(x)$ має протилежні знаки.

Метод Рибаківа (прямокутників) дає можливість знайти всі дійсні корені рівняння при умові, що $F'(x)$ обмежена.

За початкове наближення приймається $x = a$.

$$dx = F(x_i) / M$$

де $M = \max F'(x)$ на проміжку $[a; b]$.

x_i вважається коренем рівняння. Наступне наближення

$$x = x_i + dx.$$

Інші методи розв'язування рівнянь потребують ще більш складних обчислень.

Для використання цих методів у практичній програмі необхідно попередньо будь-яким способом виконати ізоляцію коренів, тобто виконати наближені обчислення.

Тепер розглянемо алгоритм розв'язання рівнянь n -го степеня.

1. Введення коефіцієнтів рівняння.

Для створення простої та невеликої за розміром програми введення коефіцієнтів рівняння здійснюється в командному рядку ДОС.

2. Введення меж та точності.

Межі значень аргументу і точності ізоляції коренів та значення коренів виводиться з файлу, який створюється за допомогою будь-якого текстового редактора і містить лише чотири рядки з числами довжиною не більше 10 символів.

3. Друкування на екрані тексту рівняння.

На екран виводиться текст рівняння та межі області значень. При цьому враховуються варіанти запису рівнянь для степенів менших за 2-й.

4. Обчислення значень многочлену.

Обчислення значення функції здійснюється за схемою Горнера.

5. Ізоляція коренів.

Для ізоляції коренів використовують побудову графіка функції. Для цього існують спеціальні програми, наприклад, Gran1, Eureka, які дають можливість знаходити і самі корені з великою точністю.

Для автоматичного визначення проміжків програмою застосовується метод прямокутників з досить великим кроком. Це дає можливість при задовільній швидкості знаходити корені рівняння на великих проміжках. Цей крок для ізоляції коренів в більшості випадків не може перевищувати 0,1. При більших значеннях кроку цей проміжок може містити більше, ніж один корінь.

За корінь рівняння приймається той кінець інтервалу, на якому модуль значення многочлену менший. Це дає можливість точніше знаходити корінь рівняння. Всі корені та відповідні значення многочленів запам'ятовуються.

6. Обчислення коренів на інтервалі.

Для обчислення коренів на вибраному інтервалі застосовуються три методи:

- прямокутників з меншим кроком зміни значень аргументу;
- половинного поділу;
- січних-хорд.

7. Друкування коренів рівняння на екран.

Значення коренів виводяться на екран у вигляді:

$$F(-2.0000)=0.0000$$

що дає можливість оцінити похибку обчислень.

Якщо рівняння не має дійсних коренів, то на екран виводиться відповідне повідомлення.

8. Вдосконалення програми.

Для вдосконалення програми можна випробувати різні методи ізоляції та обчислення коренів з тестуванням їх на швидкість виконання, повноту, точність коренів.

Також можна застосувати режими читання рівнянь з файлу та запису розв'язків у файл.

МЕТОДИКА ВИКОРИСТАННЯ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ДЛЯ ЕФЕКТИВНОЇ ПІДГОТОВКИ ДОКУМЕНТІВ

Л.А. Карташова

м. Київ, Київський державний лінгвістичний університет

Для створення професійно виконаних документів може бути використане середовище MS Word. Часто виникає потреба у ефективній роботі з існуючими документами. Наприклад, необхідно коротко описати зміст документу, тезисно викласти основні питання, які розглядаються в документі, здійснити переклад фрагментів тексту на іноземну мову і т.д.

Пропонується зручний та ефективний підхід, який полягає в створенні інтегрованого інструментарію для роботи з такими документами як автореферат дисертаційної роботи, лекційний матеріал, звітні роботи, презентації з використанням ряду особливостей середовищ MS Word, Power Point, макросів та подібних їм. Застосування командних засобів програми Microsoft Word за цим зразком прискорить та полегшить роботу по створенню таких документів. Інструментарій подається у вигляді панелі інструментів MS Word і забезпечує реалізацію всіх функціональних можливостей, необхідних для ефективної роботи з існуючими документами.

Рекомендується використовувати форматування документів при допомозі стилів. Стиль – комплекс параметрів форматування, які застосовують до тексту документу для того, щоб швидко змінити його зовнішній вигляд. Використання стилю дозволяє за допомогою однієї дії застосувати відразу всю групу атрибутів форматування. За їх допомогою можна наприклад, оформити назву розділу документу. Microsoft Word містить бібліотеку стилів, яка дозволяє представити вид документу в цілому після застосування стилю з другого шаблону.

Для полегшення роботи при створенні документів в текстовому процесорі Word рекомендується використовувати режим перегляду структури документу. Це спеціальний режим, призначений для наглядного представлення внутрішньої організації документу. Структура відображається як багаторівневий список з

назв розділів. Заголовки розділів різних рівнів відрізняються по стилю та розміру відступу. Всього може бути до 9 рівнів заголовків. Якщо показувати тільки заголовки, можна отримати короткий реферат документу.

Невід'ємною частиною наукового документу являються зноски, які використовуються для розміщення додаткової інформації.

В деяких випадках документ може бути створений на основі іншого великого документу або кількох документів. Тобто використовуються витяги з них. В таких випадках рекомендується використовувати інструментальні засоби Microsoft Word такі як копилка (Spike). Скарбничка – це спеціальний елемент який дозволяє витягти фрагменти з різних місць документу або з різних документів, зібрати їх разом в нове місце документу або в інший документ. В скарбничку можна занести як текст, так і графіку.

Прискорює та полегшує таку роботу використання спеціальних панелей інструментів, створених користувачем. Як правило, на таких панелях виводяться піктограми команд, до яких найчастіше звертаються при створенні даного документу чи кількох документів.

Також зручно користуватися скарбничкою при допомозі відповідних кнопок на панелі інструментів. На стандартних панелях такі кнопки відсутні. Але їх неважко вивести на будь-яку панель із списку або на створену користувачем.

Розглянемо один із способів створення особистої панелі інструментів, на яку будуть виведені піктограми команд, що найчастіше використовуються в роботі над цим документом. В даному випадку це робота з Spike. Ввійти в пункт меню Вид ⇒ Панелі інструментов ⇒ закладка Панелі інструментів ⇒ Налаштування ⇒ Создать ⇒ на екрані з'явиться діалогове вікно “Создание панели инструментов”, в якому в рядку “Панель инструментов” присвоїти ім'я особистій панелі інструментів (рис. 1).

В рядку “Сделать доступной для” при необхідності можна вибрати ім'я файлу до якого буде прив'язана ця панель. Це значить, що створена панель буде з'являтися тільки в вікні Microsoft Word при роботі з даним документом. В інших вікнах цю панель викликати для роботи неможливо. Її навіть не буде в загальному

списку панелей “Панели инструментов”.

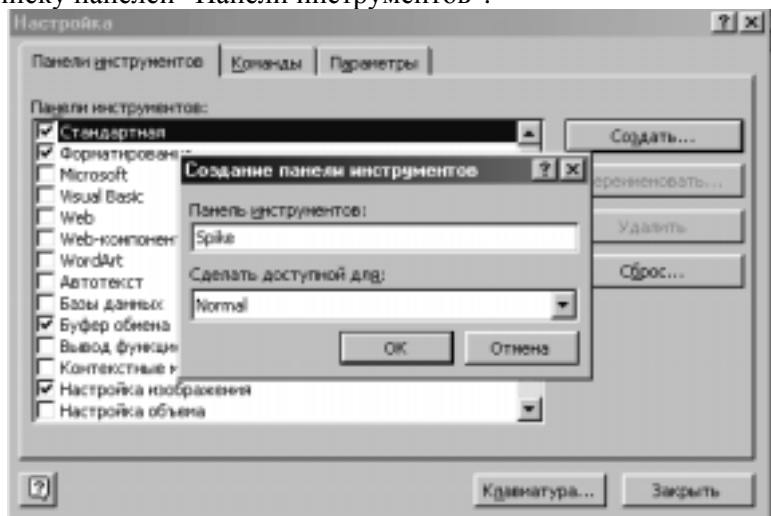


Рис. 1. Вікно створення панелі інструментів

Якщо ж користувачу необхідно створити панель, кнопки якої будуть використовуватись при роботі з різними документами, то в рядку “Сделать доступной для” вибрати Normal.dot.

Коли панель з’явилася у вікні документу, то на неї можна вивести необхідні команди. В даному випадку продовжуємо працювати з командами, які стосуються копилки. Вийти в пункт меню Вид ⇒ Панели инструментов ⇒ закладка Команды ⇒ вибрати категорію Все команды ⇒ Spike (рис. 2).

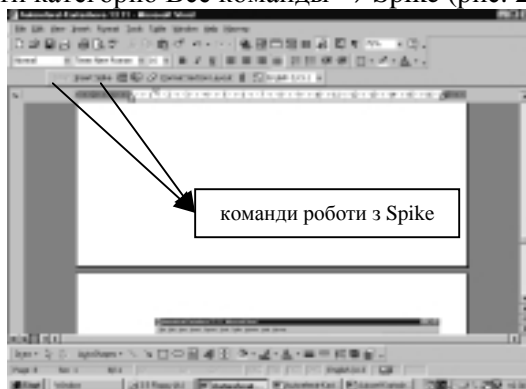


Рис. 2. Приклад створеної користувачем панелі інструментів.

Команду Spike методом протягування перемістити на створену панель інструментів. Кнопка Spike буде використовуватися для накопичування інформації. Вона стане активною тільки при виділенні фрагменту тексту, малюнка, схем та інш. Далі таким же чином в категорії “Все команди” вибрати команду Insert Spike та перетягти на панель інструментів. Ця кнопка буде активна тоді, коли в копилці буде міститися хоч би один фрагмент документу.

Кнопка Insert Spike дозволить вставити текст із скарбнички в текст документу. Текст заноситься в те місце документу, де стоїть курсор.

З великим документом легко працювати, якщо він обладнаний предметним покажчиком. Покажчик містить список термінів та понять, які використовуються в документі та мають особливе значення. В списку є вказівка на номери сторінок, які зв’язані з цими поняттями. Якщо документ дуже об’ємний та містить велику кількість слів, які необхідно ввести пропонується використовувати індексний файл (словник). Словник – це окремий файл, створений в вигляді таблиць та термінів, які повинні бути занесені в предметний покажчик.

Необхідно звернути увагу на те, який вид повинна мати така таблиця. Це має принципове значення. Необхідно відкрити нове вікно для нового документу ⇒ створити таблицю на дві колонки ⇒ в першу чергу занести текст, який необхідно знайти в основному документі та відмітити як елемент покажчика ⇒ ТАВ ⇒ в другу колонку занести текст, який необхідно занести в предметний покажчик ⇒ ТАВ. Для введення додаткового елемента предметного покажчика – ввести головний елемент ⇒ дві крапки (:) ⇒ текст додаткового елемента.

Повторити ці кроки для всіх елементів предметного покажчика та зберегти цей документ як окремий файл.

- Для прискорення роботи з файлом словника рекомендується створювати словник при допомозі скарбнички. В першу колонку необхідно занести всі можливі форми тексту, який буде введено в покажчик, в другу колонку занести текст в тому вигляді, який він повинен мати в покажчику.

- Потрібно врахувати, що кожне слово повинно бути занесено в окрему комірку, як в правій, так і в лівій колонці.

Приклад таблиці словника:

текстом	текст
тексту	текст
тексті	текст

Файл словника використовується для автоматичної помітки елементів предметного покажчика в документі – Вставка ⇒ Оглавление и указатели ⇒ Указатель ⇒ Автопометка ⇒ відкрити файл словника – автоматично відмічаються всі входи в тексті – закривається вікно ⇒ Вставка ⇒ Оглавление и указатели ⇒ Указатель – вибрати вид списку елементів покажчика ⇒ ОК.

Інколи виникає необхідність редагування тексту, який занесено до списку покажчика. В такому випадку потрібно внести зміни в основному тексті, а потім встановити курсор зліва від списку та натиснути F9. Виконається автоматичне поновлення тексту.

Коли в Word необхідно багаторазово виконувати одну і ту ж задачу, то краще це зробити при допомозі макросів. Макрос — це серія команд, згрупованих разом для прискорення повсякденної роботи. Замість того, щоб вручну виконувати дії, які повторюються та займають багато часу, можна створити та запускати один макрос, який буде виконувати цю задачу. Макроси часто використовуються для наступних цілей: для прискорення операцій редагування та форматування, які часто виконуються; для об'єднання кількох команд; для спрощення доступу до параметрів в діалогових вікнах; для автоматизації обробки складних послідовних дій в задачах.

Для створення макросу в Word можна використовувати два методи: за допомогою засобів для запису макросів чи редактора Visual Basic. В редакторі Visual Basic можна також написати дуже потужні та складні макроси, які не можуть бути записані.

Створення інтегрованого інструментарію для роботи з великими документами (автореферат дисертаційної роботи, лекційний матеріал, звітні роботи, презентації) з використанням ряду особливостей середовищ Microsoft Office забезпечує реалізацію всіх функціональних можливостей, необхідних для ефективної роботи з документами.

РОЗВИТОК ПІЗНАВАЛЬНОЇ АКТИВНОСТІ ШКОЛЯРІВ ЗАСОБАМИ КОМП'ЮТЕРНОГО НАВЧАННЯ

Н.П. Козаченко

м. Кривий Ріг, Криворізький державний педагогічний університет

Питання розвитку пізнавальної активності школярів як однієї з основних передумов успішного навчання завжди було і є важливою проблемою педагогіки (методики, дидактики). Адже, як казали древні мудреці: «Знання – це пожива для розуму, і, щоб вони краще засвоювалися, їх потрібно поглинати з апетитом».

Практично кожна дитина народжується з бажанням і здатністю пізнавати світ в усіх його проявах. Всі діти йдуть до школи прагнучи здобувати знання, добре вчитись і більшості з них в початкових класах це вдається. Але потім, особливо в середніх та старших класах цей інтерес до знань значно згасає, пізнавальна активність істотно знижується, що негативно впливає на якість знань та навичок учнів, і, як наслідок, на підготовку до майбутньої професійної діяльності та соціальної активності. Тому завдання вчителя полягає в тому, щоб різними шляхами та методами розвивати пізнавальну активність дітей, спонукати їх до постійної творчої розумової діяльності, формувати в них органічну потребу постійно поповнювати свої знання та самовдосконалюватись.

Проблема пізнавальної активності досліджувалась багатьма вченими, підхід до неї неоднозначний. Досить точно й повне означення пропонує М.С. Головань: «Пізнавальна активність – це складне інтегративне утворення особистості, що має структуру, яка складається з трьох компонентів: мотиваційного, змістовно-операційного та емоційно-вольового».

До мотиваційного компоненту автор відносить пізнавальну потребу, пізнавальний інтерес, спрямованість особистості на навчальну діяльність, позитивне ставлення до учня, надситуативність, пізнавальну ініціативу. До змістовно-операційного компоненту – систему ведучих знань та способів учіння, всі мислительні операції, пізнавальні процеси, пов'язані з розумовими перетвореннями у пізнанні. Емоційно-вольовий компонент пізнава-

льної активності охоплює такі якості особистості як старанність, схильність до дискусії, вдумливість, впевненість у собі, почуття власної гідності, відсутність страху помилки, вміння організувати свою навчальну діяльність, цілеспрямованість в роботі.

Сьогоднішня привносить свої корективи в зміст методів розвитку пізнавальної активності. Педагогічна наука, нажаль, все ще відстає від шкільної практики. Враховувати програмні зміни у викладанні предметів не встигають і підручники. Досить активно на будь-які зміни реагує програмне забезпечення комп'ютерів. Найбільш якісні із сучасних педагогічних програмних засобів виконані з урахуванням дидактичних та технічних вимог до забезпечення комп'ютерного навчання. Під комп'ютерним навчанням будемо розуміти навчання, в якому комп'ютер виступає засобом навчання.

Виходячи з концепції О.М. Леонтьєва про постадійний розвиток дитини і беручи за основу потребу у знаннях, можна визначити основні ступені, які проходить в своєму розвитку пізнавальна активність дітей:

1. Допитливість, викликана первинним інтересом до навчання й потребою в пізнанні навколишньої дійсності.
2. Відповідальне ставлення до навчання, зумовлене суспільно значущими мотивами.
3. Пізнавальний інтерес.
4. Пізнавальна самостійність.
5. Науковий пошук, творчість.

У пропонованій статті зупинимося саме на розвитку пізнавальної самостійності, яка, у свою чергу, формується на базі активності.

Нажаль, сьогодні порівняно невелика кількість шкіл забезпечена комп'ютерами на потрібному рівні. Але кількість їх в індивідуальному користуванні невпинно зростає. Все більше школярів мають змогу працювати з комп'ютером удома. Великою методичною помилкою вчителя було б ігнорування цього факту. Педагог повинен допомогти учню перетворити комп'ютер з іграшки на незамінного помічника і мудрого наставника. Для цього потрібно внести необхідні корективи в дидактичну структуру організації самостійної роботи учнів, пристосувати до роботи з комп'ютером. Сучасна система оцінювання знань резервує най-

вищий бал (“12”) для оцінювання самостійної, творчої, пошукової роботи. Учитель повинен навчити учня використовувати комп’ютер саме для досягнення цієї мети.

Пізнавальна самостійність характеризується багатьма вченими як властивість чи якість особистості. Так, М.М. Скаткін визначав активність і самостійність як якість особистості та вказував, що пізнавальна самостійність формується на базі активності. Ознаками пізнавальної самостійності при цьому є прагнення та вміння самостійно мислити, здатність орієнтуватися в новій ситуації, знаходити свій підхід до нової задачі, бажання не лише зрозуміти засвоювані знання, а й способи їх здобування, критичний підхід до суджень інших та незалежність власних суджень.

Л.С. Коновалець розглядає пізнавальну самостійність як якість особистості, що поєднує у собі вміння набувати нові знання та творчо застосовувати їх в різних ситуаціях із прагненням до такої роботи. Цей феномен являє собою єдність двох компонентів – мотиваційного та процесуального (змістовно-операційного). Перший відображає потребу у процесі пізнання, другий – знання даної предметної області та прийомів діяльності, що сприяють здійсненню цілеспрямованого пошуку.

Розділяючи поняття активності та самостійності, Л.П. Арістова зазначає, що сутність самостійності – у здатності об’єкта діяти без сторонньої допомоги. І.Я. Лернер вважає активність умовою самостійності, оскільки не можна бути самостійними, не будучи активним, і головне завдання вбачає в тому, щоб підняти активність до рівня самостійності.

В.А. Крутецький встановлює зв’язок між самостійністю та активністю як відношення між поняттями «активне мислення», «самостійне мислення» та «творче мислення» і пропонує представити його у вигляді концентричних кіл. Це різні рівні мислення, з яких кожен наступний є видовим по відношенню до попереднього, родового. Він підкреслює, що мислення буде самостійним та активним, проте не всяке активне мислення є самостійним і не всяке самостійне мислення є творчим.

Розвиваючи самостійність учнів, вчитель розвиває практично кожен компонент структури пізнавальної активності. Головну цінність самостійних дослідницьких робіт складає те, що вони вчать учнів не лише розв’язувати, але й ставити задачі, планува-

ти дії і способи їх виконання, шукати нові, об'єктивно цінні методи їх розв'язку.

Складаючи програму самостійної діяльності учня бажано передбачати ті процеси, що відбуваються при виконанні роботи в практичних діях і в пізнанні учнів. Досвід кращих вчителів та методистів показує, що будь-яка форма самостійної діяльності учнів при педагогічно доцільній її організації сприяє підвищенню пізнавального інтересу та активності. Таким чином, доцільними були б зміни співвідношення в методиці навчання між інформативними і такими активними методами, що потребують самостійності учнів. Будь-яка самостійна робота повинна передбачати ясні дидактичні цілі навчання і спиратись на активні пізнавальні процеси. При цьому важливо, щоб сам учитель розумів, активізації яких психічних процесів потребує виконання певних видів роботи: яка робота розвиває гнучкість мислительних операцій та уяву, яка робота опирається на чисто репродуктивну, яка на творчу діяльність. Творче завдання дає змогу відносно самостійно вибрати напрям розширення чи поглиблення обсягу знань. До того ж знання, набуті самостійно, мають незрівнянно більшу суб'єктивну цінність, ніж колективно здобуті. Самостійна робота учнів з використанням комп'ютера – один із найбільш удалих способів реалізації принципу індивідуалізації навчання, спрямований на підвищення пізнавального інтересу та власної пізнавальної активності учня.

Як приклад пропонуємо план самостійної роботи з теми «Гармонічні коливання». У старших класах учні отримують початкові відомості про гармонічні коливання і починають використовувати їх у різних галузях. Шкільна практика показує, що це відбувається без належного усвідомлення, що зумовлюється обмеженістю навчального часу. Маючи комп'ютер удома, учень може сам опанувати цей матеріал. Така робота повинна бути оцінена вчителем як виконання творчого завдання і принести подвійну користь: більш глибоке усвідомлення матеріалу і бажані 12 балів.

У шкільному курсі подано способи аналітичного представлення гармонічних коливань (вільних та вимушених), вводяться поняття частоти, амплітуди та фази.

Графічним представленням гармонічного коливання є графі-

ки функцій $y=A\cdot\cos(\omega\cdot t+\varphi)$ або $y=A\cdot\sin(\omega\cdot t+\varphi)$, де A – амплітуда, ω – частота, φ – початкова фаза. Наочно побачити це ученя зможе, скориставшись ППЗ GRAN1.

Для поглиблення обсягу знань із даної теми ми пропонуємо учням для самостійного виконання наступну систему завдань.

1. Побудуйте графік гармонічного коливання, амплітуда якого 2, частота 0.25, причому в початковий момент руху зміщення дорівнювало нулю.

2. Дослідіть графіки гармонічних коливань при різних параметрах (змінюючи значення амплітуди, частоти та фази). Зробіть загальний висновок про залежність вигляду графіків від параметрів коливань.

3. Дайте відповіді на питання:

Чим відрізняються графіки коливань, що мають різні частоти ($\omega_2=2\omega_1$)?

Графіками вільних чи вимушених коливань є графіки функцій $y=A\cdot\cos(\omega\cdot t+\varphi)$ та $y=A\cdot\sin(\omega\cdot t+\varphi)$?

Чим характеризуються вільні коливання? Як можна аналітично задати вільні коливання? Як виглядає графік таких коливань?

Якими міркуваннями керуються, обираючи вид функції (\sin або \cos) при аналітичному поданні коливань?

Зі шкільного курсу учням відомо, що реальні вільні коливання завжди затухаючі. Проте характер зміни амплітуди необхідно конкретизувати. З цією метою можна розглянути коливання, що визначаються рівнянням

$$y=e^{-x}\sin(x),$$

які часто реалізуються у системах із в'язким тертям (рис. 1).

Для наочності на цьому рисунку зображено і графік функції $y=e^{-x}$, що відображає зміну амплітуди коливань.

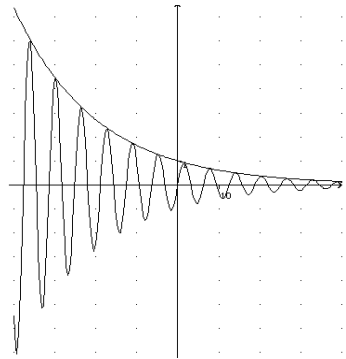


Рис. 1

Цікаву графічну реалізацію також має відоме явище додавання коливань. Виконати додавання графіків вручну досить

важко, в цьому випадку доцільно використати можливості GRAN1, а саме – параметричне подання функцій.

Завдання: Графічно зобразіть результат додавання двох коливань, розташованих уздовж однієї осі.

В GRAN1 це можна зробити, задавши в полі **Функція** суму функцій, якими задано коливання. При невеликій різниці в частотах отримаємо *биття*.

На рис. 2 зображений типовий результат додавання гармонічних коливань вздовж однієї осі

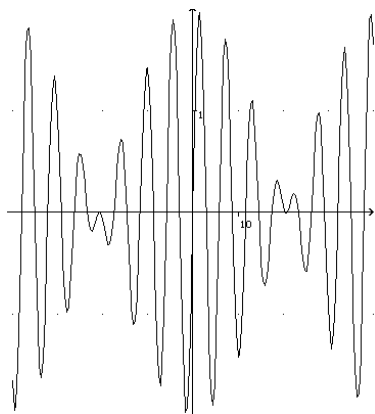


Рис.2

для випадку коли функція суми має такий вигляд: $y = \sin(x) + \sin(1.15x)$ на проміжку $x \in (-40; 40)$.

Виконати додавання таких коливань можна шляхом подання функцій у параметричному вигляді. Для цього в полі **опції** – **встановити тип** вибрати параметричне подання і в полі **функція** вказати коливання, які будуть здійснюватись по осі абсцис і по осі ординат.

На рис. 3 приведений результат додавання коливань, заданих такими рівняннями:

$$y(t) = \sin(4t), \quad x(t) = \sin(5t).$$

Завдання: Графічно зобразіть результат додавання двох ко-

ливань, що відбуваються вздовж взаємно перпендикулярно

Завдання: За графіком визначити частоту і період результуючого коливання. Порівняти отримані значення з обчисленими за формулою.

GRAN1 також надає можливість дослідити результат додавання коливань, що відбуваються вздовж взаємно перпендикулярно

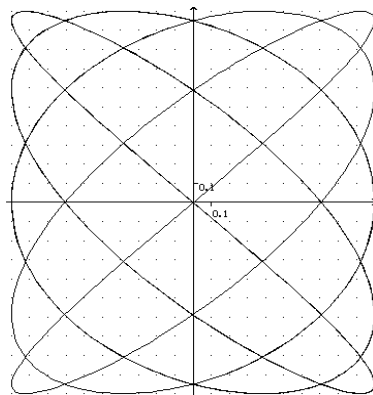


Рис. 3

ливань, що відбуваються вздовж взаємно перпендикулярних напрямів.

Завдання: Дослідіть додавання коливань при частотах, відношення яких становить 2:3, 1:3 тощо.

На рис. 4 показана фігура, утворена в результаті додавання коливань, частоти яких відносяться як 3:1. $y(t)=\sin(3t)$, $x(t)=\sin(t)$ при початкових фазах, що дорівнюють нулю.

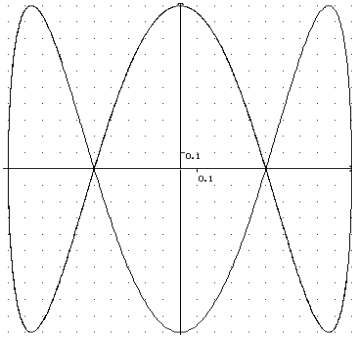


Рис. 4

Експериментуючи з додаванням взаємно перпендикулярних коливань, учень зможе з'ясувати залежність результату від параметрів коливань. Фігури, утворені таким чином, носять назву фігур Лісажу. В залежності від співвідношення частот коливань і початкової фази руху отримуємо різні фігури.

У пропонованій системі завдань реалізується принцип проблемного навчання. Передбачається значний ступінь самостійності учнів у розв'язанні навчальної проблеми, сформованої на основі аналізу вихідної постановки навчального завдання. Не менш важливою умовою підвищення активності школярів є використання міжпредметних зв'язків у творчій роботі.

Використання комп'ютера при самостійному виконанні творчої роботи значно розширює можливості подання учбової інформації. Застосування кольору, графіки, всіх сучасних засобів відеотехніки дозволяє відтворювати дійсні умови діяльності. Комп'ютер дозволяє посилити мотивацію навчання. Можливість інтерактивно працювати з програмою (змінювати значення параметрів і одразу ж спостерігати за змінами результату) та регулювати подання учбових задач за складністю.

Комп'ютер впливає на мотивацію учіння, розкриваючи практичну значущість матеріалу, що вивчається, надаючи учням можливість випробувати розумові сили і проявити оригінальність, поставивши цікаву задачу, задавати будь-які питання і пропонувати будь-які розв'язки без ризику отримати за це низький бал, – все це сприяє формуванню позитивного відношення

до навчання.

Спілкування з комп'ютером виробляє особливий стиль мислення, прищеплює певні навички розумових дій:

- вміння планувати структуру дій, необхідних для досягнення заданої мети за допомогою фіксованого набору засобів (знаючи кінцеву мету, визначити послідовність простих більш чи менш стандартних операцій, які до неї ведуть і які доступні);
- вміння алгоритмічно мислити (представити складну дію у вигляді організованої сукупності простих);
- вміння організувати пошук інформації необхідної для розв'язування поставленої задачі. Ці вміння мають загальнокультурну, загальноосвітню та загальнолюдську цінність і необхідні всім.

Комп'ютер усе більше входить до найближчого інтелектуального оточення дитини, впливає на формування навчального середовища та на навчальну діяльність дитини в цілому. Активне залучення комп'ютера в процес виконання творчих робіт дозволяє учням розвивати творчий підхід до навчання та виховувати інтерес до проведення робіт дослідницького характеру, що сприяє комплексному розвитку структури пізнавальної активності.

Література

1. Абасов З.А. Познавательная активность школьников // Советская педагогика. – 1989. – №7. – С. 40–43.
2. Гончарова М.С. Методичні основи розвитку пізнавальної активності у процесі вивчення алгебри і початків аналізу на основі НІТ // Комп'ютерно-орієнтовані системи навчання. – К.: Комп'ютер в школі та сім'ї. – 1998. – С. 50–55.
3. Гринько О.М. Проблеми стимулювання самостійної навчально-пізнавальної активності учнів // Рідна школа. – 1998. – №5. – С. 72–75.
4. Дидактика современной школы: пособие для учителей. / под редакцией Онищука В.А. – К.: Радянська школа, 1987. – 351 с.
5. Жалдак М.І. Комп'ютер на уроках математики: Посібник для вчителів. – К.:Техніка, 1997. – 303 с.
6. Лозовая В.И., Троцко А.В. Познавательная активность как педагогическая проблема. // Советская педагогика. – 1989. – №11. – с. 25–31.

АКТУАЛЬНІСТЬ ВИВЧЕННЯ МОВИ JAVA У КУРСІ “ІНФОРМАТИКА І КОМП’ЮТЕРНА ТЕХНІКА” ПРИ ПІД- ГОТОВЦІ ФАХІВЦІВ СПЕЦІАЛЬНОСТІ “ЕКОНОМІЧНА КІБЕРНЕТИКА”

С.Р. Красильников, І.С. Пятін, О.Ю. Рудик
м. Хмельницький, Технологічний університет Поділля

Характерною рисою сучасного розвитку суспільства є тотальна інформатизація. Важлива роль у цьому процесі припадає на мережі Інтернет та Інтранет.

Інтернет – це сама велика світова комп’ютерна мережа, що поєднує комп’ютери з усім світом. Інтернет є глобальною обчислювальною мережею, вона утворена з регіональних мереж. Її основне призначення – обмін інформацією будь-якого виду між комп’ютерами, підключеними в мережі.

Розглядаючи цю передову технологію у курсі “Інформатика і комп’ютерна техніка”, неможливо не зупинитися на одному із сьгоднішніх прогресивних напрямків роботи з інформацією із використанням мови об’єктно-орієнтованого програмування Java. За своєю структурою мова Java схожа на мову С++ [1]. Однак ця мова, за наявності спеціальної програми (віртуальної машини), здатна працювати на будь-якій платформі. У такий спосіб вирішується проблема різномірності стандартів у різних мережах і системах.

Однією із умов знайомства із всіма можливостями мови Java у вищому навчальному закладі є наявність мереж Інтернет та Інтранет. Така умова, зокрема, створена у Технологічному університеті Поділля (ТУП), де з 1996 року функціонує і постійно розвивається мережа Інтранет. Вона охоплює 7 навчальних корпусів і поєднує біля 300 комп’ютерів. Ця потужна система задіяна для підготовки фахівців різних напрямів і спеціальностей в ТУП, в тому числі і майбутніх бакалаврів з економіки та підприємництва.

Фундаментальна загально-фахова підготовка фахівців такого напрямку розпочинається на I-му курсі дисципліною “Інформатика і комп’ютерна техніка”, вивчення якої відбувається протягом 3 навчальних семестрів у межах 216 годин.

Інваріативна компонента програми вищезгаданого курсу відповідає вимогам нормативних документів, які діють з 1997 року [2]. Варіативна частина оновлюється в зв'язку із постійним стрімким розвитком сучасних інформаційних технологій і змін парадигм програмування.

Так, перегляд програми курсу, з урахуванням перспективи використання Інтранет-технології в навчальному процесі та майбутній професійній діяльності показав доцільність введення навчальної теми “Основи мови об’єктно-орієнтованого програмування Java”, розрахованої на 20 аудиторних годин. Завданням цієї теми є ознайомлення студентів із середовищем мови Java (інструментальними засобами Java Developer’s Kit (JDK), розроблених фірмою Sun Microsystems), синтаксисом мови та технологією створення Java-апплетів. Ефективним засобом осмислення навчального матеріалу теми та методом формування практичних навичок використання Java-технологій, на думку авторів [3], є лабораторно-практичні заняття. Орієнтовний перелік лабораторно-практичних занять може бути таким:

1. Перевірка Java-компілятора і віртуальної Java-машини.
2. Тестування роботи утиліти appletviewer.
3. Створення документації для Java-програм. Тестування роботи утиліти javadoc.
4. Вивчення синтаксису мови Java.
5. Створення Java-апплетів.
6. Використання набору абстрактних властивостей організації вікон (AWT).

Структура і зміст лабораторно-практичного заняття №1 “Перевірка Java-компілятора і віртуальної Java-машини” наведено нижче.

Лабораторна робота №1

Перевірка Java-компілятора і віртуальної Java-машини

Мета: Ознайомитися з компілятором Java (команда javac), інтерпретатором Java (команда java) і технологією створення і запуску Java-додатків.

Матеріально-технічне забезпечення:

Програмний пакет JDK фірми Sun, ЕОМ з процесором Pentium MMX 200, операційна система Windows 9x або NT.

Дидактичне забезпечення: Методичні вказівки до вико-

нання лабораторної роботи.

Порядок виконання роботи.

Java-програми (як додатки, так і аплети) не є автономними і потребують наявності середовища виконання Java. Java-додатки виконуються інтерпретатором (компілятором). Для перевірки інсталяції пакета JDK можна створити маленький Java-додаток.

При створенні Java-додатку необхідно пройти такими етапами:

1. Встановити структуру каталогів для зберігання Java-коду, включаючи головний каталог та підкаталоги.
2. Ввести та зберегти початковий код.
3. Відкомпілювати початковий код.
4. Виконати додаток: передати файлу байтовий код, який було створено в процесі компіляції.

Створення нового Java-проекту.

Структура каталогів додатків

Для збереження проектів створіть новий каталог. Його можна назвати USER і помістити окремо від каталогу JDK; у цьому випадку робочий каталог не потрібно переписувати при установці нової версії пакета JDK. У середині робочого каталогу створіть підкаталог із назвою групи (наприклад, ЕК-11), в якому розташуйте підкаталог JAVA_LAB. Результат виконання описаних дій показано на рис. 1.1.



```
Microsoft(R) Windows 98
(C) Copyright Microsoft Corp 1981-1998.
C:\WINDOWS>cd ..
C:\>cd USEREK-11\JAVA_LAB\
C:\USEREK-11\JAVA_LAB>dir

Том в пристрої C має метку SYSTEM
Серійний номер тома: 3035-11E1
Середня величина блоку: C:\USEREK-11\JAVA_LAB

.                <DIR>          15.02.01  14:53 .
..               <DIR>          15.02.01  14:53 ..
                @  файл          @  файл
                2 файл      448 717 312 байт свободно
C:\USEREK-11\JAVA_LAB>
```

Рис. 1.1. Приклад створення підкаталогу JAVA_LAB для зберігання початкового коду

Ввід та збереження початкового коду

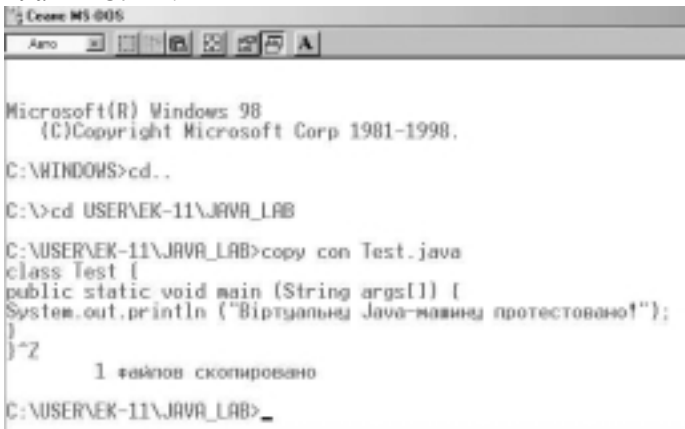
За допомогою будь-якого текстового редактора створіть

файл за назвою Test.java і наберіть у ньому такий текст:

```
class HelloWorld {  
    public static void main(String args[]) {  
        System.out.println("Віртуальну Java-машину протестова-  
но!");  
    }  
}
```

Не забудьте включити розширення java, інакше компілятор не побачить ваш файл. Нехай синтаксис поки вас не цікавить; набирайте текст як є, збережіть його і вийдіть з редактора. Переконайтеся в тім, що текст збережений у виді звичайного ASCII-файлу.

На рис. 1.2 показано результат створення файлу Test.java у режимі "Сеанс MS-DOS" після виконання команди copy con Test.java, набору тексту і закриття файлу натискуванням комбінації клавіш Ctrl+Z.



```
Microsoft(R) Windows 98  
(C)Copyright Microsoft Corp 1981-1998.  
C:\WINDOWS>cd..  
C:\>cd USER\EK-11\JAVA_LAB  
C:\USER\EK-11\JAVA_LAB>copy con Test.java  
class Test {  
    public static void main (String args[]) {  
        System.out.println ("Віртуальну Java-машину протестовано!");  
    }  
}  
^Z  
1 файл(ів) скопировано  
C:\USER\EK-11\JAVA_LAB>_
```

Рис. 1.2. Результат вводу та збереження початкового коду

Запуск Java-додатка в середовищі Windows чи UNIX

Компіляція початкового коду

Якщо ви працюєте в системах UNIX чи Windows, у командному рядку (сеансу DOS) наберіть команду:

```
javac Test.java
```

Після деякої паузи ви повернетеся до запрошення системи.

Результат виконання описаних дій показано на рис. 1.3.

```

C:\WINDOWS>cd..

C:\>cd USER\EK-11\JAVA_LAB

C:\USER\EK-11\JAVA_LAB>copy con Test.java
class Test {
public static void main (String args[]) {
System.out.println ("Віртуальну Java-машину протестовано!");
}
}
1 файл(ів) скопировано

C:\USER\EK-11\JAVA_LAB>javac Test.java

C:\USER\EK-11\JAVA_LAB>

```

Рис. 1.3. Результат виконання команди `javac Test.java`

Далі роздрукуйте зміст підкаталогу `JAVA_LAB` і переконайтеся в тім, що в ньому знаходяться такі файли (рис. 1.4):

HelloWorld.class

HelloWorld.java

```

C:\USER\EK-11\JAVA_LAB>copy con Test.java
class Test {
public static void main (String args[]) {
System.out.println ("Віртуальну Java-машину протестовано!");
}
}
1 файл(ів) скопировано

C:\USER\EK-11\JAVA_LAB>javac Test.java

C:\USER\EK-11\JAVA_LAB>dir

Том в устройстве C имеет метку SYSTEM
Серийный номер тома: 3035-11E8
Содержимое папки C:\USER\EK-11\JAVA_LAB

.                <ПАПКА>          13.02.01  14:53 .
.                <ПАПКА>          13.02.01  14:53 ..
TEST1   JAV          123  13.02.01  16:05 Test.java
TEST1   CLA          466  13.02.01  16:14 Test.class
2 файл(ов)      589 байт
2 папок        435 783 888 байт свободно

C:\USER\EK-11\JAVA_LAB>

```

Рис. 1.4. Зміст підкаталогу `JAVA_LAB`

Якщо виникли помилки, перевірте текст програми `Test.java`.
Якщо з'явилось повідомлення про те, що програма `javac` не

знайдена, те це означає, що ви не задали каталог JAVA/BIN у перемінної PATH. Перевірте інсталяцію пакета JDK (див. додаткову інформацію).

Виконання додатку

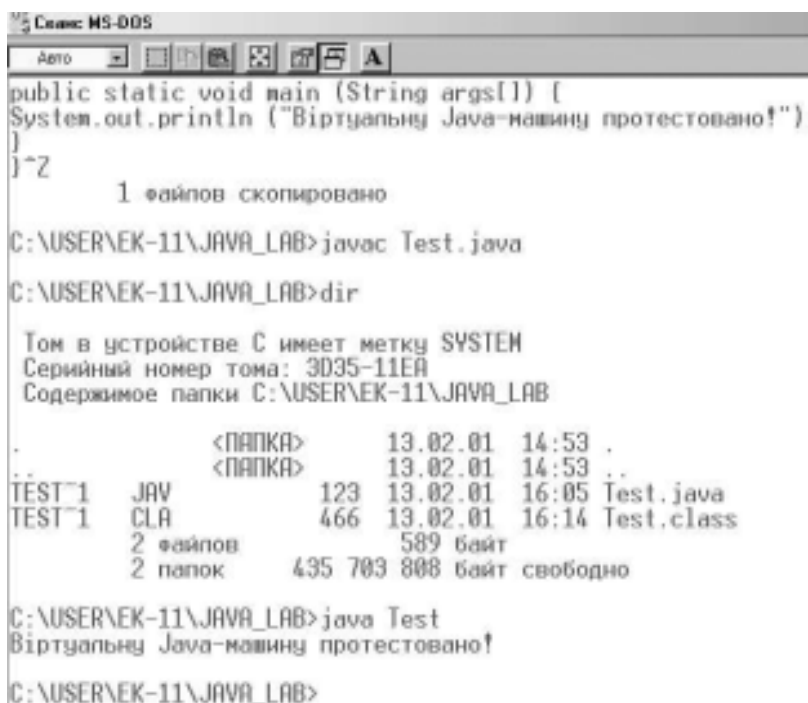
Тепер можна запускати вашу першу Java-програму! У командному рядку наберіть:

Java Test

Ви повинні побачити таку інформацію (рис. 1.5):

Віртуальну Java-машину протестовано!

Якщо це вийшло – прийміть поздоровлення! Ви виконали перший Java-додаток і, що більш важливо, правильно установили пакет **JDK**.



```
Служба MS-DOS
Авто
public static void main (String args[]) {
System.out.println ("Віртуальну Java-машину протестовано!");
}
]~2
1 файлов скопировано
C:\USER\EK-11\JAVA_LAB>javac Test.java
C:\USER\EK-11\JAVA_LAB>dir
Том в устройстве C имеет метку SYSTEM
Серийный номер тома: 3D35-11EA
Содержимое папки C:\USER\EK-11\JAVA_LAB
.                <ПАПКА>          13.02.01  14:53 .
..               <ПАПКА>          13.02.01  14:53 ..
TEST~1  JAV          123  13.02.01  16:05 Test.java
TEST~1  CLA          466  13.02.01  16:14 Test.class
2 файлов          589 байт
2 папок          435 703 808 байт свободно
C:\USER\EK-11\JAVA_LAB>java Test
Віртуальну Java-машину протестовано!
C:\USER\EK-11\JAVA_LAB>
```

Рис. 1.5. Результат виконання програми Test.java

Якщо повідомлення **Віртуальну Java-машину протестовано!** не з'явилося, ви неправильно виконали інсталяцію. Перевірте, чи містить змінна CLASSPATH посилання на поточний робо-

чий каталог (".") і на файл CLASSES.ZIP. Переконавшись в правильному написанні імені файлу: Java розрізняє рядкові і заголовні букви. Якщо все це не допомагає, встановіть пакет знов (див. додаткову інформацію).

Додаткова інформація

Інсталяція для Windows

Для роботи з Java потрібно мати Windows 95 чи Windows NT. В даний момент відсутні версії JDK для Windows 3.1, 3.11 чи Windows for Workgroups.

Інсталяція пакета досить проста, однак потрібно бути знайомим із середовищем Windows і DOS. Для Windows пакет JDK звичайно поширюється як архівований файл, що саморозкривається; ім'я файлу вказує на версію. Щоб інсталювати пакет JDK потрібно:

1. Вибрати каталог для інсталяції. У даному описі мається на увазі інсталяція в каталозі C:\JAVA. Якщо ви вибрали інший кореневий каталог, укажіть відповідний шлях (і диск, якщо потрібно). Наприклад, для установки пакета в каталог E:\TOOLS\JAVA замініте C: на рядок e:\tools у всіх наступних описах.

Увага

Не можна встановлювати пакет JDK поверх попередніх версій, особливо якщо це були бета-версії!

1. перейменуйте каталог JAVA (наприклад, у OLDJAVA). Якщо інсталяція з якихось причин перерветься, ви зможете відразу ж відновити попередню версію. При успішному результаті можна перемістити всі додаткові файли, до приклада документацію, зі старого каталогу в новий, після чого їх можна видалити зі старого каталогу.

2. Якщо ви хочете робити інсталяцію на мережний диск, перевірте, чи маєте ви дозвіл на запис.

3. Скопіюйте дистрибутивний файл JDK у C:\.

4. Розпакуйте виконуваний архів, що саморозкривається.

5. Перевірте створення на диску в кореневому каталозі C:\ наступних каталогів:

C:\JAVA C:\JAVA\CLASSES C:\JAVA\LIB C:\JAVA\BIN
C:\JAVA\DEMO

Порада

У системі Windows NT 4.0 і наступних версій можна пропус-

тити кроки 6, 7, 8 і установити змінну CLASSPATH з вікна властивостей (properties sheet). Перезавантаження не потрібно, однак варто закрити всі сеанси DOS, що повинні використовувати нову перемінну.

6. Додайте рядок C:\JAVA\BIN у перемінну PATH файлу AUTOEXEC.BAT:

set PATH=c:\windows;c:\dos;...;c:\java\bin

7. Встановіть змінну середовища CLASSPATH у файлі AUTOEXEC.BAT:

set CLASSPATH=c:\java\lib\classes.zip

8. Для того щоб змінні середовища набрали сили, перезавантажте комп'ютер.

Таким чином, актуальність вивчення мови Java очевидна і тому доцільно продовжувати працювати над розробкою структури і змісту теми, а в подальшому трансформувати її в окремий курс.

Література:

1. Вебер Д. Технология Java в подлиннике: Пер. с англ. – СПб.: БХВ-Петербург, 2000. – 1104 с.

2. Нормативні програми дисципліни фундаментального циклу освітньо-професійної підготовки бакалаврів з економіки підприємництва / Кол. авт. під загальним керівництвом А.Ф. Павленка. – К.: КНЕУ, 1997. – 237 с.

3. Казанчан А., Мішуков О. Ділова документація у вищій школі. – Херсон: ХВПУ, 1999. – 245 с.

МОЖЛИВОСТІ РЕАЛІЗАЦІЇ МОДУЛЬНО-РЕЙТИНГОВОЇ ТЕХНОЛОГІЇ НАВЧАННЯ ІНФОРМАТИКИ У ВИЩИХ ЗАКЛАДАХ ОСВІТИ ЕКОНОМІЧНОГО ПРОФІЛЮ

Ю.М. Красюк

м. Ірпінь, Академія державної податкової служби України

Сучасне інформаційне суспільство ставить перед вищою освітою актуальне завдання – формування спеціаліста з високою загальною культурою, який спроможний реалізувати власний творчий потенціал у теоретичній та практичній діяльності дослідницького характеру. Такому спеціалісту недостатньо лише володіти великим об'ємом фактологічної інформації та вміти її застосовувати до відомих проблем, він також повинен уміти перспективно використовувати дану інформацію у непередбачуваних ситуаціях. Тому у процесі навчання студенти повинні не тільки дізнаватися про нові факти, але й виявляти творчий підхід до використання наукової та технологічної інформації.

Акцентуючи увагу на тому, хто навчається, та підпорядковуючись таким принципам як системність, структуризація, модульність, динамічність, гнучкість, усвідомлення цілей, обернений зв'язок, активність та наочність, модульно-рейтингова технологія навчання надає можливість кожному студенту досягти необхідних результатів.

Загалом модульно-рейтингову технологію можна визначити як підхід до професійної підготовки майбутнього фахівця, базовими компонентами якого виступають:

- модульна організація навчання;
- методичне та програмне забезпечення аудиторної та самостійної роботи студентів;
- систематичний поетапний контроль навчальної діяльності;
- рейтингова методика оцінки знань;
- суб'єкт-суб'єктивні відносини між викладачем та студентами.

Розглянемо можливості реалізації модульно-рейтингової технології на прикладі курсу “Інформатики та обчислювальної техніки” (далі ІОТ) для економічних спеціальностей вищих за-

кладів освіти.

Діяльність викладача повинна розпочинатися з глибокого та детального вивчення навчальної програми дисципліни та відповідної навчально-методичної документації. Після цього з урахуванням вимог модульного підходу, державних професійних стандартів із предмета, кількості годин, що відведені на вивчення курсу, та технічних можливостей комп'ютерів, якими укомплектовані лабораторії навчальних закладів, розробляється робоча програма курсу та визначається перелік модулів даної дисципліни. Для кожного модуля створюється пакет навчально-методичної документації. До його складу можуть входити:

- комплект лекцій з відповідних тем;
- комплект навчальних елементів по вивченню тем, які вносяться для опрацювання на практичні та лабораторні заняття, а також на самостійне опрацювання;
- перелік тем для проведення студентами наукових досліджень, написання рефератів та статей;
- комплект завдань для проведення підсумкового контролю.

При цьому відбувається перерозподіл пріоритетів у виборі таких форм навчання як лекції, семінари, практичні та лабораторні роботи, самостійна робота студентів та наукові гуртки. Лекції, втрачаючи своє “монопольне становище”, серед інших видів занять, стають здебільшого установчими, оглядовими чи підсумковими. Основним засобом реалізації модульного підходу в навчальному процесі виступає навчальний елемент - автономний навчальний матеріал, який призначений для засвоєння деякої елементарної одиниці знань та умінь і використовується для самонавчання під керівництвом викладача. Для роботи з кожним навчальним елементом створюється відповідний навчально-методичний комплект, який має наступну структуру:

1. Назва та мета вивчення навчального елемента.
2. Обладнання, його характеристики, перелік необхідного програмного забезпечення та додаткові засоби, які необхідно мати при роботі з даним навчальним елементом.
3. Перелік супутніх навчальних елементів та матеріалів (якими навчальними посібниками може користуватися студент, з якими іншими навчальними елементами пов'язаний даний на-

вчальний елемент).

4. Зміст навчального елемента. В цьому розділі розкривається суть навчального матеріалу.

5. Комплект “допомоги” – додатковими методичними відомостями, які надаються при необхідності студентам для успішного оволодіння навчальним матеріалом.

6. Система вправ для самостійного виконання студентами.

7. Перелік тестових завдань для самоконтролю студентів, що включає коди правильних відповідей.

8. Комплект тестових завдань, який міг би використовувати викладач для проведення поточного контролю знань студентів по закінченню вивчення кожного навчального елемента.

Використання навчальних елементів створює сприятливі умови для забезпечення диференціації та індивідуалізації навчання. Отримавши комплект з відповідним навчальним елементом, кожний студент має можливість самостійно працювати над ним (вивчати теоретичні відомості, виконувати систему вправ, здійснювати перевірку власних досягнень) у темпі, який відповідає його індивідуально-психологічним особливостям. Проте, навіть у цій ситуації навчальна діяльність кожного студента потребує керуючого спрямування викладача. Адже однакові комплекти навчальних елементів не створюють навіть приблизно однакових сприятливих умов для навчання всіх студентів, оскільки вони відрізняються між собою за початковим рівнем підготовки зі шкільного курсу “Основи інформатики та обчислювальної техніки” (далі ОІОТ), за темпами оволодіння навчальним матеріалом, здібностями самостійно застосовувати засвоєні знання та вміння і т.ін. Тому для організації ефективної роботи студентів з навчальними елементами викладач повинен виконувати підготовчу роботу, яку можна умовно розділити на два етапи.

Перший етап бажано провести на самому початку вивчення курсу ІОТ. Він полягає у проведенні анкетування, яке б надавало можливість виявити глибину вивчення предмета ОІОТ кожним студентом у навчальних закладах, які він закінчив. Це дає змогу розподілити студентів за типологічними групами по підготовленості до вивчення курсу ІОТ. Відразу необхідно зауважити, що розподіл студентів між типологічними групами може змінюватися після вивчення кожного модуля залежно від виявлених ре-

зультатів навчання студентів. Відмінність між типологічними групами студентів полягатиме лише у різному виді допомоги, яку надаватиме їм викладач.

Другий етап, представлений у формі тестування, проводиться перед вивченням кожного навчального елемента та полягає у виявленні рівня підготовки студента з даного матеріалу. За результатами тестування викладач виявляє рівень допомоги, який необхідно надати кожному студенту в процесі роботи над навчальним елементом.

Попередньо з'ясувавши до якої типологічної групи входить студент та необхідний йому рівень допомоги, викладач доповнює його навчальний елемент необхідними матеріалами з комплексу “допомоги”. При цьому вид допомоги, що надається студентам, залежить від того, до якої типологічної групи вони відносяться, а рівень допомоги від рівня підготовки студентів з даної теми.

По закінченню роботи студента з навчальним елементом обов'язково проводиться підсумкове тестування, під час оцінювання якого необхідно пам'ятати, що його результати завжди потрібно порівнювати, як зазначає Г.І. Щукіна, з результатами попередніх робіт (попереднього тестування). Це дає можливість визначити зміну рівня знань кожного студента, а також встановити характерні помилки та пропуски, які збереглися у нього.

Проте навчальні елементи не можуть виступати єдиним засобом реалізації навчальних цілей. Разом з ними бажано використовувати й інші дидактичні матеріали : опорні конспекти, логічні схеми, комп'ютерні навчаючі програми, роздаткові матеріали, наочні посібники, відео- та аудіоматеріали.

При цьому виділяються наступні етапи заняття, яке побудоване на роботі з навчальними елементами:

Відний інструктаж. На цьому етапі викладач знайомить студентів з метою даного заняття, проводить мотивацію вивчення навчального елемента, який засвоюється на занятті, визначає зв'язки даного елемента з іншими.

Визначення базового рівня підготовки студентів з поточної теми. На цьому етапі студенти виконують тестові завдання, а викладач з'ясовує необхідний рівень допомоги, який повинен бути наданий кожному студенту.

Організація роботи студентів з навчальним елементом. На цьому етапі студенти отримують уявлення про структуру даного навчального елемента, усвідомлюють цілі даного елемента, визначають входні характеристики, знайомляться з умовами перевірки досягнень.

Виконання студентами навчального елемента. На даному етапі студенти засвоюють зміст навчального елемента, виконують завдання та вправи.

Оцінка досягнень та перевірка результатів роботи з навчальними елементами. На цьому етапі перевірки досягнень студенти виконують тестові завдання, самостійно або за допомогою викладача визначають рівень досягнення цілей навчального елемента.

Отже, на груповому занятті студенти проходять два етапи тестування, кожний з яких триває 5-8 хвилин та потребує оперативного оцінювання. В даній ситуації для проведення тестування та визначення результатів найдоцільніше використовувати автоматизовані системи тестування.

Крім того, по закінченню вивчення кожного модуля поряд з підсумковим тестуванням (контрольною роботою) бажано проводити дидактичні або ділові ігри з використанням завдань творчого характеру, під час розв'язування яких студенти мали б можливість застосовувати отримані знання, уміння та навички в нових умовах, а також створювати нову інформацію, пропонуючи оригінальні способи вирішення поставлених перед ними навчальних задач.

В той же час особливої актуальності поряд з проблемою контролю знань набуває проблема адекватної якісної оцінки виявленої студентами активності, творчості, самостійності й систематичності в роботі протягом усього періоду вивчення дисципліни.

В основу рейтингової системи оцінки покладено визначення рейтингу кожного студента з даного предмета. Він підраховується, як сума набраних студентом балів за визначений період і за різнобічну діяльність по опануванню дисципліни. Дана сума виступає в ролі числового показника якості роботи студента в порівнянні з максимально можливою кількістю балів і з успіхами товаришів по навчанню. При цьому кожна кількість балів, що зараховується студенту, визначається як добуток, перший множ-

ник якого – це оцінка, яка отримана під час контролю, другий – коефіцієнт, який відповідає виду виконаної студентом діяльності. Так для курсу ІОТ можливо встановлення наступної таблиці коефіцієнтів:

№	Вид діяльності	Коефіцієнт
1	Поточне тестування	1
2	Захист виконаної практичної та лабораторної роботи	2
3	Захист самостійної роботи, виконаної за навчальним елементом без керівництва викладача	3
4	Реферати, статті, виступи	5
5	Написання контрольної роботи	4
6	Виконання завдань творчого характеру	4
7	Підсумкове тестування в кінці вивчення модуля	4

Крім того, викладачі можуть, як встановлювати заохочувальні бали за неординарне виконання завдань, активну участь у наукових гуртках, студентських конференціях та інше, так і вилучати бали за несвоєчасне виконання завдань, пропуски групових занять без поважних причин і таке ін. Поточна рейтингова оцінка має створити для студента мотиви до постійної, регулярної та творчої праці.

По закінченню вивчення дисципліни підраховується підсумковий рейтинг студента, який служить основою для визначення за встановленою шкалою загальної оцінки з курсу. В цій шкалі бажано передбачити:

1) Суму балів за роботу по вивченню всього курсу (певної частини курсу), яка дає право автоматичного отримання екзаменаційної (залікової) оцінки без складання іспиту (заліку);

2) Мінімальну суму балів для отримання допуску до заліку або іспиту;

3) Суму балів, яка надає право на ліквідацію заборгованості;

4) Суму балів, яка вимагає як мінімум повторного вивчення обов'язкового курсу.

МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ ЛАБОРАТОРНОГО ПРАКТИКУМУ З ОСНОВ ІНФОРМАТИКИ

Б.Б. Круліковський

м. Рівне, Рівненський державний гуманітарний університет

В навчальних планах природничих спеціальностей курс “Інформатика і програмування” викладається на першому та другому роках навчання. При цьому на лабораторний практикум відведена досить обмежена кількість годин (18 на семестр). Зважаючи на те, що велика частка студентів закінчила школи в сільській місцевості і не мала змоги працювати з ПЕОМ, така кількість лабораторних годин, особливо в першому семестрі, явно недостатня.

Для надання студентам можливості самостійно опрацювати лекційний матеріал а також набувати навички практичної роботи з ПЕОМ розроблено і впроваджено в навчальний процес комп’ютерний варіант системи лабораторних робіт. Цикл робіт з MS DOS містить 12 робіт, цикл робіт з NC – 8, цикл робіт з архіваторами – 3. Надлишкова кількість робіт надає викладачу змогу варіювати тематику робіт для кожного студента в залежності від ступеня його шкільної підготовки.

Методичні рекомендації для кожної роботи складаються з трьох частин: теоретичного матеріалу, багатоваріантного практичного завдання та прикладів рейтингових контрольних запитань. Завдяки цьому студенти можуть виконувати кожну роботу не тільки під час аудиторних занять за розкладом, а і під час самостійної роботи в обчислювальному центрі університету в позаурочний час.

Завдяки такій організації навчального процесу студенти при виконанні практичних завдань багатократно звертаються до теоретичної частини матеріалу за допомогою, що ефективно сприяє його засвоєнню. Крім того, необхідне переключення з режиму виконання роботи в режим перегляду теоретичного матеріалу закріплює навички роботи з консоллю. Таким чином підвищується і рівень індивідуалізації навчання.

Досвід використання такої методики протягом чотирьох років однозначно свідчить про її ефективність.

К ВОПРОСУ СОВЕРШЕНСТВОВАНИЯ МЕТОДИКИ ПРЕПОДАВАНИЯ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ СТУДЕНТОВ СТРОИТЕЛЬНЫХ СПЕЦИАЛЬНО- СТЕЙ

Г.И. Кулик

г. Днепропетровск, Приднепровская государственная академия
строительства и архитектуры

В настоящее время все более высокие требования предъявляются к современному специалисту и инженеру-строителю в частности. Сегодня не достаточно усвоить определенный объем специальных знаний, чтобы уверенно чувствовать себя на рынке труда. Особенно это сказывается на карьере сегодняшних выпускников высших учебных заведений. Они оказываются не подготовленными к решению тех производственных проблем, которые им предстоит решать в своей трудовой деятельности. Это, прежде всего, отсутствие какого-либо практического опыта, с одной стороны, и, как правило, очень невысокий уровень навыков работы на компьютере. При все возрастающем интересе нового поколения к компьютеру часто приходится сталкиваться с некомпетентностью в этом вопросе на самом элементарном уровне. Существуют объективные причины для этого:

1. На занятиях у студента в большинстве случаев нет возможности закрепить материал в полном объеме из-за недостаточной оснащенности учебных классов. Не выполняется соотношение Кол. студентов=Кол. компьютеров в аудитории на занятиях.
2. Отсутствие необходимой литературы на абонементах в студенческих библиотеках и высокая стоимость этой литературы в магазине.
3. Очень часто приходится сталкиваться с отсутствием школьной компьютерной подготовки у абитуриентов. И в то же время часть студенческой аудитории имеет достаточно высокую подготовку. Разный уровень подготовки студентов требует индивидуального подхода как при низком уровне, так и в случае высокой подготовленности студента. В противном случае большая часть аудитории теряет интерес к занятиям и

эффективность занятий резко снижается.

Опыт преподавания информатики на факультативных курсах показывает, что растет заинтересованность студентов в изучении информатики, особенно у дипломников, когда они сталкиваются с реальной проблемой трудоустройства.

Требования работодателя включают обязательное наличие навыков работы на компьютере, причем требуются не только начальные навыки пользователя, а и навыки работы с профессиональными пакетами прикладных программ. Таким опытом может похвастаться далеко не каждый. Это приводит будущих выпускников на факультативные курсы, которые за сравнительно короткое время позволяют ликвидировать пробелы в знаниях.

Основными отличиями в работе в студенческой аудитории и на факультативных занятиях являются:

а) небольшая наполняемость факультативных групп и реальная возможность каждого слушателя на протяжении всех занятий работать за персональным компьютером;

б) построение занятий по схеме: выслушал объяснение, понял, реализовал на компьютере;

в) в условиях факультатива есть возможность комплектовать группы с учетом начального уровня компьютерной грамотности;

г) небольшое количество слушателей в группе позволяет подобрать каждому индивидуальное задание с учетом специфики его интересов и уровня подготовки. Работая с задачами, которые близки к реальным, возникающим на практике, студент приобретает необходимый ему практический опыт. Все это в комплексе позволяет за небольшой срок дать необходимый объем знаний будущему специалисту.

Опыт работы на факультативных курсах показал, что некоторые положительные моменты можно реализовать и при работе по общим программам.

Освоение компьютера для инженера не является самоцелью и не целесообразно стараться получить представление о всем многообразии современных программных продуктов и познакомиться с ними поближе. При подготовке инженера строителя необходимо учитывать специфику будущей специальности и сферу возможного применения компьютера молодым специалистом. Это, прежде всего, возможность выполнять расчетную и

графическую часть проекта (курсового, дипломного, реального) за рабочим столом, оборудованном компьютером, составлять необходимую проектно-сметную документацию, пользоваться всеми сервисными возможностями, которые предоставляют пользователям современные программные продукты и компьютерная техника, в том числе и сетевыми технологиями.

Пути повышения эффективности проведения занятий можно искать в двух направлениях: в изменении методики преподавания предмета и в организации проведения занятий.

Для реализации этой идеи необходима логическая завершенность курса:

- максимальное внимание уделять изучению раздела алгоритмизации в рамках курса информатики: это даст возможность пользователю продумать последовательность действий, приводящую от исходных данных к конечному результату и осмысленно использовать компьютер не только в простейших случаях, но и при решении нестандартных задач с помощью современных программных продуктов;
- в большем объеме использовать возможности приложений Windows, в частности технологии внедрения и связывания объектов, что является мощным средством для автоматизации процесса проектирования, если подключить при этом специализированные графические пакеты.

С точки зрения эффективности проведения занятий возможным решением проблемы является:

- 1) деление студенческой группы не на 2 (как принято), а на 3 подгруппы;
- 2) комплектование этих подгрупп с учетом уровня подготовленности студентов, и включение в индивидуальные задания задач повышенной сложности для наиболее подготовленных студентов;
- 3) при комплектовании групп и составлении расписания проведения занятий на вычислительном центре стараться приблизиться к оптимальному соотношению студент/компьютер.

В комплексе эти мероприятия позволят более эффективно использовать имеющиеся ресурсы компьютерной техники и учебное время, отведенное в расписании на изучение предмета.

МЕТОДИКА ИЗУЧЕНИЯ НЕКОТОРЫХ РАЗДЕЛОВ ИНФОРМАТИКИ, ОСНОВАННАЯ НА ИСПОЛЬЗОВАНИИ ОБУЧАЮЩИХ ПРОГРАММ

С.И. Куликов, С.П. Кандзюба
г. Днепропетровск, Украинский государственный химико-
технологический университет

Использование вычислительной техники в учебном процессе открывает широкие возможности для внедрения новых обучающих методик и технологий. На кафедре ВТ и ПМ разработка обучающих программ, на основе имеющихся методических материалов, ведется с 1995 года. Основной целью при создании программ было, с одной стороны, уменьшить у преподавателя долю нетворческой, рутинной работы и, с другой стороны, предоставить студентам дополнительные возможности при изучении предмета.



Рис. 1. Заставка программы «Паскаль за три часа!»

Сценарий, используемый в обучающих программах, содержит те же элементы, что и реальный учебный процесс – изучение теории, разбор примеров и выполнение зачета. Общий подход при создании обучающих программ позволяет нам в рамках данной статьи остановиться только на одной из них – обучающей программе «Паскаль за три часа!» (рис. 1).

Программа «Паскаль за три часа!» содержит набор тем, соответствующих рабочей программе по дисциплине (рис. 2). Каждая тема включает в себя теоретические сведения, подробный разбор одного или нескольких примеров и зачет.

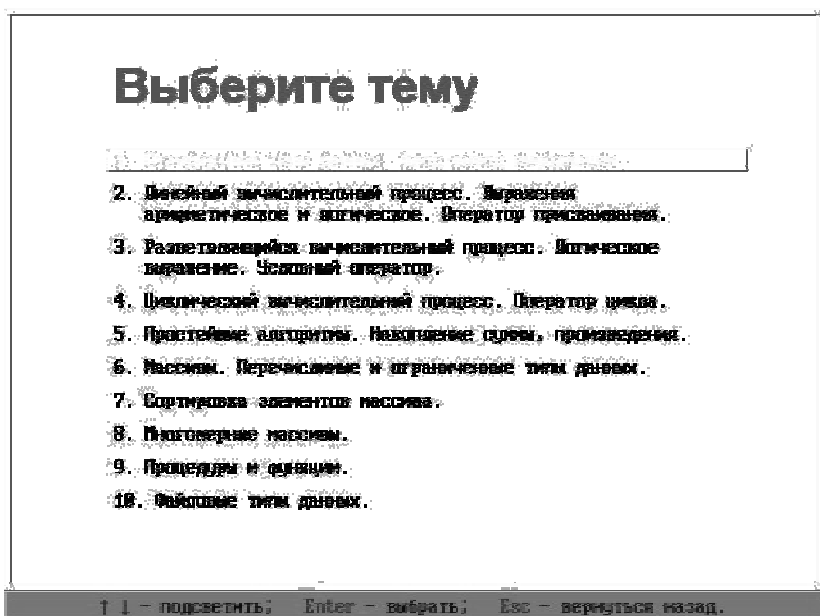


Рис. 2. Набор тем обучающей программы

Теоретическая часть обучающей программы, являющаяся аналогом лекции, содержит только самые необходимые сведения для каждой темы. Как показывает опыт, теоретическая подготовка может и должна выполняться до занятия.

Каждая тема содержит один или несколько примеров, в которых в интерактивном режиме проводится подробный разбор решения соответствующих задач (рис. 3). При демонстрации ис-

пользуются различные эффекты: изменение цвета отдельных символов или частей текста, выделение текста при помощи трехмерных поверхностей, выделение отдельных частей сцен при помощи указателей в виде стрелок, руки и т.д. Кроме того, воспроизведение каждой сцены сопровождается клипом с говорящим человеком и звуковыми эффектами. Все это в целом наряду с тщательно подобранными примерами позволяет добиться быстрого понимания и усвоения учащимися новых знаний.



<p style="text-align: center;">Задача</p> <p>Вычислить: $y = \sqrt{e^{-x} - 2} + \frac{x - \sin x}{1 + x}$ при $x = 1.0$ $z = \ln(e^{x^2} \cdot y)$</p>	<p style="text-align: center;">Введите X</p> <p>1.0 _</p>
<p style="text-align: center;">Программа</p> <pre> PROGRAM LZ2 (INPUT, OUTPUT); VAR X, Y, Z: REAL; BEGIN WRITELN('Введите X '); READLN(X); Y := EXP(1/E) * LN(EXP(X - Z) + (X - SIN(X)) / (1 + X)); Z := ARCTAN(Y) * SQRT(1 - Y * Y); WRITELN(' При X = ', X:3:1, ' Y = ', Y:6:3, ' Z = ', Z:6:3); END. </pre>	<p style="text-align: center;">Переменные</p> <p>X: 1.0 Y: неопр Z: неопр</p>
<p style="text-align: center;">Комментарий</p> <p>В правой части оператора присваивания задано выражение, выполняющее вещественный результат. Операции в нем выполняются в следующем порядке...</p> <div style="display: flex; align-items: center;"> <div style="flex: 1;"> <pre> Y := EXP(1/E * LN(EXP(X - Z) + (X - SIN(X)) / (1 + X))); </pre>  </div>  </div> <p style="text-align: center;">Enter - продолжить; Alt-X - закончить; Esc - вернуться на один шаг.</p>	

Рис. 3. Фрагмент разбора примера

Содержащийся в каждой теме зачет представляет собой полноценную контролирующую подсистему. Интересной особенностью контролирующей подсистемы программы «Паскаль за три часа!» является то, что, выбирая правильные ответы из предлагаемого списка вариантов, учащийся, тем самым, формирует текст будущей программы (рис. 4).

Традиционным можно считать подход, когда контролирующая система предлагает некоторый вопрос и, одновременно, набор вариантов ответа, из которых только один является правиль-

ным. Такой подход реализован в других программах, разработанных на кафедре ВТ и ПМ. Так, например, в контролирующей системе, которая была создана для проведения комплексной контрольной работы в соответствии с требованиями министерства образования Украины, из базы данных случайным образом выбираются по очереди пять вопросов. К каждому вопросу может прилагаться до десяти вариантов ответов, из которых учащийся должен выбрать один правильный (рис. 5).

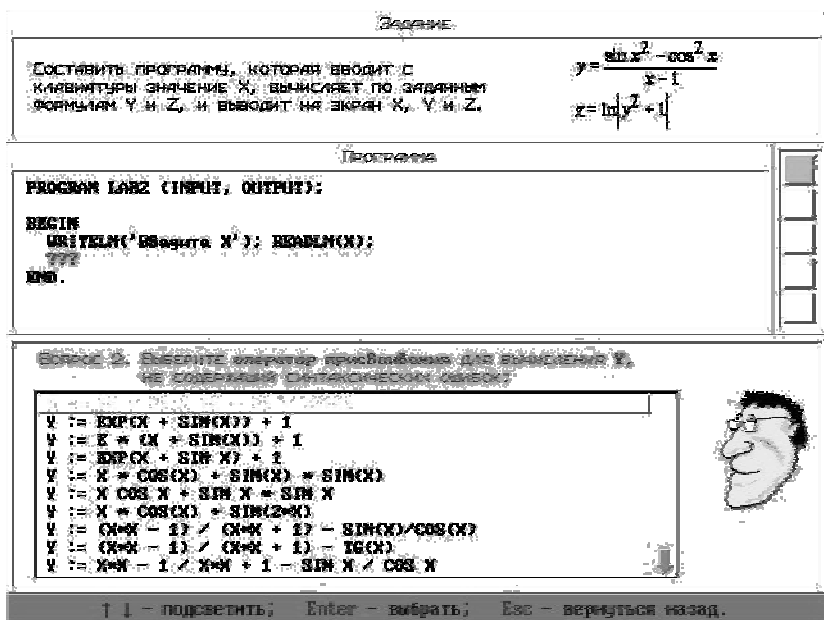


Рис. 4. Фрагмент выполнения зачета

Сценарий, используемый в программе «Паскаль за три часа!», а также в других обучающих программах, разработанных на кафедре ВТ и ПМ для изучения различных разделов информатики, может быть с успехом применен и для изучения других дисциплин. Так, например, совместно с кафедрой неорганической химии УГХТУ были разработаны две обучающие программы, используемые в настоящее время в ходе проведения лабораторных занятий.

Вопрос

Выбрать правильный фрагмент блок-схемы для решения поставленной задачи: провести, упорядочивая по элементам массива $A(10)$ по убыванию.

а)

б)

в)

ПОДСВЕТИТЕ ОТВЕТ, КОТОРЫЙ ВЫ СЧИТАЕТЕ ПРАВИЛЬНЫМ, И НАЖМИТЕ ENTER.

б)

а)

↑ I – подсвечить; Enter – выбрать; Esc – вернуться назад.

Рис. 5. Зачет в стиле «вопрос–ответ»

Использование обучающих программ на протяжении нескольких лет позволило сделать следующие выводы:

1. У студентов появляется возможность самостоятельно осваивать новый материал в удобном для них темпе.
2. Студенты со слабой подготовкой или пропустившие занятия могут изучить материал самостоятельно.
3. У преподавателя появляется больше времени для индивидуальной работы со студентами.
4. Подход, заложенный в обучающих программах, является универсальным и может быть использован при изучении различных дисциплин.

ПСИХОЛОГО-ПЕДАГОГІЧНІ АСПЕКТИ ВИКЛАДАННЯ ІНФОРМАТИКИ У 5–9 КЛАСАХ

О.В. Курочкіна

м. Кривий Ріг, Середня загальноосвітня школа з поглибленим
вивченням фізики, інформатики та історії №69

Крокуючи в нове тисячоліття і захопивши з собою всі досягнення сторіччя, ми вже не мислимо прогрес без комп'ютерів, які впевнено ввійшли в наше життя. Комп'ютер знайшов своє місце в багатьох сферах людської діяльності.

Перед сучасною школою стоїть задача підготовки грамотних користувачів комп'ютерної техніки.

Комп'ютерна грамотність, яку ще називають “другою грамотністю”, передбачає засвоєння знань, умінь та навичок, що дають змогу успішно використовувати комп'ютер при розв'язуванні практичних задач.

У середніх класах для формування основ комп'ютерної грамотності учні мають набути вмінь та навичок програмувати свою діяльність у спілкуванні з комп'ютером.

У нашій школі виділяються три основних етапи навчання:

Перший етап (V–VI класи) – пропедевтичний. На цьому етапі відбувається первісне ознайомлення учнів з комп'ютером, формуються початки інформаційної культури.

Другий етап (VII–IX класи) – курс основної школи (базовий), який забезпечує обов'язковий загальноосвітній мінімум підготовки школярів з інформатики.

Третій етап (X–XI класи) – продовження освіти в галузі інформатики як профільного навчання.

Інформатика порівняно з іншими навчальними дисциплінами молода наука, що стрімко розвивається і набуває дедалі більшої популярності. Що може дати інформатика дітям у 5–9 класах, що привносять комп'ютери в процес навчання, чим вони є – метою навчання чи засобом? Спробуємо розібратися в тому, які перспективи відкриває ця наука перед учителем і його учнями.

Питання про мету чи засіб виникає не випадково, оскільки з моменту появи інформатики в школі жодний із вказаних аспектів не став визначальним. З одного боку, комп'ютер використову-

ється для вдосконалення процесу навчання – це використання навчальних програм, програм для демонстрації наочних моделей, для тестування та контролю знань учнів. З другого боку, комп'ютер сам стає об'єктом вивчення. Учні ознайомлюються з апаратною частиною, навчаються роботі з програмним забезпеченням. Крім того, вивчення мови програмування, можливість створювати власні програми, що керують роботою комп'ютера і розкривають його можливості, відкривають нові перспективи у вивченні інформатики.

Кожний з перерахованих аспектів, безумовно, є важливим. Але спочатку спробуємо визначити, які можливості надає комп'ютер як технічний засіб навчання.

Ставши на позицію вчителя, ми можемо сказати, що навчання за допомогою комп'ютера розширює можливості подання інформації. Будь-яке навчання пов'язано зі сприйняттям, аналізом та накопиченням інформації. Наочність відіграє велику роль у зацікавленості й розумінні матеріалу. Тому традиційні плакати, стенди, роздаткові картки можуть бути замінені яскравою комп'ютерною графікою. Якщо учень працює з програмою індивідуально, то підключається ще й чуттєвий аспект отримання інформації. Дитина сама керує швидкістю подання інформації за можливостями програми, її обсягом і глибиною.

Крім того, з використанням комп'ютера з'являється можливість зробити уроки динамічнішими. Комп'ютер дає можливість перетворити урок або його частину на захоплюючу гру, що значно підвищує інтерес до предмету. Відчуття гри знімає напругу і нервозність у дітей, особливо в підлітковому віці.

Зацікавленість і доступність подання матеріалу значно підвищують можливості учня щодо вивчення інформатики. Щоб зрозуміти “як це зроблено”, він готовий опрацювати не лише ігровий матеріал, але й значно складніші розділи теорії. Крім самих уроків інформатики, на мотивацію дитини впливають телебачення, сучасні журнали, оточуюче життя, у якому, як ми вже зазначили, позиції комп'ютера стають дедалі стійкішими. Цікаво, що на підвищений інтерес дітей до комп'ютера має вплив і проблема “батьків та дітей”. Оскільки більшість батьків не володіють комп'ютерною грамотністю і відносяться до комп'ютерів з обережністю та цікавістю, то діти через обізнаність у

комп'ютерній техніці отримують можливість довести свою “дорослість”.

Великою перевагою у використанні навчальних програм на уроках інформатики є індивідуалізація навчання. Вона передбачає індивідуальний темп подання та сприйняття матеріалу, який притаманний кожній дитині окремо, тоді як у класичному варіанті вчитель вимушений орієнтуватися на “середнього учня”.

Крім прикладних програм, комп'ютер у курсі інформатики використовується також для навчання основам алгоритмізації і програмування. Навчання програмуванню розвиває в дитини абстрактне мислення. Важливим моментом алгоритмізації є вміння узагальнювати, виділяти закономірності й бачити сутність задачі.

Працюючи з програмою, налагоджуючи її, учень миттєво бачить реакцію комп'ютера на свій алгоритм. Помилки логіки в даному випадку більш очевидні, ніж помилки, виявлені вчителем у домашньому завданні.

Інформатика у 5–9 класах, як система знань, має певний рівень узагальненості, інтегративності, базується на досить великому обсязі попередніх знань й передбачає наявність в учня абстрактного мислення, яке проявляється в дитини з 12–13-річного віку.

Враховуючи вищесказане, можна впевнено стверджувати, що основний курс інформатики треба запроваджувати саме з 5 класу, одразу після закінчення початкової школи, бо саме тоді починається розвиток абстрактного мислення. В той же час не викликає сумнівів доцільність навчання основ інформаційної культури в початковій школі, як це пропонується у проекті державного стандарту з інформатики.

КОНЦЕПЦІЯ КОМП'ЮТЕРИЗАЦІЇ АГРАРНОГО УНІВЕРСИТЕТУ

Є.А. Лавров, А.Т. Ашерев, Г.А. Смоляров
м. Суми, Сумський державний аграрний університет

Рішення задач підготовки висококваліфікованих спеціалістів зумовлює використання в навчальних закладах сучасної комп'ютерної техніки та інформаційних технологій.

В Сумському аграрному університеті розвивається тенденція комп'ютеризації закладу. Для цього є належна матеріальна база, професорсько-викладацький персонал, методичні розробки та посібники.

1. Напрямки комп'ютеризації університету.

В університеті сформовано систему безперервної комп'ютерної підготовки, навколо якої вирости самостійні напрями використання ЕОМ в науково-дослідній роботі і в управлінні університетом. Вже сьогодні працюють елементи ієрархічної АСУ такі, як “Бухгалтерія”, “Навчальний процес”, “Бібліотека”.

В останній рік зазнали розвитку комп'ютеризація 2 нових галузей навчального процесу – довузівської і післявузівської підготовки. Одним з суттєвих факторів, що впливають на привабливість університету і забезпечують високі конкурси, є комп'ютерна профорієнтація і тестування школярів. Ця робота проходить на протязі всього року і дозволяє забезпечити формування континенту майбутніх студентів. Сьогоднішні школярі мають реальний доступ в комп'ютерні класи, в Internet, можуть одержати робітничу професію, пройти систему профорієнтаційних тестів. Все це суттєво підвищує довіру до університету і залучає на навчання значну кількість майбутніх студентів.

Роботу з абітурієнтами облегує комп'ютерна база “Абітурієнт”. Плануємо створення і ведення бази даних випускників, налагоджена система підготовки спеціалістів АПК з питань комп'ютеризації виробництва.

Таким чином, ректорат визначив всебічну комп'ютеризацію пріоритетним напрямком розвитку університету.

2. Комп'ютеризація навчального процесу.

Наші дослідження показали, що 71% вакансій, які пропонуються нашим випускникам, вимагають вільного володіння комп'ютером, а 79% студентів пов'язують успішність майбутньої кар'єри з умінням роботи на базі інформаційних технологій.

Виходячи з важливості питання, ректоратом сформовано Раду з комп'ютеризації університету. Нею розроблено концепцію комп'ютеризації навчального процесу. Вихідними положеннями комп'ютеризації навчання є:

- всебічна комп'ютеризація навчальної діяльності;
- індивідуалізація і активізація навчання;
- використання при навчанні задач професійної діяльності;
- оволодіння новими інформаційними технологіями.

Для забезпечення цих вихідних положень розроблено систему принципів комп'ютеризації навчального процесу. Концепцією визначено, що персональні комп'ютери і засоби телекомунікацій використовуються:

- як об'єкт вивчення;
- для виконання обчислень;
- для моделювання процесів і явищ;
- при вивченні управління технологічними процесами, машинами і механізмами;
- при вивченні автоматизації проектування;
- як засіб аналізу і статистичного опрацювання даних про явища та процеси в АПК;
- як засіб навчання.

Безперервна комп'ютерна підготовка передбачає такі рівні:

- базова комп'ютерна підготовка, що здійснюється кафедрою кібернетики та інформатики на молодших курсах;
- прикладна комп'ютерна підготовка, що здійснюється спеціальними кафедрами з 1-го по 5 курс;
- поглиблена комп'ютерна підготовка, яка проводиться за бажанням студентів.

Широке впровадження елементів поглибленої комп'ютерної підготовки стало інструментом деканатів по підвищенню престижності деяких спеціальностей.

Комп'ютерна підготовка базується на наступних принципах.

Базова:

- кожен студент повинен мати навички комп'ютерного доку-

ментообігу;

- кожен студент повинен мати навички роботи в Internet;
- одне робоче місце – один студент;
- середній час роботи одного студента за комп'ютером у навчальний день не менше 1 години;
- навчання на реальних задачах, підготовлених усіма кафедрами по кожній спеціальності;
- самостійне вивчення окремих розділів із застосуванням навчальних систем.

Прикладна:

- немає дисципліни без навчальних і контролюючих програм;
- немає курсового проекту без застосування ПК;
- немає дипломного проекту без застосування ПК.

Поглиблена:

- кожен студент може поглибити свої знання в області інформатики і комп'ютерних технологій і, здавши відповідні іспити, одержати робітничу професію;
- кожен студент повинен брати участь у науково-дослідній роботі, використовуючи знання в області інформатики і ІТ.

Сьогодні в університеті поставлені задачі:

- підвищити активізацію навчання за рахунок застосування в кожній дисципліні нових технічних засобів навчання, заснованих на комп'ютерній техніці;
- забезпечити поступовий перехід до індивідуалізації навчання за рахунок застосування в кожній дисципліні навчальних програм і комп'ютерних технологій навчання, а також за рахунок підвищення кількості комп'ютерної техніки, доступної для роботи студентів, і підвищення середнього часу щоденної роботи їх за комп'ютером.

В університеті регламентовано мінімальний контактний дисплейний час за весь період навчання, який становить для економічних спеціальностей 350-400 годин, для технічних спеціальностей 250 годин, для технологічних спеціальностей 200 годин.

Для забезпечення цих показників розроблено і впроваджено 3-рівневу систему розподілу комп'ютерів за принципом: кафедральний комп'ютер – факультетський комп'ютерний клас – загальноуніверситетський комп'ютерний клас.

Ми підтримуємось позиції концентрування комп'ютерів в

спеціалізовані класи. При цьому кожен факультет має один або декілька комп'ютерних класів.

Сьогодні в університеті 375 комп'ютерів і 14 комп'ютерних класів.

Така структура технічного забезпечення і методичні розробки кафедр дозволили факультетам впровадити систему безперервної комп'ютерної підготовки, яка виходить на нормативні значення контактного дисплейного часу і забезпечує застосування комп'ютерів в кожному семестрі, в кожній дисципліні.

3. Комп'ютеризація управління навчальним процесом.

Для управління навчальним процесом розроблена і поетапно впроваджується АСУ "Навчальний процес". Вже на протязі 5 семестрів працюють підсистеми, що забезпечують функції автоматизованої розробки розкладу і оперативного диспетчерування.

Передбачається на протязі року впровадити мережевий варіант системи і запустити підсистеми, пов'язані з контролем відвідування занять і успішності студентів, а також веденням студентських справ, починаючи з приймальної комісії і деканатів і закінчуючи факультетом післявузівського навчання.

4. Пріоритетні напрямки подальшої роботи.

4.1. Дистанційне навчання.

Сьогодні ми проводимо інтенсивні консультації з провідними лабораторіями, що займаються цією проблемою. Це Санкт-Петербурзький інженерно-економічний університет, МЕСІ, ХП, Міжнародний університет фінансів. Проведено експериментальні сеанси дистанційного навчання.

Сьогодні проблема лежить не стільки в сфері технічного забезпечення, скільки в наповненні курсів високоякісними електронними підручниками найвищого гатунку.

4.2. Продовження робіт по створенні адміністративної мережі університету і впровадженню підсистем АСУ, а також роботи в галузі інформаційного забезпечення дорадництва і впровадження геоінформаційних систем.

4.3. Важливим напрямком ми вважаємо представлення університету в мережі Інтернет. Задача в створенні і обслуговуванні сучасного розгалуженого сайту університету, який розкривав би всі сфери, в яких працює університет (навчання, наука, дорадництво та ін.)

ПОКРАЩЕННЯ ПРАКТИЧНОГО СПРЯМУВАННЯ ФАХОВОЇ ПІДГОТОВКИ УЧИТЕЛЯ ІНФОРМАТИКИ

В.В. Лапінський, В.М. Дем'яненко
м. Київ, Національний педагогічний університет ім. М.П. Драго-
манова

Завдяки цілеспрямованим зусиллям, які за останні роки були докладені по впровадженню комп'ютерної техніки у загальноосвітні навчальні заклади, матеріальна база шкільної інформатики у столиці Україні наблизилась до стану, який забезпечує навчання інформатики на сучасному рівні. Зокрема, завдяки реалізації цільової програми Київської міської державної адміністрації по оснащенню ЗОШ сучасною обчислювальною технікою, досягнуто практично 100% забезпечення київських шкіл комп'ютерами, у т.ч. – 70% шкіл оснащено найсучаснішими засобами обчислювальної техніки (ЗОТ) [1]. Оснащення шкіл найсучаснішою технікою є позитивним фактором, але і “висвічує” певні недоліки підготовки вчителя. Зокрема, оскільки у штатному розкладі і у бюджеті ЗОШ не передбачено можливостей залучення фахівців, кваліфікація яких достатня для повноцінного обслуговування і ремонту ЗОТ, частину цих функцій переймає на себе власне учитель інформатики. Таке перепризначення функцій не є новим для школи і специфічним для учителя інформатики – учитель фізики, не зважаючи на те, що у школі, як правило, є лаборант, самостійно готує демонстраційний експеримент, проводить лекційні демонстрації, ремонтує прилади. Учителі хімії, біології, трудового навчання також готують об'єкти навчально-трудової діяльності. Доступність такого виду фахової діяльності для вчителів навчальних дисциплін природничого циклу і трудового навчання визначається традиційним спрямуванням навчальних планів педагогічних вищих навчальних закладів, у яких передбачено достатній обсяг годин, спрямованих на формування практичних умінь та навичок.

При створенні програм фахової підготовки учителя інформатики зазначений напрям майбутньої професійної діяльності не завжди враховується [2, 3]. Також при цьому виявляється специфіка навчальної дисципліни – якщо, наприклад, матеріальне за-

безпечення процесу навчання фізики змінюється досить повільно і навички роботи з апаратурою, надбані 10–15 років тому, є актуальними для учителя фізики і зараз, то учитель інформатики, який зупинився у своєму фаховому розвитку на 1–2 роки, безнадійно відстав і не може ефективно організовувати навчальний процес, оскільки не зможе підготувати програмно-апаратне забезпечення для проведення навчального процесу.

Аналіз навчальної діяльності учителя інформатики, який було проведено із використанням стандартних методик педагогічного дослідження – спостереження, аудит, анкетування, показав, що кращі вчителі м. Києва володіють методами роботи із сучасними програмним забезпеченням, деякі з них уміють виконувати діагностику і нескладний ремонт апаратного забезпечення. Але значна частина вчителів, як показують результати опитування, не мають достатньої практичної і теоретичної підготовки для виконання дій по діагностиці обладнання, налагодженню апаратного і програмного забезпечення. Одночасно з цим, результати опитування учителів та аналізу навчальної діяльності свідчать про те, що ситуації, які потребують виконання таких дій, виникають майже щодня.

До задач, які постають перед учителем інформатики при підготовці до навчального процесу, за свідченням учителів, належать:

- встановлення програмного забезпечення (операційної системи, програмних засобів загального призначення, педагогічного програмного забезпечення) його налагодження та супровід;
- встановлення і діагностування апаратного забезпечення, у т.ч. засобів, які забезпечують роботу локальної мережі і пристроїв уведення/виведення та встановлення відповідних драйверів пристроїв;
- створення дидактичних матеріалів – відеоматеріалів, паперових копій, логічних структур на зовнішніх запам'ятовуючих пристроях та у локальній мережі, заготовок електронних документів (електронних таблиць, баз даних тощо).

Ефективне вирішення визначених навчальних задач можливе тільки за умови створення у майбутнього вчителя умінь та до-

статньо стійких навичок роботи з апаратним і програмним забезпеченням. У вузі такі навчальні задачі розв'язуються, як правило, створенням системи лабораторно-практичних занять, які можуть проводитись як у формі лабораторного практикуму, включеного у систему занять певної навчальної дисципліни, так і у формі спеціалізованого лабораторного практикуму, який має більш вузьку, практичну спрямованість і певну відособленість.

На кафедрі інформатики НПУ ім. М.П. Драгоманова силами викладачів та навчально-допоміжного персоналу створено, і на протязі останніх трьох навчальних років проводиться спеціалізований лабораторний практикум, предметне наповнення якого спрямовано на вирішення задач покращення практичної значимості результатів навчання студентів спеціальностей “Фізика і інформатика”, “Математика і інформатика”, майбутніх учителів інформатики.

Визначення змісту практикуму було виконане з урахуванням результатів аналізу навчальної діяльності учителя, тенденцій розвитку апаратного і програмного забезпечення та базуючись на принципах, сформульованих у [3]. Визначено такі основні напрямки навчання: вивчення основ організації та практичного застосування операційних систем; вивчення засобів і способів конфігурування програмного забезпечення; вивчення способів оптимізації програмної та апаратної конфігурації обчислювальної системи; практичне ознайомлення з можливостями сучасних програмно-апаратних засобів обміну даними. Ці напрямки навчання реалізуються у лабораторному практикумі, який проводиться з використанням бригадної організаційної форми навчання, яку обрано з економічних міркувань, оскільки було недоцільно придбати таку кількість обладнання, яка потрібна була б для фронтального проведення робіт практикуму.

У результаті виконання робіт спеціального лабораторного практикуму у студентів формуються уміння і навички встановлення програмного забезпечення, його налагодження та супровід; встановлення і діагностування апаратного забезпечення, у т.ч. засобів, які забезпечують роботу локальної мережі і пристроїв уведення/виведення та встановлення відповідних драйверів пристроїв; створення дидактичних матеріалів – відео матеріалів, паперових копій, логічних структур на зовнішніх за-

пам'ятовуючих пристроях та у локальній мережі, заготовок електронних документів.

Структура, змістове наповнення та задачі спеціального лабораторного практикуму (назви лабораторних робіт, описи їх змісту та навчальні завдання) були піддані експертній оцінці шляхом анкетування вчителів (опитано 27 учителів інформатики та викладачів вузів), результати якої подано у таблиці.

№	Чи потрібні учителю практичні уміння, які формуються при проведенні практикуму (Так – Ні)	Частка позитивних відповідей, % (кількість позитивних відповідей)	Ефективність навчальних завдань – середня експертна оцінка за 5-бальною шкалою
1	Підготовка накопичувачів на жорстких магнітних дисках до роботи	100 (27)	4.3
2	Конфігурування ПЕОМ (за-собами SETUP BIOS)	81 (22)	4.2
3	Встановлення операційної системи	100 (27)	
4	Налагодження локальної мережі	81 (22)	
5	Встановлення програмного забезпечення загального призначення	100 (27)	5
6	Встановлення педагогічного програмного забезпечення	100 (27)	5
7	Діагностування апаратного забезпечення	59 (16)	3.8
8	Ремонт шляхом заміни складових апаратного забезпечення (клавіатура, монітор, миша, системний блок тощо)	55.5 (15)	3.5
9	Ремонт шляхом заміни модулів (відео адаптер, мультикарта, процесор, модулі пам'яті, накопичувачі тощо)	33 (9)	3.1

Отже, перспективність проведення спеціалізованого лабораторного практикуму із визначеним вище змістовим наповненням і задачами підтверджується як аналізом навчальної діяльності фахівця-випускника педагогічного вузу, так і експертними оцінками учителів.

Література

1. Інтернет: Погляд у майбутнє (Матеріали із засідання круглого столу, Головне управління освіти і науки Київської міської державної адміністрації) // Комп'ютер у школі та сім'ї. – 2001. – №1. – С. 3–12.
2. Беспалько В.П. и др. Системно-методическое обеспечение учебно-воспитательного процесса подготовки специалиста / Учебно-методическое пособие. – М.: Высшая школа, 1989. – 278 с.
3. Наказ №285 від 31.07.98 Міністерства освіти України: “Про порядок розробки складових нормативного та навчально-методичного забезпечення підготовки фахівців з вищою освітою”. – Додаток №1, м. Київ, 1998 р. – 114 с.

К ВОПРОСУ О ВЫБОРЕ ПЕРВОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ

В.Л. Малорян

г. Одесса, Южно-Украинский государственный педагогический университет

Центральная роль изучения языка программирования в каждом серьезном курсе информатики сегодня, как и вчера, не вызывает сомнений. Это касается как средних, так и высших учебных заведений, образуя, таким образом, стержень проблемы ответственности изучения информатики в средней и высшей школе.

Можно выделить три главные цели, преследуемые курсом программирования:

- 1) овладение основами алгоритмизации и формирование алгоритмического, операционального стиля мышления
- 2) овладение навыками осознанного использования и управления компьютерной системой
- 3) расширение возможностей и придание гибкости используемым прикладным программным системам.

Разные языки программирования в различной степени пригодны для решения этих задач. Так, не все языки поддерживают декларативную, процедурную парадигму. Для языков, например, CASE-технологий, более характерен дескриптивный подход. Но значит ли это, что их изучение не развивает специфического, конструктивного стиля мышления? Конечно, нет. И если говорить о деятельностном подходе к изучению компьютера, то он присущ системам на CASE-основе даже в большей степени.

Задача овладения компьютерной системой тесно связывалась с изучением языка программирования в недавнем прошлом, когда отсутствовало достаточное количество прикладных программ, «хороших и разных». Сегодня эта цель скорее и проще достигается изучением набора стандартных программ, существующих для платформ всех типов. Однако, все чаще приходится сталкиваться с ситуацией, когда требуемый прикладной инструмент с особыми свойствами можно быстрее запрограммировать при помощи RAD-системы, чем найти его в необъятных просторах имен и версий. Возникающая альтернатива вечна, как мир.

Но пусть она будет! Похожая ситуация и с реализацией третьей цели.

Итак, изучение языка программирования по-прежнему необходимо. Что же изучают сегодня? Фактически, сегодня первым языком, изучаемым в средней школе, являются различные диалекты Бейсика, Паскаля, Лого или школьный алгоритмический язык. Значительно реже это С, С++, Пролог или другие языки. Безусловно, это связано с изначально учебным предназначением языков первой группы.

В высшей школе студенты компьютерных, информатических специальностей изучают С++, Дельфи, Джаву. В последнее время некоторые американские университеты внедряют Джаву, как первый язык программирования. Некоторые Европейские и Российские [3] университеты – Оберон.

Следует отметить, что существующее состояние дел формировалось в значительной мере полустихийно под сильным давлением технических специалистов и фирм, без учета психологических, возрастных и культурных особенностей студентов. Этот процесс также сопровождался смещением основных целей изучения при выборе языка. Даже сегодняшний переход к визуальным и компонентно-деятельностным системам программирования [1, 2] не решает всех проблем.

Чего стоит только пресловутая англоязычная ориентация лексики языков программирования. Или только на операционные системы от Microsoft. А ведь сегодня мировое сообщество с интересом поддерживает Linux. Да и применяемые системы совсем не учебные и никак не учитывают уровень обучаемых, их психологию. Пора серьезно подумать о внедрении кросс-систем [5] или визуальных посредников [2]. Наконец, разработать систему, поддерживающую национальную лексику и не уступающую по другим параметрам.

В рамках же сложившихся практик, следует учитывать отмеченные особенности, как при выборе языка, уровень которого должен соответствовать уровню студента, его психологическим особенностям, так и при выделении подмножеств языка для стадийного обучения. Механизмы стандартных библиотек, модулей или пакетов существенно облегчают этот процесс, но не всегда достаточны.

С другой стороны, при планировании учебного процесса мало внимания уделяется оценке сложности языка, в том числе, его грамматики (синтаксиса). Хотя достаточно давно известны методы метрического анализа алгоритмов и языков программирования [4]. Недавно была сделана еще одна интересная попытка подобной оценки, исходя исключительно из формального описания грамматики языка [3]. Результаты, полученные С. Свердловым из Вологодского педагогического университета, могли бы стать сенсацией, но они недостаточно известны в академических кругах. Вопреки рекламе фирм, сложна Джава

Необходимо продолжать исследование указанных проблем, в том числе, с использованием специальных психологических тестов, привлекая новейшие языки и системы программирования, даже не слишком широко известные или только создаваемые.

ЛИТЕРАТУРА

1. Малорян В.Л., Варбанец С.В. Компонентно-ориентированный подход к изучению курса программирования в высших учебных заведениях. / Зб. статей Другої всеукраїнської конференції молодих науковців «Комп'ютерне моделювання та інформаційні технології в природничих науках», м. Кривий Ріг, 2-5 травня 2000 р.
2. Копаєв О.В. Вплив сучасних інформаційних технологій на вивчення основ алгоритмізації в середній школі. / Комп'ютер у школі та сім'ї. – №2. – 2000. – С. 24-27.
3. Свердлов С. Арифметика синтаксиса. / PCWeek/RE, №42-43, 1998.
4. Холстед М.Х. Начала науки о программах. – М.: Фин. и стат., 1981.
5. V.L. Maloryan, D.A. Ostroukhov "Compilation-Emulation System O'Pascal." / Proceedings of Symposium on Human Interaction with Complex Systems, NC A&T SU, Greensboro, NC, 1994. – p.p. 9-18.

МЕТОДИЧНИЙ ПРОЕКТ «СИСТЕМА ЛЕКЦІЙНО-ПРАКТИЧНИХ ЗАНЯТЬ ПО ВИВЧЕННЮ ЕЛЕКТРОННИХ ТАБЛИЦЬ MICROSOFT EXCEL» В ЗАГАЛЬНО-ОСВІТНЬОМУ НАВЧАЛЬНОМУ ЗАКЛАДІ

О.М. Марченко¹, М.С. Панченко²

¹ м. Рівне, Рівненський міський економіко-правовий ліцей

² м. Рівне, Рівненський державний гуманітарний університет

Послання Президента України до Верховної Ради України «Україна: поступ у ХХІ століття. Стратегія економічного та соціального розвитку на 2000-2004 роки» та Закон України «Про Концепцію Національної програми інформатизації» визначають метою інформатизації закладу освіти створення навчального середовища, структура і складові якого сприяють досягненню цілей сучасного навчально-виховного процесу.

Поступова, але неухильна зміна освітньої парадигми, збагачення змісту освіти новими процесуальними вміннями, пов'язаними із впровадженням нових інформаційних технологій навчання, задача розвитку здібностей оперування інформацією, акцент на індивідуалізацію освітніх програм викликають ряд якісних змін в психолого-педагогічному плані роботи загально-освітніх навчальних закладів. Ці зміни, зокрема, характеризуються переходом від учіння як функції запам'ятовування до учіння як процесу розумового розвитку, від чисто асоціативної, статичної моделі знань до динамічно структурованих систем розумових дій, від зовнішньої мотивації учіння до внутрішньої морально-вольової регуляції. Особливої актуальності дані психологічні новоутворення набувають в навчально-виховному процесі старших класів.

Чітко усвідомлюючи, що сучасна українська школа повинна забезпечити засвоєння поколіннями, що формуються, вищих здобутків світової науки, культури, надбань новітніх інформаційних технологій, методичні пошуки спрямовуємо, головним чином, на побудову комплексної системи вивчення найбільш актуальних на даний час розділів курсу інформатики, серед яких особливої уваги, на нашу думку, варті: «Інформаційні технології. Інформаційна система. Технологія розв'язування за-

дач з використанням засобів нових інформаційних технологій навчання». В зв'язку з бурхливим розвитком та впровадженням в практику повсякденного життя нових інформаційних технологій вважаємо задачею першочергової важливості забезпечення сучасного науково-методичного підходу до висвітлення ключових розділів курсу «Основи інформатики», таких як «Інформаційна система, її структура. Програмне забезпечення. Операційна система, її складові. Видавничі системи. Інтелектуальні бази даних. Опрацювання даних. Розв'язування навчально-дослідницьких задач».

Саме на розв'язання окресленого кола задач націлено методичний проект «Система лекційно-практичних занять по вивченню електронних таблиць Microsoft Excel». Microsoft Excel – відносно нова для учнівського загалу нашого регіону прикладна програма з відомого Microsoft Office для Windows 95/98 – набору потужних прикладних програм, спеціально розроблених для спільного використання.

Враховуючи спеціалізацію РМЕПЛ (здобуття основ економічних знань), слід підкреслити, що дана програма використовується для ефективної обробки числових даних, дозволяє використовувати електронні таблиці, професійні діаграми і карти для аналізу і подання результатів практичної діяльності, містить доступні засоби для створення фінансових звітів і прогнозів діяльності підприємств, допомагає у пошуках оптимальних рішень на основі аналізу декількох варіантів. Потужний апарат арифметичних та тригонометричних функцій Excel дозволяє здійснити реальні кроки до інтеграції у вивченні предметів природничо-математичного циклу, про що свідчать наведені в посібнику варіанти диференційованих за складністю вправ з дидактичних матеріалів для проведення практичних занять; фінансові та статистичні функції формують в учнів реальні навички розв'язання задач в ході вивчення спецкурсів «Основи економічної теорії», «Економіка підприємства», «Економічний аналіз», «Основи статистики». Тим самим формується тверде підґрунтя для мотивованого усвідомлення напрямків майбутньої професійної діяльності, свідомий аналіз варіантів продовження здобуття освіти, і, нарешті, стає можливою чітка диференціація життєвих пріоритетів та свідомо фільтрація інформаційних по-

токів.

Запропонована система побудови лекційно-практичних занять в своїй логічній основі спирається на продуктивну ідею укрупнення дидактичних одиниць, запропоновану наприкінці 80-х років П.М. Ерднісвим. Укрупнена дидактична одиниця (базова програмна тема, доповнена системою чітко диференційованих за складністю практичних робіт) – це, образно кажучи, унікальна клітина навчального процесу [3, с. 6], що складається з логічно взаємопов'язаних елементів, які разом складають спільну інформаційну картину. Укрупнена дидактична одиниця володіє якостями системності та цілісності, чітко визначеними часовими межами в контексті загального навчального плану, а також, що дуже важливо і вже перевірено на практиці, властивостями швидкого та продуктивного відновлення в пам'яті. При цьому формуються умови для виявлення фундаментальних закономірностей процесів мислення, які разом сприяють ефективній оптимізації навчального процесу. Спільна психологічна сутність усіх ланок системи укрупнених дидактичних одиниць полягає в тому, що в загальній картині процесу розвитку системних знань попередні та наступні в часі ланки мають достатню кількість спільних носіїв інформації, починаючи з нижчих рівнів. Логічну структуру та методи організації матеріалу системи лекційно-практичних занять з Microsoft Excel, на нашу думку, можна розглядати як спробу побудови теорії викладання передбачених навчальним планом розділів вивчення прикладних програм, яка реалізує на практиці закономірності економного та високоякісного навчання шляхом укрупнення навчальної інформації. Лейтмотивом уроку, побудованого по системі укрупнення дидактичних одиниць є чітке правило [3, с. 16]: не відстрочена в часі реалізація набутих теоретичних знань, не механічне повторення, а дійове перетворення фрагменту теорії в результаті практичних дій, що відбувається негайно, на цьому ж уроці, націлене на пізнання об'єкту теорії в його розвитку. Таким чином, якщо запропонована система набуття знань працює завдяки укрупненим одиницям інформації, то тим самим створюються кращі умови для виникнення системної якості знань.

Розуміючи, що система знань має властивість ієрархії, а для кожного рівня ієрархії системоутворюючий фактор різний, вва-

жаємо, що кожна дидактична одиниця навчального матеріалу повинна давати цілісну картину методу обробки інформації. При цьому система організації творчої діяльності учнів на уроці підпорядкована чіткій ієрархії засобів подання, опрацювання, закріплення та самостійної роботи учнів, а саме:

1. Блок входу – повідомлення теми, місця блоку в темі і курсі, завдання, актуалізація знань учнів через опитування попереднього матеріалу, первинна мотивація теми уроку.

2. Блок мінімізації – проблемно-зв'язний виклад мінімуму матеріалу, введення в інформаційно-понятійне поле.

3. Блок первинного контролю – контроль за засвоєнням нових понять.

4. Блок цільової настанови – завдання для самостійної творчої роботи, робота з трансформованими завданнями.

5. Блок самостійно-пошуковий – різні форми самостійної роботи учнів, творчі та наукові роботи.

6. Блок систематизації та узагальнення – визначення науково-світоглядних цілей теми, роль і місце навчального матеріалу в системі нових інформаційних технологій.

7. Блок закріплення (творчий) – закріплення матеріалу шляхом вирішення нетрадиційних завдань. Саме цей блок найбільш наочно демонструє систему взаємопов'язаних підходів до навчання в рамках реалізації ідеї укрупнення дидактичних одиниць: спільне і одночасне вивчення взаємопов'язаних дій та операцій, забезпечення процесів створення та розв'язання задач, аналіз взаємопереходів в трансформованих завданнях, перетворення структури вправи, що створює умови для протиставлення вихідного і перетвореного завдання, виявлення складної природи інформаційних процесів, досягнення системності знань.

Факторами, що забезпечують високу якість укрупненого пізнання виступають загальний графічний образ, спільність символіки, єдність у використанні програмних засобів: діалогових вікон, контекстних меню, загального дизайну вікон прикладних програм.

Таким чином, маємо усі підстави для практичної реалізації основних методичних принципів блочної системи укрупнених дидактичних одиниць: навчання основам інформатики є системою, яка включає в себе взаємопов'язані між собою мету, зав-

дання, зміст, методи, засоби та відповідні організаційні форми реалізації, контролю результатів та їх корекції. Навчання, виховання, розвиток здійснюються у їх взаємозв'язку. Навчальний процес спрямований на формування особистості, побудову рейтингу позитивних пріоритетів, найбільш повну самореалізацію завдяки сформованим навичкам раціонального добору та конструктивного використання вірно дозованої інформації, що підпорядкована свідомому контролю.

В основі навчально-виховного процесу лежить усвідомлена, вмотивована, спрямована на досягнення кінцевого результату діяльність учня. Зміна принципу «мене вчать» на «я вчусь». Самостійна, організаторська, дослідницька та контролююча діяльність учня по здобуванню, а не засвоєнню знань, поданих у готовому вигляді.

Методологічними основами проекту є: поетапне формування вмінь та навичок як засіб розумового розвитку, диференціація та індивідуалізація навчання, перспективна мотивація діяльності школяра через опору на попередньо засвоєні знання, навички та вміння, ретельна розробка та адекватний вибір дидактичного матеріалу, створення оптимальних умов інтелектуального розвитку та позитивного психологічного клімату. Учитель при цьому виступає організатором навчально-пізнавальної діяльності учня, а не передавачем готової інформації. Матеріал викладається як єдина система знань, методів діяльності та їх творчого використання.

При практичній апробації даного методичного проекту ми послідовно впроваджували зміну відношення до помилок учнів; учень не помиляється, а шукає. Завданням вчителя при цьому є корегування результатів самостійної роботи учня, створення сприятливого психологічного клімату з метою підтримання інтересу до навчання, створення дидактичного комплексу, який може одночасно виступати і джерелом інформації, і організатором діяльності учнів з метою пошуку закономірностей та встановлення причинно-наслідкових зв'язків.

Безперечно, основним наслідком впровадження в життя блочної системи укрупнених дидактичних одиниць є формування цілісної системи знань учнів. Адже, як зазначав К.Д. Ушинський, розум – це добре організована система знань. Тому головним напрямом впровадження розвиваючого навчання,

яке прямо орієнтоване на врахування закономірностей розумового розвитку дитини, є організація набутих знань в цілісну систему. І. Кант визначав систему знань як єдність різноманітних знань, об'єднаних однією ідеєю. Це лаконічне означення відображає сутність створюваного в процесі навчання інформаційного всесвіту старшокласника, в якому, в ідеалі, повинні панувати порядок, організація, цілісність.

Розглядаючи практичну реалізацію блочної системи укрупнених дидактичних одиниць в загальному контексті системно-розвиваючого навчання, зазначимо, що вивченню прикладного програмного забезпечення в курсі «Основи інформатики» при стандартному розподілі годин календарного планування відводиться 34 лекційних і практичних годин. З них на ознайомлення з роботою електронних таблиць Microsoft Excel в Рівненському міському економіко-правовому ліцеї заплановано 20 лекційних і практичних годин. Уся сумарна кількість годин забезпечує початкове опанування методики і практики роботи з електронними таблицями. Нижче наведено планування навчальних годин можна розглядати як орієнтовне і при наявності резерву часу перелік навчальних тем та лекційно-практичних годин може бути збільшеним.

№ п/п	Тема уроку	Кількість теоретичних годин	Кількість практичних годин
1.	Перший крок знайомства з Microsoft Excel. Призначення та функції електронних таблиць. Створення робочої книги Microsoft Excel. Введення та редагування даних. Побудова формул для обчислень.	1	1
2.	Практична робота № 1.		1
3.	Другий крок знайомства з Microsoft Excel. Деталізація устрою робочої книги. Інтерпретація даних, що вводяться. Техніка введення тексту.	1	1
4.	Продовження знайомства з Microsoft	1	1

	Excel: аналіз даних з допомогою формул, поняття про оператори та посилання, переміщення і копіювання формул і посилань. Спрощення роботи із посиланнями за допомогою імен.		
5.	Практична робота № 2.		1
6.	Спрощення формул з допомогою функцій робочого аркуша. Робота з Майстром функцій. Управління обчисленнями.	1	1
7.	Практична робота № 3.		1
8.	Поняття про діаграму як засіб графічного подання даних робочого аркуша. Робота з Майстром діаграм. Створення графічних об'єктів на робочих аркушах і діаграмах.	1	1
9.	Практична робота №4.		1
10.	Робота з масивами. Застосування формул масивів. Поняття про масив констант.	1	1
11.	Використання списків для організації даних. Поняття про списки як бази даних. Ведення записів з допомогою форми даних. Обробка списків з допомогою сортування і фільтрації. Поняття про комплексні критерії.	1	1
12.	Практична робота № 5.		1
13.	Підсумковий урок узагальнення набутих знань. Приклади застосування Microsoft Excel для ведення бухгалтерських розрахунків та планування економіки.	1	

Безперечно, наведений варіант планування є орієнтовним, але, на нашу думку, охоплює основні теоретичні розділи та напрямки практичного використання електронних таблиць Microsoft Excel. В даному проекті розподілу навчального навантаження закладене підґрунтя для реалізації блочної системи укрупнених дидактичних одиниць, яка своєю кінцевою метою має:

- 1) уміння аналізувати різноманітні процеси і явища;
- 2) планувати структуру дій, спрямованих на досягнення по-

ставленої мети за допомогою фіксованого набору засобів;

3) алгоритмічно мислити, тобто подавати складні дії у вигляді певної послідовності простіших;

4) будувати інформаційні структури для опису об'єктів та інформаційних систем;

5) правильно, чітко і однозначно формулювати думки;

6) організовувати пошук інформації, яка потрібна для розв'язання поставленої задачі.

Таким чином, можна вести мову про завершений дидактичний цикл навчання як компонент навчального процесу, що об'єднує в собі можливості учня, досвід та знання вчителя, змістовний та процесуальний аспекти навчання.

Література

1. Основи нових інформаційних технологій навчання: Посібник для вчителів / Авт. кол.; за ред. Ю.І. Машбиця / Інститут психології ім. Г.С. Костюка АПН України. – К.: ІЗМН, 1997. – 264 с.

2. Марченко О.М., Лотюк Ю.Г. Робота з електронними таблицями Excel в Microsoft Office 97. Навчально-методичний посібник. – Рівне, 1999. – 136 с.

3. Эрдниев П.М., Эрдниев Б.П. Укрупнение дидактических единиц в обучении математике: Кн. для учителя. – М.: Просвещение, 1986. – 255 с.

ИСПОЛЬЗОВАНИЕ CGI+PERL ДЛЯ СОЗДАНИЯ ПРОСТЕЙШЕЙ БАЗЫ ДАННЫХ

А.А. Мясищев

г. Хмельницкий, Технологический университет Подолья

Стремительный рост популярности Internet – это, прежде всего, заслуга World Wide Web. После появления в 1993 году первого графического Web-браузера сеть Internet стала развиваться семимильными шагами. Тогда количество узлов Internet удваивалось каждые 20 месяцев, а сейчас – каждые 12 месяцев. Количество внутренних сетей – intranet – увеличивается еще быстрее. С небывалой скоростью стали развиваться Web-технологии. Сейчас Web-сервера предоставляют компьютерам-клиентам уже не просто статические Web-странички, а динамические с постоянно изменяющейся информацией, с возможностью интерактивного взаимодействия с серверами баз данных через Web-сервера, используя технологию CGI (Интерфейс общего шлюза).

С целью ознакомления студентов с современными Web-технологиями рассмотрим лабораторную работу по созданию пакета программ, позволяющего формировать простые базы данных, дополнять, распечатывать их в алфавитном порядке, редактировать и удалять записи. Пакет программ должен содержать общее меню по выполняемым действиям с установленным счетчиком его посещения. Он должен также оперировать с графическими объектами (например, фотографии сотрудников). Все программы, включая счетчик посещений, должны быть написаны на языке Perl с использованием CGI. Вызов программы счетчика посещений должен выполняться из главного меню с использованием технологии SSI – server side includes. После изучения представленных ниже текстов программ студенту необходимо разработать подобный пакет программ для БД, заданной преподавателем. Например, БД может включать в себя поля:

1. Название изделия
2. Время выпуска
3. Краткая характеристика изделия
4. Фотография изделия

В качестве изделия могут рассматриваться автомобили, самолеты, суда и др.

Эта лабораторная работа может быть рекомендована студентам для изучения, например, курса «Основы компьютерных сетей» для специальностей «Экономическая кибернетика», «Информационные системы в менеджменте», «Информационные технологии проектирования» и др. Может также рассматриваться как факультативное занятие в курсе «Информатика и компьютерная техника» для углубленного изучения раздела, посвященного сети Internet. Лабораторная работа довольно объемная и в Технологическом университете Подолья студентами выполняется за 8 часов. Ниже представлены рабочие тексты кодов на HTML и Perl с включенными в них комментариями.

Тексты HTML документов для формирования форм:

index_13.shtml

```
<html> <head> <title>Menu</title> </head>
<body> <b>
<a href="http://ism.tup.km.ua/~ismm25/form1_13.html">
1.Добавить запись в базу данных </a><br>
<a href="http://ism.tup.km.ua/ismm25/cgi-
bin/form2_13.pl">
2.Распечатать базу данных </a><br>
<a href="http://ism.tup.km.ua/~ismm25/form3_13.html">
3.Найти запись по Ф.И.О. </a><br>
<a href="http://ism.tup.km.ua/~ismm25/form4_13.html">
4.Удалить запись </a><br>
<a href="http://ism.tup.km.ua/~ismm25/form5_13.html">
5.Редактировать запись</a>
</b> <!--#include virtual="/ismm25/cgi-bin/count.pl"-->
</body> </html>
```

form1_13.html

```
<html> head> <title>form1_13</title> </head> <body>
<form method=post action="/ismm25/cgi-bin/form1_13.pl">
<br><b>Введите Ваше имя:</b>
<br><input name="user_name" value="" size=20>
<br><b>Введите номер Вашего телефона:</b>
<br><input name="phone" value="" size=10>
<br><b>Введите Ваш адрес и краткую характеристику:</b>
<br><textarea name="address" rows=10 cols=60>
</textarea>
```



```
<br><b>Введите имя фото лат.буквами:</b>
<br><input name="img" value="nofoto.jpg" size=20>
<br><input type="submit" value="Зарегистрировать">
</form> </body> </html>
```

form3_l3.html

```
<html> <head> <title>form3_l3</title> </head> <body>
<form method=post action="/ismm25/cgi-bin/form3_l3.pl">
<b>Введите нужную фамилию:</b><br>
<input name="user_name" value="" size=30>
<input type="submit" value="Послать!">
</form> </body> </html>
```

form4_l3.html

```
<html> <head> <title>form4_l3</title> </head> <body>
<form method=post action="/ismm25/cgi-bin/form4_l3.pl">
<b>Введите фамилию, запись по которой будет удале-
на:</b><br>
<input name="user_name" value="" size=30>
<input type="submit" value="Послать!">
</form> </body> </html>
```

form5_l3.html

```
<html> <head> <title>form5_l3</title> </head> <body>
<form method=post action="/ismm25/cgi-bin/form5_l3.pl">
<b>Введите фамилию, запись по которой будет отредактиро-
вана:</b><br>
<input name="user_name" value="" size=30>
<input type="submit" value="Послать!"> </form> </body>
</html>
```

Тексты программ на языке Perl:

form1_l3.pl

```
#!/usr/bin/perl
#
# Этот скрипт предназначен для создания базы данных и ее
# дополнения
# Загружаем модуль CGI
use CGI qw/:standard/;
use strict;
# Присваиваем локальным переменным значения, переданные
# формой
my $user_name=param('user_name');
my $phone=param('phone');
```

```

my $address=param('address');
my $img=param('img');
my $len;
# Определяем длину введенного поля user_name
$len=length($user_name);
# Заменяем символ новой строки на пробел и удаляем
# символ возврата
# Это необходимо, если ввод данных проводится через форму
# с дескриптором <TEXTAREA>
$address=~s/\n/ /g;
$address=~s/\r//g;
# Если len больше 0 добавляем запись в файл
if ( $len > 0 )
{
open (FIL,">>names.txt");
# При записи в качестве разделителей используем :::
print FIL
$user_name,":::",$phone,":::",$address,":::",$img,"\n";
close(FIL);
}
# Формируем HTML страничку
print header(-charset=>'Windows-1251'),
start_html('form1_l3.pl'),
a({href=>'http://ism.tup.km.ua/~ismm25/index_l3.shtml'},"
Возврат в меню"),
hr, end_html;

```

form2_l3.pl

```

#!/usr/bin/perl
# Этот скрипт выводит все имена, находящиеся в базе дан-
ных
# Перед выводом производит сортировку базы данных
use strict;
use CGI qw /:standard/;
my ($user_name,$user,@file,@newfile);
# Чтение базы данных в массив @file, сортировка этого
массива,
# запись отсортированных строк из массива @newfile
# в файл БД names.txt
open(OLD,"names.txt");
@file=<OLD>;
close(OLD);
@newfile=sort(@file);
open(NEW,">names.txt");
print NEW @newfile;
close(NEW);
# Формирование документа HTML

```

```

print header(-charset=>'Windows-1251'),
start_html('form2_l3.pl'), "<pre>";
open (FIL,"names.txt");
while (<FIL>)
{
($user_name)=split(':::');
$user=$user_name;
# Замена пробелов в Ф.И.О. на символ "_" , т.е. например
# Иванов И.И. будет представлен как Иванов_И.И..Если это
# не # сделать, методом GET будет передано "Иванов", т.е.
# без имени # и отчества. Это может привести к
# некорректной работе программы, если фамилия встречается
# несколько раз с различными именем и отчеством.
$user_name=~s/ /_/g;
print
a({href=>"http://ism.tup.km.ua/ismm25/cgibin/forma_l3.pl?
name=$user_name"},
$user), "\n";
}
close(FIL);
print "</pre>",
a({href=>"http://ism.tup.km.ua/~ismm25/index_l3.shtml"}, "
Возврат в меню"), end_html;

```

form3_l3.pl

```

#!/usr/bin/perl
# Скрипт предназначен для поиска записи в базе данных по
# введеному имени. Причем имя может быть не полным.
#
use strict;
use CGI qw /:standard/;
my $user_name_form=param('user_name');
my $user_name_form_len=length($user_name_form);
my ($user_name,$phone,$address,$img,$user);
print header(-charset=>'Windows-1251'),
start_html('form3_l3.pl');
# Если имя введено в форму, происходит обращение к базе
# данных
if ($user_name_form_len != 0)
{
open(FIL,"<names.txt");
# Чтение всех записей из базы данных до обнаружения конца
# файла
while(<FIL>)
{
($user_name,$phone,$address,$img)=split(':::');
# Сохранение в переменной $user_name первых символов в

```

```

# количестве, равному числу символов, введенных в форме
# поиска. Запись их в $user.
$user=substr($user_name,0,$user_name_form_len);
# Если введенное значение в форму совпадает с $user,
# сформировать HTML документ с полным выводом в браузер
# найденной записи
if($user_name_form eq $user)
{
print qq[<center><IMG
SRC='http://ism.tup.km.ua/~ismm25/pics/$img'
HEIGHT=100></center>],
"<br>",
hr(),
"<u>Имя:</u><br>", $user_name,
hr(),
"<u>Номер телефона:</u><br>", $phone,
hr(),
"<u>Адрес и краткая характеристика:</u><br>", $address,
hr();
}
}
close(FIL);
}
print qq[<a
href='http://ism.tup.km.ua/~ismm25/index_l3.shtml'>Возврат
в
меню</a>],
end_html;

```

form4_l3.pl

```

#!/usr/bin/perl
# Скрипт предназначен для удаления записи по имени
use strict;
use CGI qw /:standard/;
my $user_name1=param('user_name');
my $len_user_name1=length($user_name1);
my ($user_name,$phone,$address,$img);
my $i=0;
# Если имя введено в форму, происходит обращение к базе
# данных
if ($len_user_name1 != 0)
{
open(FILUP,"names.txt");
open(FILOUT,">names1.txt");
while(<FILUP>)
{
($user_name,$phone,$address,$img)=split(':::');

```

```

# Если имя, введенное в форму, соответствует имени записи
БД,
# данная запись не записывается в новый файл names1.txt
БД.
if ($user_name1 ne $user_name)
{
$i=1;
print FILOUT
$user_name, "::::", $phone, "::::", $address, "::::", $img;
}
}
close(FILUP);
close(FILOUT);
# Удаляется старая БД (файл names.txt)
unlink("names.txt");
# Переименовывается новая БД (файл names1.txt) в файл
старой
# БД (names.txt)
rename("names1.txt", "names.txt");
}
#
print
header(-charset=>'Windows-1251'),
start_html('form4_l3.pl');
if ( $i == 1 ) { print "Запись не найдена<br>"; }
print
qq[<a
href='http://ism.tup.km.ua/~ismm25/index_l3.shtml'>Возвра
т в
меню</a>],
end_html;

```

form5_l3.pl

```

#!/usr/bin/perl
# Скрипт предназначен для формирования формы
# редактирования записи, удаления из БД редактируемой
записи
# и передачи новых данных скрипту для формирования БД с
# внесенными изменениями
use strict;
use CGI qw /:standard/;
my $user_name1=param('user_name');
my ($user_name, $phone, $address, $img);
# Открывается файл БД и формируется документ HTML
open(FIL, "names.txt");
print header(-charset=>'Windows1251'),
start_html('form5_l3.pl');

```

```

while(<FIL>)
{
# Если имя записи БД совпадает с введенным, формируется
# форма для редактирования записи
($user_name,$phone,$address,$img)=split(':::');
if($user_name1 eq $user_name)
{
print "<IMG SRC='http://ism.tup.km.ua/~ismm25/pics/$img'
HEIGHT=100>";
print qq[<form method=post action="/ismm25/cgi-
bin/form6_l3.pl">];
# Тип hidden не высвечивает поле user_name, но позволяет
# передать его значение программе form6_l3.pl
print "<input type=hidden name='user_name'
value='$user_name'>";
print '<b>Телефон:</b><br>';
print "<input name='phone' value='$phone'
size=10>","<br>";
print '<b>Адрес и характеристика:</b><br>';
print "<input name='address' value='$address'
size=30>","<br>";
print '<b>Фотография:</b><br>';
print "<input name='img' value='$img' size=20>","<br>";
print "Обязательно нажми на кнопку, иначе запись будет
утеряна!";
print '<br><input type="submit" value="Ввести!">';
print '</form>';
}
}
close(FIL);
# Считывается файл БД names.txt и удаляется из него
# редактируемая запись
open(FILUP,"names.txt");
open(FILOUT,">names2.txt");
while(<FILUP>)
{
($user_name,$phone,$address,$img)=split(':::');
# Если имя, введенное в форму, соответствует имени записи
# БД, данная запись не записывается в новый файл
# names2.txt БД.
if ($user_name1 ne $user_name)
{
print FILOUT
$user_name,":::",$phone,":::",$address,":::",$img;
}
}
close(FILUP);
close(FILOUT);

```

```

# Удаляется старая БД (файл names.txt)
unlink("names.txt");
# Переименовывается новая БД (файл names1.txt) в файл
# старой БД (names.txt)
rename("names2.txt", "names.txt");
print end_html;

```

form6_13.pl

```

#!/usr/bin/perl
# Скрипт предназначен для внесения изменений в БД после
# редактирования записи
use strict;
use CGI qw /:standard/;
my $user_name=param('user_name');
my $img=param('img');
my $address=param('address');
my $phone=param('phone');
# Добавление в файл БД отредактированной записи
open (FIL, ">>names.txt");
# При записи в качестве разделителей используем :::
print FIL
$user_name, ":", $phone, ":", $address, ":", $img, "\n";
close(FIL);
# Формирование документа HTML
print
header(-charset=>'Windows1251'),
start_html('form6_13.pl'),
a({href=>'http://ism.tup.km.ua/~ismm25/index_13.shtml'},
"Возврат в меню"), end_html;

```

forma_13.pl

```

#!/usr/bin/perl
# Скрипт предназначен для распечатки параметров записи по
# конкретной Ф.И.О.
use strict;
use CGI qw /:standard/;
my $name=param('name');
my ($user_name, $phone, $address, $img);
# Замена символа "_" на пробел для преобразования
# переданного значения параметром 'name' в действительное
# значение Ф.И.О.
$name=~s/_/ /g;
# Формирование документа HTML
print header(-charset=>'Windows-1251'),
start_html('forma_13.pl');
# Чтение файла с распечаткой параметров записи, для кото-
рой

```

```

$user_name
# совпадает с переданным значением Ф.И.О.
open(FIL1, "names.txt");
while(<FIL1>)
{
($user_name, $phone, $address, $img)=split(':::');
if($name eq $user_name)
{
print qq[<center><IMG
SRC='http://ism.tup.km.ua/~ismm25/pics/$img'
HEIGHT=100></center>],
"<br>",
hr(),
"<u>Имя:</u><br>", $user_name,
hr(),
"<u>Номер телефона:</u><br>", $phone,
hr(),
"<u>Адрес и краткая характеристика:</u><br>", $address,
hr();
}
}
close(FIL1);
print qq[<a
href="http://ism.tup.km.ua/~ismm25/index_l3.shtml">Возврат
меню</a>],
end_html;

```

Текст программы счетчика посещений

```

#!/usr/bin/perl
open (FIL1, "<count.dat");
$count=<FIL1>;
close(FIL1);
$count=$count+1;
open (FIL2, ">count.dat");
print FIL2 $count;
close(FIL2);
print "Content-Type:text/html\n\n";
print "<br><br>", "Страницу посетили ";
print $count;
print " раз";

```

В этом же каталоге должен первоначально существовать файл count.dat с содержимым "0".

Ознакомиться с работой представленного здесь пакета программ можно по адресу <http://ism.tup.km.ua/~ismm25/index.shtml>.

ПРО ДЕЯКІ АСПЕКТИ ВИКОРИСТАННЯ СИСТЕМ ВІЗУАЛЬНОГО ПРОГРАМУВАННЯ В ОСВІТІ

С.М. Прийма

м. Мелітополь, Мелітопольський державний педагогічний
університет

Початок ХХІ століття знаменується зміною філософії всієї освіти, її реформуванням, методологічною переорієнтацією процесу навчання. Це зумовило новий підхід до розуміння предмету інформатики, його місця та ролі в шкільній освіті. Адже саме ці питання на протязі півтора десятка років, з моменту введення інформатики в школу, продовжують залишатися дискусійними.

Аналіз відповідних нормативно-правових документів державного рівня, напрямків розвитку освіти України, новітніх загальноосвітніх тенденцій в галузі освіти, існуючого вітчизняного і світового практичного та науково-педагогічного досвіду, сучасних поглядів вчених і вчителів-практиків з обов'язковим врахуванням реального соціально-економічного стану держави і суспільства [1], синтез обґрунтованої концепції розвитку інформатики, як освітньої галузі, дозволив вирізнити пріоритетний – світоглядний – напрямок вивчення базового курсу інформатики. Цей напрямок пов'язаний із формуванням уявлення про системно-інформаційний підхід до аналізу оточуючого середовища, про роль інформації в управлінні, про загальні закономірності інформаційних процесів у системах різної природи. Доцільність та актуальність цього напрямку не викликають сумніву. Адже саме поняття інформації дозволяє переосмислити роль інформатики та інформаційних процесів в розвитку природи і суспільства, загальнонаукове значення інформаційного підходу як методу наукового пізнання, і тим самим визначити інформатику не як науку про “навколокомп'ютерну” діяльність, а як фундаментальну наукову дисципліну [2]. Звісно значимість даного напрямку вимагає ретельного підходу до вивчення питання, наповнення відповідною системою завдань та вправ, методичною літературою цього розділу [3].

Але не вщухають дискусії навколо частки двох інших розділів базового курсу програми ОІОТ. Аналіз навчальних програм, що реалізують національні освітні стандарти в різні

проміжки часу, дає підставу зробити висновок, що в курсі шкільної інформатики провідна роль належить вивченню програмного забезпечення обчислювальних машин, а час, відведений на теми алгоритмізації та програмування, постійно зменшується. Прибічники “користувачького” напрямку, що пов’язаний із формуванням комп’ютерної грамотності, підготовкою школяра до практичної діяльності в умовах широкого використання інформаційних технологій [4], проголошують ідею відмови від вивчення основ програмування у школі. Багато хто висловлює тезу відносно того, що (на відміну від початку 80-х рр.) у сучасних умовах розвинутого прикладного програмного забезпечення вивчення програмування втратило своє значення як засіб підготовки школярів до праці, фахової діяльності [2]. Не можна не погодитися з деякими аргументами цього підходу, але вивчення основ алгоритмізації та програмування, що лежить в основі третього напрямку, пов’язано з цілим рядом вмінь і навичок загальноінтелектуального характеру, розвитку мислення школярів, формування багатьох прийомів розумової діяльності [2]. Та й сама “комп’ютерна грамотність”, з позиції сучасного розуміння мети та цінності шкільної освіти, вже не може розглядатися як пріоритетна задача, головний сенс вивчення інформатики в школі [2].

На нашу думку, саме в умовах даної дискусії, свідками якої (а іноді і активними учасниками) ми стали на протязі останніх років, певним компромісом є застосування в системі освіти так званих середовищ візуального програмування. Саме такі системи швидкого створення додатків в певній мірі поєднали позитивні риси обох напрямків, наповнюючи новим змістом розділ алгоритмізації та програмування, дозволили по новому оцінити знання інформаційних технологій.

Слід зауважити, що метою даної публікації є викреслення питання вивчення загальних принципів середовища візуального програмування та методики їх викладання, а не зосередження уваги на конкретній системі швидкого створення додатків. Надалі ми будемо висловлювати певні конкретизації для уточнення деяких питань. В цьому випадку ми будемо мати на увазі середовище візуального програмування Delphi, яке базується на об’єктно-орієнтованій мові програмування Object Pascal, що вит-

римує класичну мовну лінію Н. Вірта та має ряд переваг перед середовищами Visual Basic та Visual C++ в системі освіти. Саме в плані розподілу сучасних інструментів розробки на “мова-середовище” актуальність першої складової знижується. Це пояснюється тим, що при збереженні особливостей синтаксису в цілому йде вирівнювання функціональних можливостей систем програмування і розмова, в більшості випадків, йде про середовище, а не про мову програмування. Загальновідомо, що успіх того чи іншого засобу визначається не тільки вишуканістю його синтаксичних можливостей, а й оптимальною відповідністю потребам певної сфери застосування, в нашому випадку в галузі освіти. Але ми ще раз наголошуємо, що розкриті нижче принципи функціонування та методика викладання можуть бути з успіхом застосовані для інших систем швидкого створення додатків.

В чому ж полягають вищезгадані переваги та чим так цікаві системи візуального програмування для використання в освіті?

По-перше, при вивченні систем візуального програмування в повній мірі реалізується принцип доступності [5], що полягає у виділенні рівнів навчання. Як буде показано далі, матеріал можна нормувати (дозувати), викладати порціями (блоками), що впливає на успішне вивчення даної теми в умовах диференційного навчання. Основи програмування на базі середовищ візуального програмування можуть вивчатися без залучення складного математичного апарату, з учнями, в яких переважає гуманітарний склад мислення. Отримуючи практичний досвід конструювання застосування, вони виходять на рівень відповідних теоретичних узагальнень. Ряд школярів з емпіричним стилем мислення успішно виконують конструювання, обмежуючись мінімальними можливостями. Учні з розвинутим теоретичним мисленням, звільнившись від рутинності організації інтерфейсу, мають можливість більше уваги приділяти опрацюванню функціональної частини додатку.

Не секрет, що справжні знання народжуються тільки на етапі активної діяльності при наявності правильних зразків виконаних дій. Системи швидкого створення додатку мають ряд переваг по відношенню до інших систем саме з цієї точки зору. Досить часто, у зв'язку з економією часу, роботи учнів мають посередній

інтерфейс. Системи швидкого створення додатків дозволяють в стислі строки побудувати застосування з необхідним рівнем якості інтерфейсу користувача, надати додатку завершеності, і, найголовніше, зосередити увагу на пошуку алгоритму рішення самої задачі. Можливість створення технічно завершеного застосування позитивно впливає на мотивацію навчання, відходячи від принципу “програмування заради програмування”.

Аналіз літератури з методики викладання інформатики показав, що широке використання в навчанні учбової мови програмування Лого пояснюється можливістю створення наочних програм, і, як наслідок, підвищення інтересу учнів. Візуалізація процесу проектування дозволяє в значній мірі швидше побачити результат зусиль, роблячи його наочним і таким, що опирається на достатньо глибокі поняття та навички учнів, використання його як джерела позитивних емоцій, особливо при досягненні мети [5].

Досвід роботи в середовищі Windows та з чисельними офісними застосуваннями допоможе на інтуїтивному рівні сприйняти основні прийоми використання інтегрованого середовища розробки системи візуального програмування. В основу процесу навчання основ візуального програмування можна покласти метод адаптації, коли учні самостійно переносять знання, отримані раніше, на вивчення нового матеріалу, що особливо актуально в умовах скорочення. Існує і зворотній зв'язок, зміст якого полягає у більш осмисленому використанні принципів роботи елементів керування операційної системи на основі вивчення принципів роботи з конкретними компонентами.

Державний стандарт освіти передбачає вивчення об'єктно-орієнтованого підходу на рівні знайомства з основними принципами цього підходу. Багато методів та прийомів візуального програмування дозволяють розкрити учням основні положення технології ООП, виконуючи при цьому пропедевтичну функцію.

Запропонований нами підхід у вивченні основ програмування не має на меті вивчення окремих середовищ. Важливо, на нашу думку, щоб учні зрозуміли загальний вектор технологічної ідеї, акцентуючи увагу на осмислені сенсу візуального програмування без занурення в машинно-залежні деталі [5].

Класична схема методики викладання програмування раніше

полягала у наступному: спочатку теоретично вивчався синтаксис мови, а вже потім виконували конкретні завдання. Сьогодні ж, для написання застосування, не обов'язково знати особливості мови – достатньо розуміти загальну логіку процесу розробки за допомогою засобів візуального програмування.

Запропонована нами методика викладання засобів візуального програмування складається з наступних етапів:

- *етап проектування додатків в інтегрованому середовищі розробки (робота з візуальними об'єктами практично без використання навичок програмування);*
- *використання готових компонентів з написанням більш складних застосувань, ознайомлення з основними програмними конструкціями мови, що є базовою для цього середовища;*
- *створення власних компонентів та процес включення їх до бібліотеки компонентів;*
- *розробка практично корисних застосувань.*

На нашу думку, сучасні засоби візуального програмування більш доступні учням, ніж засоби традиційного програмування, вони викликають зацікавленість, володіють широкими дидактичними можливостями, дозволяють розкрити творчий потенціал учня і можуть бути з успіхом використані в системі освіти.

Література:

1. Дорошенко Ю.О., Прокопенко Н.С. Інформатика у структурі 12-річної середньої освіти. Тези доповідей Всеукраїнської науково-практичної конференції. (Дрогобич, 14-16 листопада 2000 р.). – Дрогобич: ДПУІФ, 2000. – С. 132-138.
2. Кузнецов А.А. О концепции содержания образовательной области «Информатика» в 12-летней школе // Информатика и образование. – 2000. – №7. – С. 2-7.
3. Жалдак М.І., Морзе Н.В. Методика ознайомлення учнів з поняттям інформації // Комп'ютер в школі та сім'ї. – 2000. – №4. – С. 3-8.
4. Носенко В.М. Завдання шкільної інформатики на сучасному етапі. Тези доповідей Всеукраїнської науково-практичної конференції. (Дрогобич, 14-16 листопада 2000 р.). – Дрогобич: ДПУІФ, 2000. – С. 140-142.
5. Бочкин А.И. Методика преподавания информатики: Учеб. пособие. – Мн.: Выш. шк., 1998. – 431 с., ил.

ДІАГНОСТИКА ЗНАНЬ І ВМІНЬ СТУДЕНТІВ ЗА ДОПОМОГОЮ ЕКСПЕРТНИХ СИСТЕМ

І.М. Пустинникова¹, В.В. Локтюшин²

¹ м. Донецьк, Донецький національний університет

² м. Донецьк, Донецький відкритий університет

Проектування навчальних експертних систем (ЕС) здійснюється з використанням оболонки Bayes Expert System Shell – BESS [1]. Рішення в цій ЕС комп'ютер приймає на основі байєсівського методу прийняття рішень. Математичний фундамент цього методу складає теорема Байєса. Вона дозволяє визначити апостеріорні ймовірності підтвердження і спростування гіпотез симптомами [1].

При використанні інструментальної системи BESS створення ЕС полягає, по суті справи, у створенні бази знань (БЗ). БЗ байєсівської ЕС складається зі знань про гіпотези і знань про симптоми. Знання про гіпотезу містять у собі ім'я гіпотези, текст припису, апріорну ймовірність гіпотези і множину симптомів, що впливають на гіпотезу, кожний із яких має ймовірності підтвердження і спростування цієї гіпотези. Знання про симптом містять у собі ім'я симптому, текст питання, відповідь на яке дозволяє визначити значення симптому, текст допомоги для користувача при відповіді на запропоноване питання й інформацію про тип відповіді (альтернативний, інтервальний чи лінгвістичний). Апріорні ймовірності гіпотез, а також ймовірності підтвердження і спростування гіпотез симптомами є експертними знаннями.

ЕС можуть виступати як інструмент навчання методам і засобам виділення головного в навчальному матеріалі. Інакше кажучи, ЕС виступає як засіб, за допомогою якого організується навчальна діяльність.

Застосування ЕС для діагностики може здійснюватися двома шляхами. Перший (традиційний) шлях полягає в тому, що студент працює з готовою (створеною заздалегідь) ЕС, і діагностика здійснюється за його відповідями на запропоновані системою питання. На нашу думку, такий підхід при великому обсязі матеріалу програє спеціальним діагностуючим системам. Він є ефек-

тивним при оперативному контролі знань, наприклад, при допуску до лабораторної роботи, експрес-контролі тощо [2, 3].

Другий шлях полягає в тому, що студент самостійно розробляє БЗ, і діагностика здійснюється за тими питаннями і завданнями, що сконструйовані самим студентом. Цей шлях, на нашу думку, є більш ефективним, оскільки несе велике навчальне навантаження. Він вимагає від студентів самостійності, творчого підходу, дає студенту можливість глибше розібратися в загальних і відмітних рисах досліджуваних їм фізичних явищ, процесів і законів, а викладачу побачити, де саме уявлення студентів помилкові або не зовсім точні, і відкоригувати їх. Допомогти студентам вникнути в деталі визначень, явищ, процесів, “відчути” їхні механізми – це одна з цілей занять, присвячених розробці БЗ ЕС.

Завдання по побудові БЗ ЕС використовуються при систематизації й узагальненні знань студентів як своєрідна підсумкова, залікова робота. При такому засобі діагностики знань/умінь студенти не просто репродуктивно відтворюють матеріал, а повинні систематизувати його, виділити основні поняття, правильно установити взаємозв'язки між ними.

Для того, щоб змусити ЕОМ що-небудь робити, моделювати або управляти процесом, останній спочатку треба зрозуміти [4, с. 10]. Ми вважаємо, що якщо студент зумів “пояснити” суть явища комп'ютеру, то можна бути впевненим, що він зрозумів матеріал. Зрозуміти щось можна і без пояснення, але пояснити щось не можна без розуміння цього “щось” [5, с. 63].

При розробці БЗ ЕС пропонується така методологія формування БЗ [6]. У першу чергу, для даної теми визначають усі поняття, які в ній вивчають. Текст припису по гіпотезі містить назву відповідного поняття. Потім аналітик (студент) виявляє всі знання (симптоми), що характеризують це поняття, тобто істотні, загальні і відмітні, необхідні і достатні ознаки понять. При цьому рекомендується така технологія. Використовуючи літературу і свій досвід (знання), аналітик спочатку визначає множину характеристик (ознак), необхідних, із його точки зору, для опису поняття. Після того, як проведена характеристика всіх понять, перевіряється можливість диференціації (поділу) кожної пари понять. Якщо в якомусь випадку диференціація неможлива, то уто-

чнюються характеристики понять, що дозволяють її здійснити. Потім аналітик формує їхні ймовірнісні оцінки.

Для визначення апіорних ймовірностей гіпотез будемо користуватися поняттям рівноможливості. Це значить, що викладач, пропонуючи завдання студентам, може вибрати будь-яку гіпотезу з набору, наявного в ЕС, і ймовірність вибору цих гіпотез однакова (наприклад, ймовірність того, що при роботі студента з експертною системою “Види руху матеріальної точки” викладач дасть завдання: “Результатом консультації з ЕС повинен бути припис “Тіло рухається рівносповільнено по прямолінійній траєкторії”, дорівнює ймовірності використання викладачем (у якості завдання для студента) будь-якого іншого припису, закладеного в БЗ даної ЕС), тому апіорні ймовірності гіпотез:

$$P(H_i) = 1/k \quad (i=1, 2, \dots, k), \quad (1)$$

де k – загальна кількість гіпотез [6].

Для визначення ймовірностей підтвердження і спростування гіпотези даним симптомом необхідно скласти таблицю відповідності між гіпотезами і симптомами, кількість стовпчиків якої відповідає кількості гіпотез (k), а кількість рядків – кількості симптомів (n). На перетинанні рядка і стовпчика ставиться знак “+”, якщо симптом має місце для даної гіпотези (відповідь “так” при відповіді на питання, що означає симптом) і знак “-” – у протилежному випадку (ця ж таблиця спрощує перевірку можливості або неможливості диференціації гіпотез за допомогою даного набору симптомів).

Ймовірності підтвердження гіпотез різноманітними симптомами дорівнюють або 0, або 1. Якщо в таблиці відповідності стоїть знак “+”, то ймовірність підтвердження гіпотези симптомом дорівнює 1, якщо знак “-”, то – 0.

Для знаходження ймовірності спростування гіпотези симптомом S_j необхідно визначити кількість N знаків “+” у рядку, що відповідає симптому S_j , і, якщо ймовірність підтвердження дорівнює 1, то обчислити ймовірність спростування за формулою [6]:

$$p = \frac{N-1}{k-1}; \quad (2)$$

якщо ймовірність підтвердження дорівнює 0, то обчислити ймовірність спростування за формулою [6]:

$$p = \frac{N}{k-1}. \quad (3)$$

Потім необхідно побудувати базу знань, використовуючи оболонку ЕС.

Узагальнюючи викладене вище, можна запропонувати такий алгоритм для побудови БЗ ЕС [6]:

1. Визначити явища (гіпотези), що треба описати, пояснити.
2. Скласти текст припису для кожної гіпотези.
3. Визначити властивості (симптоми), що характеризують дані явища (гіпотези).
4. Скласти таблицю відповідності “гіпотези – симптоми”.
5. Продумати тексти питань, при відповіді на які користувач означить симптоми.
6. Перевірити можливість диференціації кожної гіпотези за допомогою даного набору симптомів (у випадку неможливості диференціації повернутися до п. 2).
7. Визначити апіорні ймовірності гіпотез.
8. Визначити ймовірності підтвердження і спростування гіпотез симптомами.
9. Побудувати базу знань, використовуючи оболонку ЕС.

Розглянемо, наприклад, побудову БЗ “Фазові переходи першого роду”, використовуючи наведений вище алгоритм.

Як гіпотези прийняті такі фазові переходи:

H_1 – перехід “тверде тіло – рідина”;

H_2 – перехід “тверде тіло (лід і вісмут) – рідина”;

H_3 – перехід “рідина – тверде тіло”;

H_4 – перехід “рідина – тверде тіло (лід і вісмут)”;

H_5 – перехід “рідина – пара”;

H_6 – перехід “пара – рідина”;

H_7 – перехід “тверде тіло – пара”.

Їх визначають в загальній сукупності 4 симптоми, які є ознаками цих фазових переходів:

S_1 – упорядкування руху молекул;

S_2 – початковий стан – рідина;

S_3 – збільшення питомого об’єму;

S_4 – кінцевий стан – пара.

Нижче наведені тексти питань про значення відповідних симптомів:

1. Чи стає рух молекул у результаті процесу більш упорядкованим?
2. Чи знаходяться молекули на початку процесу у близькому порядку?
3. Чи збільшився питомий об'єм при фазовому переході?
4. Чи рухаються молекули наприкінці процесу хаотично?

Текст припису по гіпотезі є назвою відповідного їй фазового переходу (наприклад, текст припису по гіпотезі H_2 “Ви розглядаєте процес плавлення або льоду, або вісмуту”).

Апріорні ймовірності гіпотез визначаються за формулою (1) і для цієї БЗ $P(H_i)=1/7=0,143$ ($i=1, \dots, 7$).

Для визначення ймовірності підтвердження та спростування гіпотези даним симптомом побудуємо таблицю відповідності між гіпотезами і симптомами, в якій “+” значить, що на вище наведені питання по значенню симптому для даної гіпотези можна відповісти “так”, а “-” – “ні”.

Табл.1. Відповідність “гіпотези-симптоми” для бази знань “Фазові переходи першого роду”.

$S_j \backslash H_i$	H_1	H_2	H_3	H_4	H_5	H_6	H_7
S_1	-	-	+	+	-	+	-
S_2	-	-	+	+	+	-	-
S_3	+	-	-	+	+	-	+
S_4	-	-	-	-	+	-	+

Ймовірності підтвердження гіпотез різними симптомами дорівнюють або 0, або 1. Для знаходження ймовірностей спростування скористаємося формулами (2) та (3).

Наприклад, ймовірність підтвердження гіпотези H_2 симптомом S_1 $P_2^+(S_1)=0$, ймовірність спростування гіпотези H_2 симптомом S_1 (визначена за формулою (3)) $P_2^-(S_1)=3/7=0,429$.

Часто при формальному заучуванні визначення (без попереднього аналізу) у пам'яті студента залишаються лише “шматочки”, а “несуттєві” (із погляду студента) деталі зникають. Для побудови БЗ ЕС необхідно досконально розібратися в навчальному матеріалі. Дослідження показали [7, с. 170], що основним прийомом по осмисленню тексту є постановка читачем перед собою прихованого питання і пошук відповіді на нього.

При побудові ЕС одним з основних моментів є складання питань, при відповіді на які означаються симптоми. Ідея про діалог як засіб пізнання і про головну роль питань у діалозі розглядається у філософії [8, с. 8]. Ми намагаємося використати її як центральну при діагностиці знань студентів шляхом побудови БЗ ЕС, оскільки питання є формою руху думки, у ньому яскраво виражений момент переходу від незнання до знання, від неповного, неточного знання до більш повного і більш точного знання [9]. Справедливо писав С.Л. Рубінштейн: «Возникновение вопросов – первый признак начинающейся работы мысли и зарождающегося понимания» [10, с. 352]. Л.П. Доблаєв [7] схильний розглядати самостійну постановку питань студентами і пошук відповіді на них як основний прийом досягнення найкращого розуміння тексту.

Ступінь оволодіння матеріалом визначається по якості створеної студентом БЗ. Викладач перевіряє БЗ, тестуючи її або сам, або за допомогою іншого студента. У випадку адекватної реакції системи БЗ може надалі використовуватися для діагностики традиційним методом (за відповідями студента). Інакше визначається причина неадекватності, якою може бути неправильне застосування алгоритму побудови ЕС або недостатня якість засвоєння предметних знань студентом, і вносяться виправлення. Таким чином, даний підхід дозволяє діагностувати як предметні знання, так і вміння будувати ЕС.

При використанні готової БЗ ЕС із метою діагностики знань викладач повідомляє студентам різноманітні гіпотези, які є цілком експертизи. Задача студентів полягає в тому, щоб відповідаючи на питання, запропоновані ЕС, домогтися збігу “припису”, що видається ЕС наприкінці роботи, із гіпотезою, яка була задана викладачем. Якщо це вдалося, то, виходить, студент знає, які ознаки (симптоми) характеризують дану гіпотезу, якщо ні – то, використовуючи підсистему пояснення, студент може самостійно визначити, де він помилився, і, при повторній роботі із системою, успішно справитися з завданням.

Приклади побудови ЕС наведені в роботах [2, 3, 6].

Перелік посилань

1. Петрушин В.А. Экспертно-обучающие системы. – К.: Наук. думка, 1992. – 196 с.
2. Атанов Г.А., Пустынникова И.Н. Обучение путем построения баз знаний для экспертных систем // Искусственный интеллект. – 1998. – № 2. – С. 42–48.
3. Пустынникова И.Н. Методология конструирования диагностирующей экспертной системы (на базе оболочки BESS) // Вісник Донецького університету. – Серія А. Природничі науки. – 1998. – № 1. – С. 182–187.
4. Вайзенбаум Дж. Основной доклад // Образование и информатика. Укрепление международного сотрудничества: Тезисы междунар. конгресса (ЮНЕСКО, Париж 12–21 апреля 1989 г.): Пер. с англ., фр. – К.: Ин-т кибернетики им. В.М. Глушкова АН Украины, 1992. – Т. 1. – С. 4–12.
5. Черняк В.С. О смысле понимания и понимании смысла // Вопросы философии. – 1986. – № 8. – С. 59–63.
6. Атанов Г.А., Пустынникова И.Н. Диагностика знаний / умений с помощью экспертных систем: Учебное пособие для студентов физического факультета. – Донецк: ДонГУ, 1997. – 64 с.
7. Доблаев Л.П. Смысловая структура учебного текста и проблемы его понимания. – М.: Педагогика, 1982. – 176 с.
8. Любарский Ю.Я. Интеллектуальные информационные системы. – М.: Наука, 1990. – 227 с. (Пробл. искусств. интеллекта)
9. Лимантов Ф.С. О природе вопроса // Вопрос. Мнение. Человек. – Л.: Уч. записки ЛГПИ им. Герцена, 1971. – Т. 497. – С. 4–20.
10. Рубинштейн С.Л. Основы общей психологии. – М., 1946. – 704 с.

ПРОБЛЕМИ ІНФОРМАТИЗАЦІЇ СУЧАСНИХ ТЕХНОЛОГІЙ НАВЧАННЯ

Ю.П. Рева

м. Кривий Ріг, Криворізький технологічний факультет
Кременчуцького інституту економіки та нових технологій

Система освіти нашої країни переживає період фундаментальних змін, які характерні новим розумінням її мети та цінностей, усвідомленням необхідності переходу до неперервної освіти, новими концептуальними підходами до розробки та використання сучасних технологій навчання тощо. Реалізація багатьох із поставлених перед системою освіти задач неможлива без використання методів та засобів інформатизації.

Провідне місце в розробці проблеми інформатизації технологій навчання на сучасному етапі займають роботи, присвячені аналізу дидактичних можливостей комп'ютерів, їх програмного забезпечення.

Такий підхід, безперечно, є важливим і необхідним, але він достатньо односторонній і пов'язаний з аналізом потенціалу тільки одного компонента педагогічних технологій – засобів навчання. Разом із тим, сучасне розуміння педагогічної технології, розкрите в роботах В.П. Безпалька, В.І. Богомолова, О.О. Вербицького, О.Я. Савельєва, Н.Ф. Тализіної, В.В. Юдіна, Ф. Янукевича та інших, дає можливість говорити про педагогічну технологію як про цілісну систему компонентів, до якої входять задачі, мета, зміст, процес навчання і т. ін.

Таким чином, можна констатувати, що має місце протиріччя між необхідністю системного підходу до аналізу процесу інформатизації технології навчання, що включає аналіз впливу інформатизації на всі компоненти технології як цілісної системи, і переважаючим у теорії та практиці одностороннім підходом, пов'язаним, в основному, з аналізом дидактичних можливостей засобів інформатизації процесу навчання.

Слід відмітити, що до сьогоденного дня автори ряду досліджень, присвячених даній проблемі, виходили, як правило, з аналізу можливостей і потенціалу самих засобів інформатизації, а вже потім виявленні можливості намагались

реалізувати, “прикласти” до існуючих технологій. Ми ж дотримуємось іншого підходу до поставленої проблеми – йти від аналізу суті технологій навчання, складу і взаємозв’язків їх компонентів до аналізу й обґрунтуванню використання методів та засобів інформатизації при їх розробці.

Постановка проблеми в такому аспекті вимагає нових підходів до аналізу змісту, методів, організаційних форм і засобів навчання в умовах інформатизації процесу навчання.

Ми вважаємо, що ефективність інформатизації технологій навчання буде досягнуто, якщо: по-перше, самі технології навчання будуть представлені як системний метод проектування (від мети до результатів навчання), реалізації, корекції і наступного відтворення процесу навчання; по-друге, інформатизація технологій навчання повинна бути спрямована на всі його компоненти, а не лише впровадження нових (комп’ютерних) засобів навчання; по-третє, інформатизація технологій навчання буде орієнтована не лише на потреби та специфіку змісту навчальної дисципліни, але, насамперед на розвиток особистості учня.

Перехід до неперервної освіти, використання нових інформаційних технологій навчання не могли не вплинути на розуміння сутності й трактування поняття “технологія навчання”. На сьогоднішній день існують різні термінології у поясненні цього поняття. Достатньо довгий проміжок часу технологія навчання ототожнювалась з простим використанням технічних засобів навчання. Вона розглядалась лише як засіб технологізації процесу навчання.

Після цього технологію навчання стали визначати (Є.І. Машбіц, Н.Ф. Талізін та ін.) як спосіб забезпечення науковими принципами процесу проектування нової чи модернізованої практики навчання. Саме це і дає можливість наукового обґрунтування мети та задач навчання, відбір адекватного предметного змісту, вибір об’єктів вивчення та видів діяльності для розробки учбового курсу; методики оцінки досягнень кінцевого продукту цього курсу і т.п.

Важливим для розуміння суті освітніх технологій є їх співставлення з методикою навчання або методичною системою навчання. А суттєвою характеристикою технології навчання є її

орієнтація на попередньо задані та чітко описані результати навчання.

Таким чином, головними особливостями технологій навчання є:

- проектування технологій навчання здійснюється не на основі узагальнення досвіду, а на основі наукового пізнання практики навчання;
- освітня технологія відрізняється стійкістю результатів незалежно від багатьох факторів та умов навчання;
- технології навчання орієнтуються на попередньо задані та чітко сформульовані, а не припущені результати навчання.

Отже, технологія навчання – це системна категорія, орієнтована на дидактичне застосування наукових знань, наукових підходів до аналізу й організації учбового процесу з урахуванням емпіричних інновацій викладачів та спрямованості на досягнення значних результатів у розвитку особистості учня.

Структурними складовими такої системи є:

- задачі навчання та діагностично-задані заплановані результати навчання;
- зміст навчання;
- засоби діагностики та контролю стану результатів навчання;
- методи навчання;
- організація учбового процесу;
- засоби навчання;
- учні;
- викладач;
- результати діяльності.

Такий підхід до розуміння суті та структури технології навчання визначає, що під інформатизацією технологій розуміють не тільки застосування засобів інформатизації навчання, але й системний підхід до цього процесу, який включає інформатизацію цілого ряду інших компонентів технології.

Розглядаючи вплив інформатизації освіти на розвиток технології навчання, ми, насамперед, повинні відмітити два основних аспекти цієї проблеми.

Перший аспект, який умовно можна назвати “змістовним”, пов’язаний з впливом інформатики (як фундаментальної галузі

наукових знань) та інформаційних технологій (як постійно змінної сфери діяльності людини) на мету та зміст навчання. Очевидно, що місце та роль вивчення інформатики та інформаційних технологій у загальноосвітній, загальнонауковій і спеціальній підготовці будуть посилюватись і цей процес, безперечно, знайде відображення в технології навчання.

Другий аспект, який умовно можна назвати “процесуальним” пов’язаний з впливом інформатизації освіти на проектування та реалізацію технологій навчання.

В цьому смислі під інформатизацією технологій навчання розуміють наукові концепції, основані на інформаційному підході до аналізу будови, функціонуванню й управлінню системою освіти, методи та засоби інформатизації, пов’язані з визначенням нових форм і засобів представлення знань та вмій у змісту навчальних курсів, вдосконаленням методів і організації навчання, застосуванням нових способів навчання, які базуються на інформаційних та телекомунікаційних технологіях.

Одною з основних проблем теорії та практики розробки технології навчання залишається проблема відбору змісту. Вирішення цієї проблеми пов’язано, зокрема, із систематизацією основних понять курсу навчання. Для ефективності реалізації процедур аналізу й синтезу системи понять і побудови моделі знань необхідно використання як традиційних, так і формальних методів. Поширеними формальними методами, які використовуються в цьому випадку є графові, а також сітьові моделі. Інший напрямок застосування методів та засобів інформатики пов’язаний з обробкою результатів моніторингу та даних перевірки знань і вмій учнів.

І останнє. Важливим аспектом застосування методів інформатики в технологіях навчання є побудова інформаційних моделей процесу навчання з позиції теорії управління і з позиції інформаційних процесів, які реалізуються в освітніх системах.

МЕТОДИКА ОЦІНКИ ЗНАТЬ З ДИСЦИПЛІНИ “ІНФОРМАТИКА І КОМП’ЮТЕРНА ТЕХНІКА”

О.Ю. Рудик, С.Р. Красильников
м. Хмельницький, Технологічний університет Поділля

Одним з основних джерел знань є лекція, яка формує погляди та переконання студентів. Вона озброює їх методикою роботи над курсом, книжкою, дає можливість вияснити головне і другорядне у дисципліні, орієнтуватись у літературі. Проведення лекцій у мультимедійному класі засобами PowerPoint сприяє кращому засвоєнню матеріалу.

Основне призначення практичних і лабораторних занять – закріплення знань, отриманих на лекціях і при роботі над підручником; надбання навичок у розв’язку основних типів задач під керівництвом викладача. Задачі складаються з урахуванням профілю університету і вибраної студентами спеціальності. В кінці кожного практичного чи лабораторного заняття вказується тема наступного заняття, рекомендується відповідна література і попереджається про проведення індивідуальної контрольної роботи з теми проведеного заняття.

Для контролю відвідування занять та оцінки знань і навичок студентів розроблена електронна таблиця (табл. 1), в яку заносяться дані про їх присутність на заняттях та оцінки з усіх видів занять.

Таблиця 1

№ сем	Прізвище, ім'я, по-батькові	п1	п2	п3	п4	п5	п6	п7	п8	п9	п10	п11	п12	п13	п14	п15	п16	п17	п18	
		п1	п2	п3	п4	п5	п6	п7	п8	п9	п10	п11	п12	п13	п14	п15	п16	п17	п18	
10	Штофко Ірина Степанівна			5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
11	Ялук Валентина Володимирівна			5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Ла61 - VBA

	IfThenExample
IfThenElseExample	SelectCaseExample

Ла62 - VBA

	Макрос FormattingMacro
Кнопка FormattingMacro	СurrentDate
	Довжина кола

Ла63 - VBA

	InputDialog
MsgBoxExample1	MsgBoxExample2

Ла64 - VBA

	ForNextExample
	ForEachNextExam
DoLoopExample1	WithEndWithExam
DoLoopExample2	

Лекції (л): проводиться вибіркове опитування; якісна оцінка відповіді проводиться за загальноприйнятими критеріями.

Практичні заняття (пр) – виставляється три оцінки:

1. Кожне заняття (крім першого) починається контрольною індивідуальною роботою, у якій 5 завдань, правильна відповідь на кожне з яких оцінюється в один бал (тому шкала оцінок 0, 1, 2, 3, 4, 5).

2. Під час проведення контрольної роботи перевіряється домашнє завдання, яке оцінюється за шкалою 0...5 (“0” – завдання з розв’язком відсутнє).

3. Оцінюються практичні навички з розв’язування задач біля дошки за загальноприйнятими критеріями; перші три студенти, які розв’язали задачу без помилок раніше студента біля дошки, отримують “5”.

Індивідуальна робота з теорії та перевірка виконання завдань з теми попереднього заняття дозволяють вяснити, засвоєння яких питань і методів розв’язку задач викликає у студентів найбільші труднощі. При цьому відстаючим студентам надається така допомога, яка не знижує їх активності на заняттях.

Заохочення перших трьох студентів відмінними оцінками викликає дух змагання і примушує індивідуально працювати над завданнями, а не бездумно переписувати розв’язок з дошки. Але при цьому наголошується, що студентам, які не зрозуміли суть та методику розв’язування задач, краще орієнтуватись на розв’язуванні задач на дошці. Від студента біля дошки вимагаються детальні пояснення.

Лабораторні роботи (лаб) – виставляється три оцінки:

– у кожній лабораторній роботі три завдання, оцінка виконання яких проводиться за шкалою 0...5 (“0” – завдання не почали виконувати – тобто оцінюється також і швидкість роботи на ПК).

Згідно прийнятої шкали оцінок у випадку відсутності студента на практичних чи лабораторних заняттях в електронну таблицю виставляється “н” і три нулі. Це дає змогу програмі підраховувати кількість пропусків та рівень знань: “0” і “1” свідчать, що студент не виконав більшість із запропонованих завдань.

Отже, на кожному практичному (крім першого) і лабораторному занятті виставляється по три оцінки, а у першу клітинку –

присутність (вірніше, відсутність) студента на занятті – буква “н”. Останніх два стовпчика таблиці після кожного введення даних автоматично підраховують:

– ср. оц. – середню оцінку із заокругленням у більшу сторону [Excel: =ROUND(AVERAGE(range);0)];

– к. пр. – кількість пропусків, тобто букв “н” [Excel: =COUNTIF(range;”н”)].

Якщо середня оцінка за певний період становить 0, 1 чи 2, студент рахується неатестованим за даний період і не допускається до екзамену, доки не виправить незадовільні оцінки:

– отримані на лекційних та практичних заняттях – оформленням та захистом рефератів з тем, які оцінені незадовільними оцінками;

– отримані на лабораторних заняттях – відробкою лабораторних робіт (під таблицею вказані назви тем і завдань).

Електронні таблиці (тобто журнали) розміщуються на кожному ПК (Windows 98, DeskTop). Оголошені викладачем оцінки заносяться кожним студентом індивідуально у свій журнал, щоб в кінці курсу мати наступний вибір:

– погодитись із середньою оцінкою для занесення її у залікову книжку;

– перездати завдання, оцінка яких їх не задовольняє, з метою підвищення середньої оцінки;

– підтвердити вищий рівень знань на екзамені.

Таким чином, запропонована методика забезпечує наочність системи оцінки знань, самодисципліну, сприяє покращенню успішності, автоматизує процес її контролю на протязі вивчення курсу як зі сторони викладача, так і зі сторони студента.

ОРГАНІЗАЦІЯ ОБЛІКУ ВИКОРИСТАННЯ РЕСУРСІВ INTERNET У НАВЧАЛЬНОМУ ПРОЦЕСІ

М.В. Сальніков, В.Г. Кудрявцев
м. Хмельницький, Технологічний університет Поділля

Будь яка корпоративна мережа, що має доступ до ресурсів Інтернету сплачує досить великі кошти провайдерам, тому є природнім, щоб системний адміністратор мав повну та достовірну інформацію про те, яким чином ці ресурси використовуються. Основним показником того, як використовуються ресурси Internet, є обсяг отриманої інформації. Крім того, більшість провайдерів диференціюють ціни на свої послуги в залежності від часу доби, тому час коли співробітники організації користуються Internet також повинен фіксуватися. До того ж, звичайно потрібно знати, чи використовуються ресурси Internet за призначенням (для цього потрібно реєструвати адреси серверів, з яких було отримано інформацію).

Щоб отримати всю вищеперераховану інформацію, використовується спеціальна програма Sqmgrlog. Ця програма також має багато інших можливостей, наприклад, може вводити квоти на обсяги отриманої інформації для кожного користувача. Sqmgrlog встановлюється на машині, що використовується як Proху-сервер. Proху-сервер веде спеціальні log-файли, де серед іншого знаходяться відомості про:

1. IP-адресу машини, до якої надсилається запит,
2. метод (GET, POST, ...);
3. URL об'єкту;
4. статус – результат запиту (TCP_HIT – для об'єктів, що були взяті з кеша, TCP_MISS – якщо об'єкт взято не з локального кешу);
5. HTTP – код результату, що повертає HTTP (200 – для вдалих, 403 – для перенаправлень, 500 – для помилок...);
6. Size – розмір переданої інформації (байт);
7. Hier_status – результат запитів до братніх або батьківських кешів Parent_Miss, Sibling_Hit, ...;
8. Hier_Host – хост, з якого було отримано об'єкт;
9. Time – час у мілісекундах, починаючи з 1.01.1970 р.

10. Elapsed – час який було затрачено в мілісекундах;

Саме на основі log-файлів (для Proxy-сервера Squid це файл Access.log) програмою Sqmgrlog формується вся результуюча інформація.

Щоб встановити Sqmgrlog, потрібно виконати наступні дії:

1. Перейти в каталог, куди було скопійовано інсталяційні файли;

2. run ./configure;

3. make;

4. make install;

5. Перейти в каталог /usr/local/etc/sqmgrlog та змінити конфігураційний файл sqmgrlog.conf згідно ваших потреб. Так, наприклад, потрібно буде вказати каталог де знаходиться файл access.log та де будуть розміщені результуючі HTML документи.

Більш докладну інформацію щодо інсталяції можна знайти в файлі README.

Після інсталяції необхідно запустити програму /usr/local/bin/sqmgrlog. Результатом виконання програми sqmgrlog буде створення каталогу /usr/local/etc/httpd/htdocs/squid-reports (цей шлях використовується за замовчуванням). В цьому каталозі знаходяться HTML-файли, що показують використання Internet кожним користувачем (з вказанням IP-адреси) та інша додаткова інформація. Щоб статистика оновлювалася автоматично, потрібно запускати sqmgrlog щодня. Для цього використовується команда операційної системи – crontab, яка дозволяє виконувати певні дії з вказаною періодичністю. Більш докладно про використання команди crontab можна прочитати в літературі, присвяченій операційній системі Unix.

Не зважаючи на широкі можливості закладені в програму sqmgrlog, для великої організації (такої, як наприклад, університет) їх може бути недостатньо. Якщо організація складається з кількох підрозділів, то необхідно вести статистику в розрізі частин організації. Крім того, так як рахунки провайдером сплачуються щомісяця, то потрібно сумувати скільки було отримано інформації по місяцях. Для вирішення цих проблем нами було розроблено систему, яка складається з двох частин. Перша – це програми, що написані на мові PHP. Друга частина – це база даних, розташована на SQL-сервері (нами було обрано SQL-

сервер – MySQL 3.22.23b). Докладніше опишемо кожну з цих частин.

Почнемо з опису структури бази даних, адже програми призначені для роботи саме з нею. Назва бази даних STAT. В MySQL 3.22.23b для Red Hat Linux 6.0 БД знаходиться в каталогах. Так БД STAT розташована в каталозі /var/lib/mysql/stat/. База даних складається з шести таблиць. Їх структура та призначення наведені нижче:

1. DSQUID – довідник Проху-серверов;

Таблиця 1

idsquid	namesquid
int(11)	char(20)

2. DKAF – довідник кафедр та підрозділів університету;

Таблиця 2

idkaf	nkaf
int(11)	char(250)

3. IP – тимчасова таблиця, до якої заносяться щоденні обсяги інформації отриманої з Internet;

Таблиця 3

ip	size	idsquid
char(20)	int(11)	int(11)

4. IPMONTH – таблиця яка містить інформацію про те яка машина, через який Проху-сервер, за який місяць, скільки отримала інформації;

Таблиця 4

iddate	ip	sum	idsquid
char(7)	char(20)	int(11)	int(11)

5. SOOTV – довідник, який вказує відповідність між кафедрами та IP-адресами комп'ютерів, що належать до них.

Таблиця 5

idkaf	ip
int(11)	char(20)

6. CHECK – допоміжна таблиця, що показує чи було занесено дані по певному дню до бази даних (створена на випадок нестабільного зв'язку між SQL-сервером та Проху-сервером).

Таблиця 6

iddate	flag	idsquid
char(10)	char(1)	int(11)

Зв'язки між таблицями встановлюються при написанні SQL-запитів, а самі запити виконуються як частина програм, використовуючи засоби мови PHP3. Далі розглядається програмна частина системи.

Перша програма `proba.php3` на основі HTML файлів, що генеруються програмою `sqmgrlog`, створює текстовий файл `temp.txt`, з структурою, аналогічною до структури таблиці IP.

Цей файл знаходиться вже на SQL-сервері (в папці `/var/lib/mysql/stat/temp.txt`), тому програма, використовуючи відповідний SQL-запит, заносить зміст файлу до таблиці IP. Після цього `Proba.php3` запускає програму `Input.php3`.

`Input.php3` спочатку визначає системну дату. Далі перевіряє, використовуючи дані з таблиці CHECK, чи заносилися сьогоднішні дані з `temp.txt` до бази даних. Якщо "так", то програма закінчує свою роботу. В іншому випадку програма робить вибірку з таблиці IP і перевіряє, чи зустрічається кожна ip-адреса в підсумковій таблиці IPMONTH. Якщо "ні", то в IPMONTH заноситься новий запис, інакше до поля SUM таблиці IPMONTH додається кількість байт отриманих на поточний день (з таблиці IP). Після занесення всіх даних програма виконує SQL-запит, який записує в таблицю CHECK прапор, який вказує на те, що поточні дані вже занесені до бази даних.

Ці програми повинні запускатися щоденно після того, як закінчить свою роботу `sqmgrlog`.

Також написані програми, які дозволяють побачити результати роботи системи (вони відображують зміст таблиці IPMONTH у зручному для сприйняття вигляді).

Отже використовуючи систему, що описана вище, можна досить легко та наочно організувати облік ресурсів INTERNET у навчальному процесі для досить великих навчальних установ.

НАСТУПНІСТЬ ВИВЧЕННЯ КУРСУ “КОМП’ЮТЕРНА ТЕХНІКА”

В.А. Сергієнко

м. Суми, Сумський державний аграрний університет

Для нашого часу є характерним надзвичайно швидкий технічний прогрес. Комп’ютери та інформаційні технології відіграють у цьому важливу роль. Комп’ютер повинен стати звичайним помічником у роботі для будь якого спеціаліста. Тому процес знайомства з комп’ютерною технікою розпочинається в школі.

Викладання предмета “Основи інформатики та обчислювальної техніки” в загальноосвітніх школах ведеться з 1984 року. За цей період тричі змінювалася направленість шкільного курсу інформатики. Якщо в програмі з інформатики зразка 1985 року домінував “програмістський” ухил (програмування вважалося основним засобом інформаційної технології, а сам курс був орієнтований на обробку інформації), то в курсі інформатики зразка 1993 року й особливо зразка 1996 року більша увага приділяється прикладній інформатиці (40% навчального часу), скорочено розділ інформатизації та програмування (30% навчального часу). Крім того акцент змістився з вивчення технічної будови комп’ютерів до вивчення його програмного забезпечення.

Починаючи з 1985 року загальноосвітні школи комплектуються комп’ютерною технікою: спочатку КУВТ “УАМАНА”, КУВТ-86, КОРВЕТ. Наприкінці 80-х років, згідно з проектом “Пілотні школи”, починається оснащення шкіл ІВМ-сумісними комп’ютерами (в Україні було створено 160 таких шкіл), з 1991 року в школи почали надходити навчально-інформаційні комплекси на базі комп’ютерів “Пошук-2”. З 1996 року скорочуються темпи постачання шкіл комп’ютерною технікою, швидко морально та фізично старіє існуюча техніка, лише деяким школам вдається придбати сучасні комп’ютери.

Отже, на сьогоднішній день викладання інформатики в загальноосвітніх школах ведеться за:

- безмашинним варіантом. У випускників таких шкіл відсутні практичні навички роботи з комп’ютером.
- машинним варіантом з використанням “застарілої” техніки.

- машинним варіантом з використанням сучасних комп'ютерів. В більшості таких шкіл проводиться розширене вивчення інформатики (починаючи з 8-го класу). Учні цих шкіл здобувають навички роботи з прикладним програмним забезпеченням загального та цільового призначення, роботи з електронною поштою, пошуку інформації в мережі Інтернет, складання та реалізації найпростіших алгоритмів і програм з використанням засобів навчальної алгоритмічної мови та реальних мов програмування і операційних систем.

В зв'язку з цим при продовженні навчання в навчальних закладах освіти III–IV рівнів акредитації виявляється різний рівень підготовки першокурсників, тому програма дисципліни “Комп'ютерна техніка та програмування”, який викладається на першому курсі, орієнтована на недостатню підготовленість основної маси студентів і знайомить їх з основами комп'ютерної техніки, програмним забезпеченням ЕОМ, алгоритмізації та програмування. При цьому меншість першокурсників володіють навичками роботи з комп'ютером.

Тому доцільно, щоб при вступі до вузу абітурієнти володіли навичками роботи з основними складовими сучасного програмного забезпечення ЕОМ, були ознайомленими з функціональним призначенням основних пристроїв комп'ютера та принципами їх будови і дії, основами технологій розв'язання задач за допомогою комп'ютера, починаючи від їх постановки й побудови відповідних інформаційних моделей і закінчуючи інтерпретацією результатів, отриманих за допомогою комп'ютера.

Не зважаючи на ряд об'єктивних причин роботу в школах потрібно направити на вирішення цих проблем.

В Сумському державному аграрному університеті проблему комп'ютерної грамотності майбутніх студентів вирішує факультет довузівської підготовки. На протязі навчального року на базі кафедри кібернетики та інформатики в рамках програми підготовчих курсів проводяться заняття для абітурієнтів, що навчаються на факультет довузівської підготовки. Тут вони здобувають навички роботи з пристроями введення-виведення інформації, прикладним програмним забезпеченням загального та цільового призначення: програмами для роботи з електронною поштою, пошуку інформації в глобальній мережі Інтернет.

ДОСВІД ВИКЛАДАННЯ ДИСЦИПЛІН КОМП'ЮТЕРНОГО ЦИКЛУ ПРИ ПІДГОТОВЦІ СПЕЦІАЛІСТІВ АПК

Г.А. Смоляров

м. Суми, Сумський державний аграрний університет

Широке розповсюдження та застосування комп'ютерної техніки у всіх галузях діяльності людини гостро ставить питання опанування комп'ютерною грамотністю.

Тому одним з важливих завдань при реалізації ступеневої системи підготовки фахівців АПК є формування у студентів на кожному ступені навчання знань та навичок застосування сучасних комп'ютерних технологій в практичній роботі. Слід також відмітити, що інформатика дуже динамічна наука – з'являються нові моделі ПЕОМ та пакети прикладних програм і, разом з ними, нові інформаційні технології, а використання обчислювальних мереж дозволяє будувати нові, ефективні інформаційні системи підприємств та організацій.

Концепція викладання комп'ютерних дисциплін для економічних спеціальностей аграрних вузів передбачає вивчення таких дисциплін: “Комп'ютерна техніка”, “Програмне забезпечення ЕОМ”, “Інформаційні системи в менеджменті”.

Метою дисципліни “Комп'ютерна техніка”, яка викладається на першому курсі, є надання студентам навичок практичної роботи з ПЕОМ, вивчення технічного та системного програмного забезпечення, навчання основам алгоритмізації та програмування. Вивчення дисципліни включає аудиторні лабораторні та практичні заняття, а також роботу з навчаючими та тестовими програмами.

Викладання цієї дисципліни пов'язано з тим, що в більшості контингент студентів вперше знайомиться з комп'ютером. Це вимагає організацію індивідуального підходу та інтенсифікації процесу навчання, надання студентам можливості вільного доступу в аудиторії для додаткової роботи на комп'ютері. При контролі знань використовується модульно-рейтингова система, яка дозволяє структурувати дисципліну, своєчасно, в повній мірі оцінювати знання студентів та активізувати їх навчання.

Метою дисципліни “Програмне забезпечення ЕОМ”, яка ви-

кладається на другому курсі, є поглиблене вивчення етапів рішення задач на ЕОМ та програмування, навчання навичкам застосування обчислювальної техніки та прикладних програм в процесі розв'язання економічних задач. Організація викладання цієї дисципліни передбачає лекційний курс, аудиторні лабораторні та практичні заняття під керівництвом викладача, а також виконання самостійних індивідуальних завдань. Це забезпечує закріплення теоретичних знань та сприяє отриманню практичних навичок і розвитку самостійного наукового мислення.

Оволодівши основами інформатики та програмування, спеціаліст зможе працювати з ПЕОМ, використовувати отримані знання для розв'язання задач обліку, аналізу, планування, при вивченні спеціальних дисциплін, курсового та дипломного проектування.

Завершальною дисципліною комп'ютерного циклу є дисципліна "Інформаційні системи в менеджменті", мета якої полягає в вивченні основ побудови інформаційних систем, їх застосування в процесі автоматизації функцій управління підприємством та вирішення економічних та управлінських задач. В залежності від спеціальності ця дисципліна викладається на третьому або четвертому курсі. Особливістю організації викладання цієї дисципліни є те, що застосування інформаційних систем сьогодні треба розглядати не тільки як відокремлену дисципліну, а як предмет, тісно пов'язаний з основами менеджменту, маркетингу, обліку, аналізу, фінансово. діяльністю та бізнесом. Використання сучасної комп'ютерної техніки та програмних засобів залежить не тільки від рівня комп'ютерної грамотності, а й від уміння використовувати сучасні інформаційні технології для вирішення практичних задач виробництва. Виходячи з цього, слід відмітити необхідність коригування учбових програм в зв'язку з поновленням технічних та програмних засобів, які використовуються в учбовому процесі та відповідності програми комп'ютерних дисциплін сучасним вимогам. Таким чином, досвід викладання дисциплін комп'ютерного циклу при підготовці економістів АПК в СДАУ показав, що вирішення завдань навчання сучасного спеціаліста потребує не тільки застосування новітніх технічних та програмних засобів, а й підтримки рівня їх викладання у відповідності з вимогами часу.

ОСОБЛИВОСТІ ВИКЛАДАННЯ ІНФОРМАТИКИ ДЛЯ НЕКОМП'ЮТЕРНИХ СПЕЦІАЛЬНОСТЕЙ У ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДАХ

А.М. Стрюк

м. Кривий Ріг, Криворізький технічний університет

Викладання основ обчислювальної техніки є найбільш актуальним для навчальних закладів будь-якого рівня. Сучасний спеціаліст повинен крім глибоких знань з безпосередньої сфери своєї діяльності, володіти принаймні загальними навичками роботи з обчислювальною машиною. Тоді комп'ютер стане незамінним помічником йому в роботі. Від сучасних навчальних закладів вимагається надавати майбутньому спеціалісту якомога глибше розуміння основ побудови та функціонування обчислювальної техніки, її можливостей, а також імовірну сферу її використання та особливості розв'язання задач за допомогою ЕОМ.

Ще зовсім недавно перед навчальними закладами поставали дуже складні проблеми, пов'язані з відсутністю необхідної обчислювальної техніки або невідповідністю її сучасним вимогам. Зараз ці проблеми поступово відходять на другий план. Постає інша проблема: як побудувати процес навчання таким чином, щоб дати студенту найбільш глибокі та універсальні знання, які він міг би застосовувати в майбутній своїй діяльності.

Відомо, що найбільш глибокі знання і, головне, розуміння основних принципів функціонування обчислювальної машини дає практика програмування обчислювальної машини з використанням будь-якої мови програмування високого рівня. Але, розуміючи це, велика кількість викладачів схилиються до того, щоб зробити акцент у викладанні на прикладні програмні засоби, такі як текстові редактори та електронні таблиці, а програмуванню приділити другорядну увагу, або не приділяти уваги взагалі. Аргументується це тим, що сучасних прикладних програм достатньо для вирішення більшості задач, а засвоєння основ програмування вимагає від студента набагато більших зусиль, використовуватись на практиці здобуті ж знання не будуть. З цим не можна погодитися. Звичайно, загальні знання про розмаїття сучасних прикладних програм потрібні, як потрібна і практика роботи з

декотрими з них. Сучасні програмні засоби дійсно дозволяють вирішити більшість тривіальних задач. Але вони можуть виявитися нездатні вирішити якусь нескладну, але нетипову задачу, оскільки їх використання виявиться занадто не раціональним. До того ж вони не дають глибокого розуміння сутності функціонування ЕОМ і дуже мало стимулюють творчий підхід до роботи з комп'ютером. Забезпечити все це може лише вивчення основ програмування. Це підтверджують переконливі результати спостережень, за роботою студентів, які отримали знання з основ програмування. Вони дуже легко і швидко освоюють будь-яку прикладну програму і знаходять їй ефективне та раціональне застосування, в той час як більшість з учнів, що отримали відомості лише про прикладні програми, дуже важко адаптуються при роботі з іншими програмними продуктами і не в змозі оптимально підійти до вирішення нестандартних задач. А в тому, що програмування складно вивчається і мало застосовується студентами на практиці провина не самого програмування, а форм його викладання. І особливо це стосується побудови та проведення практичних занять. Традиційно при вивченні програмування на практичних заняттях ставляться ті ж самі тривіальні задачі, що легко можна вирішити, не застосовуючи знань з основ програмування. А найголовніше те, що ці задачі, ілюструючи якусь окрему програмну структуру, є відокремленими одна від одної. Студент не отримує повного уявлення про процес програмування, про взаємодію різних програмних структур тощо. Це нагадує учня першокласника, що вже вивчив кожну букву окремо, але ціле слово прочитати ще не може. Звичайно, отримавши навички на такому рівні, студент не зможе застосувати їх для вирішення реальної задачі. Для того, щоб дати студенту повну картину, перед ним потрібно поставити задачу максимально наближену до реальної, в процесі виконання якої він застосував би всі отримані знання і навички. При вивченні більшості дисциплін такою задачею для студента є написання курсової роботи чи проекту. Дійсно, така робота є дуже ефективною. Але для некомп'ютерних спеціальностей курсова робота не передбачена програмою. Вирішити це протиріччя можна, якщо самі практичні заняття побудувати таким чином, щоб всі вони в сукупності були присвячені вирішенню якоїсь одної задачі. Для цього потрібно

перед кожним студентом з самого початку занять поставити завдання, яке було б важко або неможливо вирішити традиційними засобами, яке було б максимально наближене до реальних задач і вирішення якого вимагало б від студента застосування всіх знань з програмування, що він буде поступово набувати в процесі навчання. По мірі того, як на лекційних заняттях студент познайомиться з тою чи іншою програмною конструкцією, на практичних він зможе використати отриманні знання. Таким чином поступово наприкінці курсу, наче з цеглинок, складаючи свою програму, він отримає повноцінний програмний продукт.

Де ж знайти таку задачу? Найкраще використати для цього задачі, що відповідають основній спеціальності студента. Якщо це студент гірничої спеціальності, то слід підібрати відповідні завдання з галузі відкритої чи закритої розробки, якщо економічної – задачу з економічними розрахунками тощо. Бажано, щоб ці завдання включали в себе математичні розрахунки, можливі розгалуження, обробку табличних величин, потребували б побудови графіків, роботи з файлами та пристроями друку. Це може бути частиною їх курсового проекту, що вони виконують або будуть виконувати. В будь-якому випадку варто порадитися з викладачами фахових дисциплін і разом з ними підібрати індивідуальні чи групові завдання студентам. Враховувати, звичайно, потрібно при цьому і різний рівень їх підготовки. Наприклад, для слабших підібрати ряд нескладних типових задач, а найбільш складні задачі дати одному чи декільком, в залежності від задачі, добре підготовленим студентам.

В цілому організація роботи зі студентами некомп'ютерних спеціальностей може відбуватися таким чином:

- Близько третини часу, відведеного на вивчення обчислювальної техніки можна присвятити загальному ознайомленню з обчислювальною машиною: описати апаратну та програмну складову обчислювальної машини.
- При вивченні програмного забезпечення ЕОМ важливо показати розмаїття існуючих програм, подаючи при цьому їх порівняльну характеристику. Це дасть студентам більш широкий погляд на комп'ютерну техніку взагалі, а також буде стимулювати їх пізнавальну діяльність.
- Те саме бажано робити, описуючи інші види програмного за-

безпечення: текстові та графічні редактори, електронні таблиці тощо.

- Важливо в кожному окремому випадку показувати головні спільні риси різних програм, а освоєння нюансів залишити на практичні заняття.
- На практичних заняттях студенти, звичайно, більш глибоко зможуть ознайомитися з якимось одним представником з класу програм, що вивчаються (операційною системою, текстовим або графічним редактором, електронною таблицею, СУБД та ін.) Задачі, що повинні вирішуватися за допомогою даного програмного забезпечення також бажано підбирати найбільш наближеними до реальних задач, що можуть постати перед студентом в його професійній діяльності.
- Інший же час, не менше двох третин, слід приділити основам програмування і на самому початку видати студентам завдання, щоб допомогти їм в подальшій роботі. З цією метою можна дати їм приблизний план їх виконання, в якому кожний пункт буде відповідати окремому практичному заняттю, на якому студент буде застосовувати той чи інший засвоєний на лекції та під час самостійної роботи матеріал. Орієнтовний план може мати такий вигляд:

№	Тема теоретичних занять	Завдання на практичну роботу
1	Поняття про алгоритми. Способи описання алгоритмів.	Побудова загального алгоритму вирішення поставленої задачі та окремих її частин.
2	Особливості мови програмування. Типи даних.	Визначення з кількістю даних, що будуть використані в проекті та з їх типами.
3	Лінійні алгоритми.	Реалізація арифметичних розрахунків та інших частин проекту, що не мають розгалуження.
4	Алгоритми з розгалуженням.	Реалізація тих частин програми, що мають розгалуження.
5	Алгоритми з повторенням.	Реалізація циклічних фрагментів програми, наприклад обробка табличних даних.

6	Текстовий режим дисплея. Його можливості. Бібліотечні функції роботи з дисплеєм.	Побудова інтерфейсу програми.
7	Керування клавіатурою.	Удосконалення інтерфейсу програми. Побудова меню.
8	Графічний режим дисплея. Його можливості. Бібліотечні функції.	Реалізація графічної частини програми: демонстрація діаграм, графіків, креслень тощо.
9	Робота з файлами.	Організація в програмі роботи з файлами: зчитування початкових даних та збереження результатів.
10	Робота з пристроями друку.	Забезпечення можливості надрукувати отриманні результати роботи програми.

Кожен з цих пунктів може бути розділений на підпункти, бо на вивчення майже кожної теми потрібно декілька і практичних, і теоретичних занять. Кількість основних пунктів, а також їх послідовність теж можуть варіюватись. Головне, щоб все, що було отримано студентом на теоретичних заняттях, реалізовувалось в його практичній роботі. І робота ця сприймалась як одне ціле. Тоді наприкінці курсу студент отримає створений його власними руками завершений працездатний програмний продукт, який, крім того, що покаже засвоєнні ним знання з обчислювальної техніки, допоможе йому і, можливо, іншим учням при написанні курсових та дипломних робіт за фахом. При ретельному виборі завдань, не виключено, що таким чином можна забезпечити комп'ютерний клас вузу багатим асортиментом програм для розрахунку типових курсових та дипломних проектів тощо.

Звичайно такий підхід до проведення практичних занять вимагає від викладача більших затрат і часу, і сил на підготовку завдань до практичних робіт, та їх перевірку, а також є творчого співробітництва з іншими викладачами, що теж не так легко буває. Але всі ці зусилля будуть виправдані в процесі підготовки нашими вищими навчальними закладами спеціалістів високої кваліфікації.

ФАКУЛЬТАТИВНИЙ КУРС «КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ З ФІЗИКИ»

І.О. Теплицький¹, О.І. Теплицький²

¹ м. Кривий Ріг, Криворізький державний педагогічний
університет

² м. Кривий Ріг, Центрально-Міська гімназія

Сьогодні вже не потребує доведення той очевидний факт, що підвищення результативності вивчення всіх без винятку шкільних навчальних дисциплін вбачається у систематичному застосуванні засобів електронно-обчислювальної техніки (ЕОТ).

Аналіз педагогічних програмних засобів з фізики показує, що більшість із них повністю або частково являють собою програмно реалізовані заміни реальних об'єктів, які дозволяють всебічно відобразити найбільш суттєві властивості досліджуваних явищ, тобто є комп'ютерними моделями.

В Центрально-Міській гімназії м. Кривого Рогу протягом останніх восьми років з учнями 9–11 класів вивчається факультативний курс основ комп'ютерного моделювання. Він передбачає ознайомлення школярів з основними принципами побудови і дослідження математичних моделей, навчання найбільш розповсюджених прийомів і методів такої роботи, формування культури ведення дослідницької діяльності засобами ЕОТ. Навчальний матеріал включає широкий спектр задач з фізики.

У якості середовища для моделювання використовуються електронні таблиці (ЕТ), які дозволяють виводити на екран результати досліджень у вигляді таблиць і дають змогу швидко й легко одержувати графіки залежностей між характеристиками досліджуваного об'єкта. Мова ЕТ хоч і є штучною, але вона найбільш проста з усіх штучних. Перехід до ЕТ, значно спрощуючи процедуру підготовки задачі до її розв'язування за допомогою комп'ютера, дозволяє з прийнятними витратами часу і мінімальними зусиллями провести необхідні підготовчі етапи через свідомість учнів. До того ж, за чинною програмою з інформатики ознайомлення з прийомами роботи в ЕТ значно випереджає у часі вивчення програмування, що дозволяє розпочати

курс моделювання раніше за програмування і незалежно від нього [4].

Поданий нижче матеріал на прикладі розв'язування задачі оптимізації ілюструє нашу методику. Сюжетною основою моделювання тут є відома задача про «м'яку посадку» на Місяць, що свого часу (у 80-і роки) обійшла сторінки таких відомих науково-популярних журналів, як «Наука и жизнь» і «Техника – молодёжи» [3] і заслужено потрапила до пробного навчального посібника [1, С. 101–119]. Ми вважаємо цю задачу-модель цінною методичною знахідкою, яка не повинна загубитися у часі та просторі. Робота з нею настільки емоційна, що коли перед тим, як запропонувати її учням для моделювання, дати їм можливість попрацювати з готовою моделлю, то надійно забезпечується висока мотивація їхньої подальшої навчальної діяльності.

Ця задача завершує розділ «Моделювання механічних рухів» нашого курсу. Якщо у даній моделі пошук оптимальної стратегії управління здійснюється шляхом спроб та помилок, то у наступному розділі розглядаються приклади автоматизованого розв'язування задач оптимізації і, отже, ця задача є перехідною.

І. Постановка задачі. З метою доставки і розміщення на поверхні Місяця апаратури для наукових досліджень планується запуск космічного корабля. Схема посадки передбачає декілька послідовних фаз: 1) виведення корабля з корисним вантажем на колову селеноцентричну орбіту; 2) відділення від корабля вантажу і перетворення його на самостійний орбітальний модуль; 3) здійснення м'якої посадки модуля на поверхню Місяця.

Заключна фаза має пройти у два етапи. Спочатку за допомогою гальмівних двигунів відбудеться гасіння швидкості модуля, внаслідок чого він зійде з колової орбіти і у певний момент на короткий час зависне на деякій висоті h_0 над поверхнею Місяця. Щоб уникнути подальшого вільного падіння, в цей момент увімкнуться двигуни м'якої посадки, що й забезпечить його плавний спуск до поверхні.

Завершенням посадки вважатиметься момент, коли значення координати h переходитиме через нуль. При цьому м'якою вважатиметься така посадка, коли швидкість апарату при його дотику з ґрунтом не перебільшуватиме 1 м/с. Передбачається, що

швидкість апарату під час опускання можна регулювати, змінюючи витрату пального, яке подається до камер згоряння, оскільки реактивна сила тяги двигунів залежить від маси пального, що згоряє в камерах.

Крім того, слід врахувати таке: а) запас пального, призначеного для роботи двигунів м'якої посадки, обмежений; б) щоб запобігти руйнуванню цих двигунів під час роботи, витрата пального за одиницю часу не може перебільшувати деякого максимально дозведеного значення; в) при здійсненні посадки швидкість витікання реактивного струменя через сопло підтримується сталою; г) сталою також залишається і маса модуля без урахування пального.

Задача полягає в тому, щоб виявити такий режим подавання пального до камер згоряння, тобто таке управління, яке забезпечуватиме м'яку посадку.

За цілком зрозумілих причин конструктори не можуть провести серію реальних експериментів. Єдиний придатний спосіб знаходження необхідного управління полягає в тому, щоб підготувати й провести відповідний *обчислювальний експеримент* на ЕОМ на основі математичної моделі заключного етапу операції.

II. Побудова моделі. Побудову моделі почнемо з *формалізації* задачі.

Введемо позначення:

$m=m(t)$ – змінна маса пального. У початковий момент ($t=0$) $m=m_0$.

r – витрата пального (маса пального, що спалюється в камерах згоряння двигунів м'якої посадки за проміжок часу від моменту t до $t+\Delta t$). Оскільки припускається, що витрату можна регулювати в ході зниження, то r є функцією часу: $r=r(t)$. Зазначене вище обмеження на витрату пального можна записати у вигляді нерівності $0 \leq r(t) \leq r_{max}$.

$v = v(t)$ – миттєва швидкість модуля під час опускання. Початкова умова «зависання» виражається рівністю $v(0)=0$.

u – швидкість витікання реактивного струменя (продуктів згоряння) через сопло.

$h=h(t)$ – висота над поверхнею Місяця, причому $h(0)=h_0$.

Решту позначень будемо вводити до розгляду в міру необхідності.

III. Фізичний аналіз і попереднє обговорення плану майбутньої роботи. Рух, що його ми збираємось дослідити, з фізичної точки зору являє собою неперервний у часі й просторі процес. Дійсно, тут неперервно відбуваються зміни значень всіх динамічних і кінематичних характеристик рухомого об'єкту: маси, діючих сил, прискорення, швидкості, імпульсу, координати тощо. Але, з іншого боку, доцільно пригадати, що математичні моделі, призначені для опрацювання за допомогою комп'ютера, повинні бути дискретними, оскільки сам комп'ютер є дискретним пристроєм, тобто виконує операції за окремими кроками. Ця обставина зумовлює необхідність у переході від неперервних моделей до дискретних.

Пошук розв'язків при опрацюванні дискретних моделей здійснюється на основі *чисельних* методів шляхом переходу від неперервної моделі до рівнянь, записаних у вигляді різницевих схем. З попереднього досвіду ми вже знаємо, що для дискретизації неперервного в часі процесу весь час руху подрібнюють на окремі достатньо малі інтервали Δt . При цьому вважають, що протягом такого інтервалу характеристики руху залишаються постійними, а їхні зміни відбуваються миттєво (стрибоподібно) в моменти часу, що відповідають або початку, або кінцю, або ж середині кожного такого інтервалу. Ці моменти, а також тривалість проміжку Δt обирає дослідник. При послідовному зменшенні Δt результати, одержані на дискретній моделі, наближаються до результатів, що їх дає неперервна модель.

Модель, яку ми будуватимемо, заснована на перших принципах: законах Ньютона і законах збереження.

З метою спрощення подальших міркувань приймемо такі припущення.

Припущення 1. Протягом проміжку часу Δt аж до його останнього моменту всі характеристики руху залишаються незмінними. У останній момент *миттєво* спалюється пальне маюсою r , що призводить до одночасної зміни значень згаданих характеристик. Ці нові значення стають початковими для наступного інтервалу Δt , оскільки кінець даного інтервалу є початком наступного.

Розглянемо окремі елементарні моделі, в результаті об'єднання яких буде побудована математична загальна модель м'якої посадки модуля на поверхню Місяця (рис. 1).

1 Модель вигорання пального.

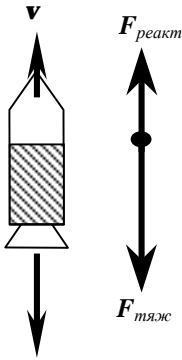
$$m(t) = m(t - \Delta t) - r(t - \Delta t) \cdot \Delta t, \quad (1)$$

де $0 \leq r(t) \leq r_{\max}$.

2. Модель гравітаційного притягання Місяця.

$$F_{\text{тяж}} = (m(t) + M) \cdot g \quad (2)$$

g – прискорення сили тяжіння поблизу поверхні небесного тіла. Для Місяця, зокрема, $g = 1,62 \text{ м/с}^2$.



3. Модель реактивної сили тяги.

Реактивна сила тяги $F_{\text{реакт}}$, що діє на модуль, дорівнює швидкості зміни імпульсу модуля:

$$F_{\text{реакт}} = \frac{\Delta((M + m(t))v)}{\Delta t}$$

Згідно закону збереження імпульсу для замкнутої системи тіл «модуль – витікаюча струмина» зміна імпульсу модуля за абсолютною величиною дорівнює імпульсу струмини:

$$\Delta(M + m(t)) \cdot v = m(t) \cdot u.$$

Таким чином,

$$F_{\text{реакт}} = \frac{u \cdot r(t)}{\Delta t}. \quad (3)$$

4. Модель руху модуля.

Внаслідок відсутності на Місяці атмосфери силу опору руху з боку середовища можна не включати до розгляду. Тому на активній ділянці спуску, коли працюють двигуни м'якої посадки, на модуль діють лише дві сили: сила гравітаційного притягання Місяця $F_{\text{тяж}}$ і протилежно напрямлена реактивна сила тяги $F_{\text{реакт}}$. За другим законом Ньютона

$$F_{\text{тяж}} - F_{\text{реакт}} = (m(t) + M) \cdot a,$$

де $a = a(t)$ – прискорення апарату (є функцією часу).

Підставляючи (2) і (3) в останній вираз, одержуємо

$$a(t) = \frac{u \cdot r(t - \Delta t)}{(m(t) + M) \cdot \Delta t} - g. \quad (4)$$

Оскільки розглядається рух тіла змінної маси, то прискорення виявляється складною функцією часу.

Припущення 2. Прийmemo наближено, що прискорення $a(t)$ залишається сталим на кожному малому інтервалі Δt , тобто рух апарату на кожному часовому інтервалі є рівноприскореним. Тоді миттєва швидкість модуля наприкінці інтервалу

$$\mathbf{v}(t) = \mathbf{v}(t - \Delta t) + a(t) \cdot \Delta t. \quad (5)$$

У свою чергу, висота $h(t)$ над поверхнею Місяця може бути задана виразом

$$h(t) = h(t - \Delta t) + \mathbf{v}(t) \cdot \Delta t + \frac{a(t) \cdot \Delta t^2}{2}. \quad (6)$$

Рівняння (1), (4), (5), (6) являють собою математичну модель м'якої посадки на Місяць.

Суттєвого спрощення подальшої роботи можна досягти, прийнявши тривалість інтервалу Δt рівною 1 с. Проте платнею за таке спрощення стане погіршення (огрублення) точності обчислень. На якісному рівні, однак, модель залишатиметься задовільною з урахуванням сказаного запишемо систему:

$$\begin{cases} m(t) = m(t - 1) - r(t - 1) \\ a(t) = \frac{u \cdot r(t - 1)}{m(t) + M} - g \\ \mathbf{v}(t) = \mathbf{v}(t - 1) + a(t) \\ h(t) = h(t - 1) + \mathbf{v}(t - 1) + \frac{a(t)}{2} \end{cases} \quad (7)$$

Цю систему рівнянь відносно $m(t)$, $a(t)$, $\mathbf{v}(t)$, $h(t)$ (при $t=0, 1, 2, \dots$) будемо розв'язувати *методом послідовних підстановок*. Суть методу полягає у такому. Нехай обрано деяке припустиме управління космічним апаратом, тобто є заданою послідовність значень $r(0)$, $r(1)$, \dots , $r(t)$, така, що $r(0) + r(1) + \dots + r(t) \leq m_0$. При $t=0$ нам вже відомі значення $m(0)$, $a(0)$, $\mathbf{v}(0)$, $h(0)$ і, отже, можна обчислити значення правих частин у рівняннях системи (7), тобто знайти $m(1)$, $a(1)$, $\mathbf{v}(1)$, $h(1)$. У свою чергу, їх можна використати для обчислення значень $m(2)$, $a(2)$, $\mathbf{v}(2)$, $h(2)$ і т.д. Процес обчислень повинен припинитися, як тільки виконається умова $h(t + \Delta t) \leq 0$, тобто буде досягнуто по-

верхні. Якщо при цьому одночасно виявиться, що $-1 \leq v(t+\Delta t) \leq 0$, то посадку будемо вважати «м'якою».

Система рівнянь (7) є *рекурентною*. Алгоритм її розв'язування передбачає виконання однотипних обчислень при $t=0, t=1, \dots$. Зрозуміло, що з подібною одноманітною роботою ЕОМ справляється краще за людину.

На сучасних ЕОМ розв'язуються системи рекурентних рівнянь з багатьма невідомими, що виникають при математичному моделюванні реальних процесів і явищ.

IV. Комп'ютерний експеримент.

1. Підготовча частина.

Створимо електронну таблицю, перший рядок якої заповнимо заголовками стовпців, стовпець G («Дано:») – іменами змінних, стовпець H – їх початковими значеннями, а комірки A2 – F2 другого рядка (для початкового моменту часу $t=0$) – посиланнями на відповідні комірки H2 – H8 (рис. 1):

	A	B	C	D	E	F	G	H
1	<i>t</i>	<i>m</i>	<i>a</i>	<i>v</i>	<i>h</i>	<i>r</i>	Дано:	
2	0	100	-1,62	0,00	10,00	0	$m_0, \text{кг} =$	100
3							$g, \text{м/с}^2 =$	1,62
4							$v_0, \text{м/с} =$	0
5							$u, \text{м/с} =$	200
6							$M, \text{кг} =$	1000
7							$h_0, \text{м} =$	10
8							$r_{max}, \text{кг} =$	10
...	

Рис. 1.

2. Заповнимо третій рядок відповідними формулами.

Комірка	Формули / числа
A3	=A2+1
B3	=B2-F2
C3	=\$H\$5*F2/(B3+\$H\$6)-\$H\$3
D3	=D2+C3
E3	=E2+D2+C3/2
F3-Fx	Заповнюються з клавіатури

3. Скопіюємо формули третього рядка у відповідні стовпці. Таблиця набуває вигляду, зображеного на рис. 2:

	A	B	C	D	E	F	G	H
1	<i>t</i>	<i>m</i>	<i>a</i>	<i>v</i>	<i>h</i>	<i>r</i>	<i>Дано:</i>	
2	0	100	-1,62	0,00	10,00	0	<i>m</i> ₀ , кг=	100
3	1	100	-1,62	-1,62	8,38		<i>g</i> , м/с ² =	1,62
4	2	100	-1,62	-3,24	5,14		<i>v</i> ₀ , м/с=	0
5	3	100	-1,62	-4,86	0,28		<i>u</i> , м/с=	200
6	4	100	-1,62	-6,48	-6,20		<i>M</i> , кг=	1000
...	<i>h</i> ₀ , м=	...
							<i>r</i> _{max} , кг=	
							...	

Рис. 2.

До речі, огляд таблиці показує, що коли не працюють двигуни м'якої посадки ($r=0$, $m=100$), модуль вільно падає, і вже наприкінці четвертої секунди в аварійному режимі ($h=-2,15$ с, $v=-2,15$ м/с) досягає поверхні. Наступні значення величин швидкості (після D6) і висоти (після E6) являють собою результати рахунку за введеними формулами і не відображують реальний процес.

2. Обчислювальний експеримент

Будемо послідовно заповнювати стовпець F довільними значеннями змінної r – витрати пального на проміжках часу $\Delta t=1$ с. Після заповнення кожної чергової комірки спостерігаємо за відповідними значеннями у комірках наступного рядка для стовпців D (швидкість) та E (висота) і з'ясовуємо, чи не відбулася м'яка посадка. Якщо при деякому черговому значенні r одночасно виконуватимуться умови $h<0$ й $-1\leq v\leq 0$, то задачу вважаємо розв'язаною і припиняємо обчислювальний експеримент.

Виявляється, що «здійснити» м'яку посадку – зовсім не проста справа. Вона ще більше ускладнюється, коли додається умова зекономити при цьому якомога більше пального.

На рис. 4 подано одне з можливих вдалих управлінь (послідовність значень щосекундної витрати пального r) при $m_0=100$ кг, що забезпечує економію пального у 44 кг.

<i>t</i>	0	1	2	3	4	5	6	7
<i>r</i>	0	6	10	10	10	10	10	
<i>v</i>	0	-1.62	-2.14	-1.92	-1.68	-1.42	-1.14	-0.84
<i>h</i>	10	9.19	7.31	5.28	3.48	1.93	0.66	-0.33

Рис. 4.

V. Поліпшення інтерфейсу користувача

Створена таблиця має чимало суттєвих вад.

1. Насамперед, у ній відсутня текстова інформація про результат процесу посадки після кожного управління. Цю інформацію ми поки що вимушені отримувати шляхом безпосереднього розгляду значень швидкості й висоти і порівнянням їх з умовою м'якої посадки. З метою автоматизації цієї справи виконаємо таке:

- після стовпця F створимо новий стовпець, який одержить ім'я G, а стовпець «Дано:» і наступний – відповідно H і I;

- у комірці G1 запишемо «Процес»;

- введемо до комірки G3 формулу

=ЕСЛИ(И(D3>=-1;D3<=0;E3<=0);"Ok!";ЕСЛИ(И(D3<-1;E3<0);

"Bad";"")),

яку скопіюємо у комірки (G4 – G12).

Завдання. Прокоментуйте запропоновану формулу.

2. Згідно умови в моделі існує обмеження $0 \leq r(t) \leq r_{max}$, яке поки що не враховане. Реалізуємо його шляхом введення до комірки H9 такої формули

=ЕСЛИ(ИЛИ(F3>\$I\$8;F4>\$I\$8;B5>\$I\$8;F6>\$I\$8;F7>\$I\$8;F8>\$I\$8;F9>\$I\$8;F10>\$I\$8;F11>\$I\$8;F12>\$I\$8);"Дивись I8 !";"").

Тепер при введенні до будь-якої з комірок B3–B13 значень, що перебільшують вміст комірки I8 (обмеження на r), у H9 матимемо повідомлення "Дивись I8 !". У випадку допустимих значень щосекундної витрати комірка H9 буде порожньою.

Завдання. Доопрацюйте наведену формулу так, щоб у ній одночасно враховувалася умова $r \geq 0$.

3. Оскільки початковий запас пального m_0 для двигунів м'якої посадки є обмеженим (комірка I2), то бажано також виводити на екран повідомлення про закінчення пального. Таке повідомлення можна здійснити, наприклад, у комірці H10 за допомогою формули

=ЕСЛИ(СУММ(F2:F12)>=\$I\$2;"ЗАКІНЧИЛОСЯ ПАЛЬНЕ";"").

Завдання. Запропонуйте інші можливі варіанти такої перевірки і порівняйте їх з наведеним.

Перевірте нові можливості таблиці шляхом тестування за пп. 1–3.

З урахуванням всіх виконаних змін попередня таблиця набуває вигляду, зображеного на рис. 5. Тут видно, що при $m_0=70$ кг і управліннях $r=0, 8, 9, 9, 10, 10, 10, 8, 6$ пальне закінчується в останній момент вдалої посадки.

	A	B	C	D	E	F	G	H	I	G	K
1	t	m	a	v	h	r	Процес	Дано:			
2	0	70	-1,62	0,00	10,00	0		$m_0=70$			
3	1	70	-1,62	-1,62	9,19	8		$g_{mic}=1,62$			
4	2	62	-0,11	-1,73	7,51	9		$v_0=0$			
5	3	53	0,09	-1,64	5,82	9		$u=200$			
6	4	44	0,10	-1,54	4,23	10		$M=1000$			
7	5	34	0,31	-1,23	2,85	10		$h_0=10$			
8	6	24	0,33	-0,89	1,79	10		$r_{max}=10$			
9	7	14	0,35	-0,54	1,07	8					
10	8	6	-0,03	-0,57	0,52	6		Закінчилося пальне !			
11	9	0	-0,42	-0,99	-0,26		Ок!				

Рис. 5.

4. Наступним суттєвим покращенням інтерфейсу користувача є можливість графічного відображення залежностей змінних величин від часу. Як відомо, така задача легко реалізується засобами електронних таблиць. На рис. 6 маємо графіки залежності від часу основних характеристик процесу у відповідності до рис. 5.

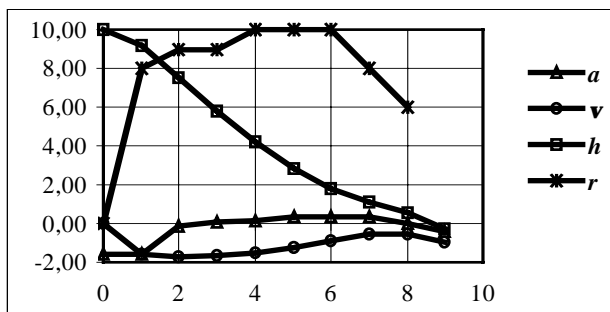


Рис. 6.

VI. Висновки.

1. Розглянута модель створює підстави для важливої розмови про сучасні підходи до розв'язування задач оптимізації. Сьо-

годні такі задачі розв'язуються на основі добре розроблених математичних методів (варіаційне числення, лінійне та динамічне програмування, дослідження операцій тощо), які забезпечують автоматизований пошук оптимальних режимів управління. Деякі з цих підходів мають надійні комп'ютерні реалізації. Як вже зазначалося вище, ознайомлення з ними становить зміст окремого розділу нашого курсу.

2. Значний обсяг досліджень останніх десятиліть переконує, що шкільна фізика належить до тих навчальних дисциплін, де використання нових інформаційних технологій є цілком виправданим і необхідним.

3. Важко переоцінити роль комп'ютерного моделювання при вивченні фізики, оскільки воно сприяє виокремленню самої суті явищ, розвитку науково-теоретичного мислення. Однак захоплення використанням готових моделей погрожує передчасним розривом зв'язку виучуваного явища з дійсністю. Це трапляється, коли учням пропонують працювати з готовими моделями, не розкриваючи процесу їх створення. Оскільки об'єктами вивчення повинні залишатися реальні явища, то підміна їх абстрактними поняттями й символами при недостатній базі спостережень і досвіду нерідко веде до згубного формалізму, коли за удаваними знаннями губиться їх сутність [2].

Література:

1. Верлань А.Ф., Распопов В.Б. Основы применения вычислительной техники: Пробное учебн. пособие для 10 кл. ср. шк. – К: Рад. шк., 1986.– 160 с.
2. Пухов М. Истинная правда // Техника – молодежи. – 1985. – №6. – С. 52–57.
3. Разумовский В.Г. ЭВМ и школа: Научно-педагогическое обеспечение // Сов. педагогика. – 1985. – № 9. – С. 12–16.
4. Теплицький І.О. Використання електронних таблиць у комп'ютерному моделюванні // Комп'ютер у школі та сім'ї. – 1999. – № 2. – С. 27–32.

ЭЛЕМЕНТЫ МОДЕЛИРОВАНИЯ НА УРОКАХ ИНФОРМАТИКИ В СПЕЦИАЛИЗИРОВАННЫХ КЛАССАХ

С.А. Томилин

г. Кривой Рог, Криворожский государственный педагогический университет

Умение планировать свои действия, предвидеть их результаты и отвечать за них – всегда было присуще человеку, находящемуся в поисках истины. Проблема поиска в последнее время облегчается использованием персональных компьютеров (ПК), и развитие науки требует прогресса машинных возможностей до нового, более высокого уровня, а также преодоления психологических и организационных барьеров, что является гарантом экономии многих месяцев и лет экспериментирования, для умеющего разговаривать с машиной на равных.

На современном этапе развития общества происходит компьютеризация практически всех отраслей человеческой деятельности. Поэтому выпускник массовой школы должен иметь хотя бы первичные навыки общения с машинами: одним вполне достаточно научиться правильно пользоваться ПК, другим же для будущей деятельности необходимо овладеть первичными знаниями и практикой составления программ на языках разного класса. Но предварительно, именно в средней школе должны быть заложены в человека основы общения с машиной и умение материального и логического моделирования [1].

В настоящее время моделирование привлекает пристальное внимание всех наук и получило необычайно широкое применение во многих областях знаний: от всевозможных гуманитарных разделов знаний до отраслей знаний точных наук, от проблем механики и ядерной физики до проблем социологии, психологии, педагогики и т.д. Современное моделирование превратилось:

1. В общенаучный, в высшей степени эффективный инструмент познания.
2. В метод прогнозирования всевозможных научных и методических разработок.
3. В метод машинной имитации долгосрочных программ и

планов в области той или иной науки, анализа и оценки различных вариантов принимаемых ответственных решений и последствий их реализации и др.

В общем случае процесс моделирования состоит из следующих этапов:

1. Постановка задачи и определение свойств оригинала, подлежащих исследованию.
2. Выбор модели, достаточно точно определяющей и хорошо фиксирующей существенные свойства оригинала и легко поддающийся исследованию.
3. Исследование модели в соответствии с поставленной задачей и выбором методов исследования.
4. Перенос результатов исследования модели на оригинал.
5. Проверка и анализ полученных результатов.

Многие области математики или других наук можно привлечь к построению модели определенного круга задач. Для задач, числовых по своей природе, можно построить модели на основе общих математических конструкций, таких как системы линейных уравнений (например, для задач расчета электрических цепей), дифференциальные уравнения (задачи прогноза роста популяций и др.). Для задач с символьными или текстовыми данными можно применить модели символьных последовательностей и формальных грамматик. Решение таких задач содержит вышеназванные этапы.

Основные этапы реализуются не только при моделировании естественнонаучных процессов в научных исследованиях, но и постоянно должны использоваться на уроках информатики при решении разного рода задач, что придает функциональность школьному курсу информатики, что особенно актуально для специализированных классов математического профиля. От решения простых «искусственных» задач (основных алгоритмов), далеких от реальных процессов, но являющихся базовыми в будущем, до более сложных, неотъемлемо связанных с окружающим миром, при решении которых и применяется моделирование с использованием алгоритмов, изучаемых ранее – таков основной путь внедрения моделирования в школьный курс основ информатики.

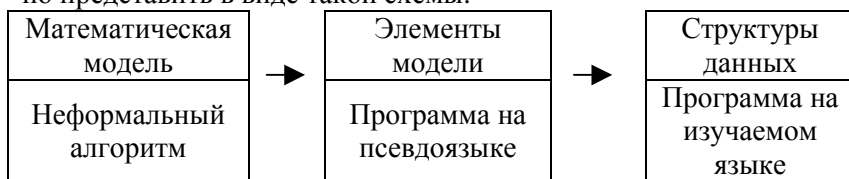
Примером задачи моделирования на первом этапе обучения

этому приему, может быть задание по моделированию определенной эволюционной схемы развития популяций. Оно разбивается на следующие этапы.

1. Постановка проблемы: рассмотреть основные процессы зарождения, жизни, и гибели отдельных популяций.

2. Выбор модели. Биологическое воспроизведение моделируется с использованием игры «Жизнь» (Кембриджский университет, Джон Хортон Конвей, 1970), которая имеет достаточно простую интерпретацию и возможности сравнивать и анализировать полученные данные с данными оригинала. Выбранная модель наглядно иллюстрирует природные биологические процессы [3, 4].

3. Поэтапное программирование. Сам же процесс программирования, т.е. написания и отладки конечной программы, можно представить в виде такой схемы:



На первом этапе строится модель исходной задачи, для чего привлекаются соответствующие подходящие математические модели. На этом этапе для нахождения решения создается неформальный алгоритм. Неформальным алгоритмом в рассматриваемом примере есть правила появления нового поколения из старого. На следующем этапе алгоритм решения записывается на псевдоязыке, например в виде блок-схемы или в смеси языка программирования со вставками на «человеческом» языке. Программа на псевдоязыке должна быть достаточно подробно описанной, чтобы в ней были зафиксированы и отражались все необходимые элементы (в последующем типы данных) над которыми выполняются операции. Третий этап процесса программирования является заключительным, в нем создается и отлаживается алгоритм созданный на предыдущих этапах посредством изучаемого языка программирования.

Основными затруднениями, которые возникают в процессе моделирования являются: во-первых, выбор моделей и, во-вторых, перенос результатов исследования моделей на оригинал.

Поэтому, школьники обычно работают только с моделями, уже созданными учителем или авторами книг. В таком варианте снижается интерес к предмету и уменьшается функциональность знаний, полученных в школьном курсе информатики.

Моделирование содействует развитию творческих способностей учащихся, т.к. учит поиску способов проверки гипотезы. Возможно, что в процессе моделирования получены данные, которые не характерны изучаемому оригиналу, или же просто не отражают свойства и явления исследуемого объекта. Значит, или модель реализована с ошибками или же она неадекватна имеющемуся оригиналу, тогда задача может принять циклический характер, т.е. необходимо вернуться к пересмотру построенной модели или к другим методам ее исследования. Таким образом, учащиеся приобретают первые навыки исследования.

Практика преподавания школьного курса информатики подтверждает, что методическими условиями успешного использования моделирования как основного фактора функциональности знаний являются:

- пропедевтика элементов моделирования в основных ранее изучаемых школьных курсах;
- обучение оптимальному выбору модели, фиксирующей основные свойства оригинала;
- регулярное использование приемов анализа, сравнения и синтеза модели и оригинала;
- творческое использование сформированных умений моделирования в новых или измененных ситуациях.

Литература.

1. Лебедев А.Н. Моделирование в научно-технических исследованиях. – М.: Радио и связь, 1989. – 224 с.
2. Леонова Н.А., Соловьев В.Н. Формирование научного мировоззрения средствами математического моделирования // Комп'ютерне моделювання та інформаційні технології в природничих науках: Зб. наук. праць. – Кривий Ріг: Видавничий відділ КДПУ, 2000. – с. 153–159.
3. Теплицький І.А. Комп'ютерне моделювання в школі як засіб розвитку творчого мислення учнів // Рідна школа. – 2000. – №9. – с. 63 – 66.

МЕТОДИКА ПОДГОТОВКИ УЧАСТНИКОВ ОЛИМПИАД ПО ИНФОРМАТИКЕ

Ю.В. Филатов

г. Кривой Рог, Криворожский государственный педагогический университет

Олимпиады по информатике традиционно считаются наиболее престижным очным турниром программистов среди учащихся средних учебных заведений. С удовлетворением можно отметить, что сложность задач предлагаемых на международных, всеукраинских, областных (Днепропетровская обл.) и городских (г. Кривой Рог) повышается с каждым годом. Причем возрастают требования не только к навыкам участников в области профессионального программирования, но и, самое главное, к их теоретической подготовке.

Естественно, что с ростом требований к участникам олимпиад растут и требования к методике их подготовки. Работа в течение ряда лет с командой юных программистов Саксаганского естественнонаучного лицея г. Кривого Рога, участие в жюри городских и областных олимпиад по информатике позволила автору статьи создать и апробировать собственную методику подготовки одаренных детей к участию в олимпиадах. Эта методика, безусловно, не является единственно правильной, однако ее использование дает стабильные и достаточно высокие результаты. Кроме того, различные разделы этой методики могут изменяться применительно к контингенту учащихся, квалификации преподавателя и количеству часов, отводимых на подготовку участников олимпиад.

Программа подготовки включает в себя две основные части:

1. Общая подготовка к участию в олимпиадах.
2. Специальная подготовка к участию в олимпиадах.

Общая подготовка предполагает теоретический и практический разделы следующего содержания:

ОБЩАЯ ТЕОРЕТИЧЕСКАЯ ПОДГОТОВКА.

Теоретические основы программирования. Необходимо рассмотреть темы: элементы теории алгоритмов; основы структурного программирования.

Эффективность программ. Необходимо рассмотреть темы: вычислительная сложность алгоритма и оценка времени работы программы; структурная оптимизация; эффективность программного кода.

Стиль программирования. Необходимо с самого начала прививать определенный стиль написания программ, рассматривать типовые для начинающего программиста ошибки.

ОБЩАЯ ПРАКТИЧЕСКАЯ ПОДГОТОВКА.

Изучение языка программирования. Необходимо излагать основы избранного процедурного языка программирования в соответствии с теорией структурного программирования. Особое внимание следует уделять использованию базовых структур, применению процедур и функций, реализации различных типов данных, в т.ч. конструируемых.

Общие методы программной обработки данных. Крайне важный этап обучения. Необходимо дать понятие о целесообразности применения таких методов, как: итерация, рекурсия, дихотомия, а также метода прогрессивных разделений и оценок (метод ветвей и границ) и динамического программирования.

Базовые алгоритмы решения стандартных подзадач. Парадигма процедурного программирования предполагает использование отработанных приемов. Необходимо научить безошибочно использовать алгоритмы выбора, обмена, поиска, сортировки, порождения комбинаторных объектов, обработки прямоугольных матриц.

Технические приемы программирования. Поскольку программы участников олимпиад тестируются в жестких временных рамках, необходимо научить программиста эффективному использованию, как аппаратных ресурсов компьютера, так и ресурсов избранного языка программирования.

Специальная подготовка также состоит из теоретического и практического разделов.

СПЕЦИАЛЬНАЯ ТЕОРЕТИЧЕСКАЯ ПОДГОТОВКА.

Методика решения задач. Автор опирается на собственную схему решения задач. Схема включает такие этапы: Анализ условия. Разработка проекта программы. Программная реализация решения. Отладка и тестирование программ. Анализ результатов работы. Подробнее работа по схеме описана в [4].

СПЕЦИАЛЬНАЯ ПРАКТИЧЕСКАЯ ПОДГОТОВКА.

Математические методы решения задач различных типов. Правильное определение типа задачи оказывает решающее влияние на весь ход решения. Поэтому рассмотрение различных алгоритмов и решение конкретных задач необходимо связывать с определенным типом задач и соответствующей теорией. Желательно рассмотреть, как минимум, такие разделы математической теории: теория чисел, численные методы, аналитическая геометрия, комбинаторика, теория множеств, теория графов, линейное программирование, теория расписаний, криптология.

Литература:

1. Ван Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ. – М.: Мир, 1985.
2. Кнут Д. Искусство программирования, т. 1–3. – М.: Мир, 1976–1978.
3. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы. – М: Издательский дом «Вильямс», 2000. – 384 с.
4. Филатов Ю.В. Решение задач повышенной сложности по информатике. – Видавничий відділ КДПУ / Збірник наукових праць, Кривий Ріг, 2000.

СРАВНЕНИЕ ДИСКРЕТНОГО И НЕПРЕРЫВНОГО ПОДХОДА К ЧИСЛЕННОМУ РЕШЕНИЮ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

А.А. Хараджян

г. Кременчуг, Кременчугский государственный политехнический университет

В настоящее время при моделировании динамических электромеханических систем очень мало внимания уделяется дискретным методам. Разработано много методов синтеза и анализа дискретных систем, однако все они ориентированы на использование в системах управления, а не в моделировании.

Численное решение системы дифуравнений обладает рядом недостатков: сложность вычислений; чувствительность к параметрам интегрирования (шаг интегрирования, точность и др.). В общем случае для нелинейной системы дифуравнений решение может стать неустойчивым, а смоделированный процесс – расходящимся, хотя реально он стремится к устойчивому состоянию.

Одним из способов решения данной проблемы является преобразование непрерывной системы в дискретную, и моделирование уже дискретной системы.

Рассмотрим оба пути численного решения системы дифуравнений. Для примера возьмем в качестве объекта моделирования двигатель постоянного тока (ДПТ), описываемый следующей системой уравнений:

$$u_{\alpha}(t) = R_{\alpha} i_{\alpha}(t) + L_{\alpha} \frac{di_{\alpha}(t)}{dt} + \omega(t) k \Phi(i_{\alpha}(t)),$$
$$i_{\alpha}(t) k \Phi(i_{\alpha}(t)) - M_c(t) = J \frac{d\omega(t)}{dt}.$$

1. Метод численного интегрирования Рунге-Кутты 4 порядка.

Если ввести следующие обозначения y_i – вектор состояния координат на текущем шаге, y_{i-1} – вектор состояния координат на предыдущем шаге, h – шаг интегрирования, $\frac{dy}{dt} = f(t, y)$, то можно записать следующие выражения:

$$y_i = y_{i+1} + \frac{h}{6}(f_1 + 2f_2 + 2f_3 + f_4), \text{ где}$$

$$f_1 = f(t_i, y_i), \quad f_2 = f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}f_1\right),$$

$$f_3 = f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}f_2\right), \quad f_4 = f(t_i + h, y_i + hf_3)$$

Отсюда видно, что для моделирования процесса необходимо как минимум четыре раза вычислить (возможно, численно) производную вектора состояния.

Для ДПТ вектор состояния будет иметь вид $y = \{i_{я}, \omega\}$, а производная вектора состояния:

$$\frac{dy}{dt} = \left\{ \begin{array}{l} di/dt \\ d\omega/dt \end{array} \right\} = \left\{ \begin{array}{l} \frac{1}{L_{я}}(u_{я}(t) - R_{я}i_{я}(t) - \omega(t)k\Phi(i_{г}(t))) \\ \frac{1}{J}(i_{я}(t)k\Phi(i_{г}(t)) - M_c(t)) \end{array} \right\}$$

Определим количество операций, необходимых для одного шага интегрирования: операций типа сложение–вычитание – 29, а операций умножения – 30.

Критерием выбора шага интегрирования является максимальный корень характеристического уравнения системы. Шаг интегрирования в значительной степени влияет устойчивость интегрирования и соответственно на точность.

2. В общем виде интегрирование дискретной системы состоит в преобразовании системы к системе разностных уравнений. Разностные уравнения строятся для каждой координаты на основе дискретного преобразования Лапласа или z -преобразования. Дальнейшие вычисления при интегрировании производятся по рекуррентным формулам. Если ПФ имеет вид

$$W(z) = \frac{a_n z^n + a_{n-1} z^{n-1} + \dots + a_0}{b_m z^m + b_{m-1} z^{m-1} + \dots + b_0} \text{ при } n \leq m, \text{ то значение выходной}$$

координаты в текущий момент времени можно вычислить по формуле:

$$y[Tk] = \frac{1}{b_m} (a_n x[T(k+n-m)] + \dots + a_0 x[T(k-m)] - b_{m-1} y[T(k-1)] - \dots - b_0 y[T(k-m)])$$

Для ДПТ дискретные передаточные функции будут иметь вид:

$$W_i(z) = \frac{I}{\Delta U} = \frac{(1 - e^{-T_\delta/T_\alpha})/R_\alpha}{z - e^{-T_\delta/T_\alpha}},$$

$$W_\omega(z) = \frac{\omega}{\Delta M} = \frac{T_\delta}{z - 1}.$$

Отсюда получаем разностные уравнения:

$$i_\alpha[T_\delta k] = (u[T_\delta k] - \omega[T_\delta k]k\Phi) \frac{1 - e^{-T_\delta/T_\alpha}}{R_\alpha} + i_\alpha[T_\delta(k-1)]e^{-T_\delta/T_\alpha},$$

$$\omega[T_\delta k] = (i_\alpha[T_\delta k]k\Phi - M_c) \frac{T_\delta}{J} + \omega[T_\delta(k-1)].$$

Как видно, в данном методе количество вычислительных операций на один шаг интегрирования значительно меньше и составляет: для операций типа сложение-вычитание – 4, а операций умножения – 5.

При дискретном методе шаг интегрирования влияет исключительно на точность, а устойчивость обеспечивается тем, что дискретное преобразование является аналитическим и, если исходная система была устойчивой, то и ее дискретное изображение будет устойчивым.

Период дискретности в обоих случаях можно определить исходя из теоремы Котельникова-Шеннона про частоту дискретизации. При этом необходимо учитывать как частоту среза самого объекта, так и частоту воздействия на него.

Из рассмотренных примеров можно сделать вывод о возможности применения дискретного моделирования системы без ухудшения точности, по сравнению с непрерывными методами. При этом дискретные методы обеспечивают значительное повышение скорости вычислений за счет уменьшения арифметических операций.

1. Вагапов В.Б. Автоматика радиоэлектронных систем. – К.: Вища школа, 1988. – 351 с.
2. Бабак В.П., Хандецкий В.С., Е. Шрюфер. Обробка сигналів: Підручник. – К.: Либідь, 1996. – 392 с.
3. Остапенко Ю.О. Ідентифікація та моделювання технологічних об'єктів керування. – К.: Задруга, 1999. – 424 с.
4. Гурецкий Х. Анализ и синтез систем управления с запаздыванием. Пер. с польского. – М.: Машиностроение, 1974. – 328 с.

ТЕХНОЛОГІЯ ТАБЛИЧНИХ ПРОЦЕСОРІВ – ЕФЕКТИВНИЙ ЗАСІБ РОЗВИТКУ КРИТИЧНОГО СТИЛЮ МИСЛЕННЯ СТУДЕНТІВ-АГРАРІЇВ

Т.І. Хачумян

м. Суми, Сумський державний аграрний університет

Успіх соціально-економічних і політичних перетворень в Україні залежить не тільки від об'єктивних умов, а й від ефективності роботи фахівців усіх рівнів. Мінливі умови ринкових відносин, конкуренції, все більша інтелектуалізація праці, життя в інформаційному суспільстві, інтеграція в світову економіку вимагають від спеціаліста не тільки високого рівня кваліфікації, професійної компетентності, а й високого інтелектуального розвитку, критичного, аналітичного мислення, уміння приймати самостійні рішення, тобто певних особистих якостей, нового тиру розумової діяльності. Отже, в сучасній системі освіти завдання інтелектуального розвитку особистості набувають особливої актуальності.

Поняття критичного мислення досить багатогранне, йому притаманний цілий комплекс інтелектуальних умінь, якостей, дій та операцій, найважливішими серед яких є:

- інформаційні: вміння самостійно працювати з різними джерелами інформації, збирати необхідну інформацію, сортувати її, відбирати головне, найістотніше, користуючись сучасними системами та засобами пошуку, зберігання, обробки інформації [6, 8];
- прогностичні: передбачення ймовірних результатів, вибір системи можливих дій для їх досягнення [4];
- дослідницькі: вміння спостерігати, аналізувати, порівнювати, класифікувати та систематизувати; моделювати процеси взаємодії об'єктів та прогнозувати можливі варіанти їх розвитку; висувати гіпотези та перевіряти їх, планувати і проводити експеримент, оцінювати його результати, формулювати висновки [3, 6, 7];
- комунікаційні навички: обговорення проблеми в колективі, групі; чітке викладання своєї думки, аргументований її доказ;

- здатність відстоювати свої ідеї та уважне, зацікавлене ставлення до чужих [8];
- здатність аналізувати отримані результати, мотиви та перебіг власних мислительних дій, що є характеристикою змістової рефлексії [2, 6, 10].

Зазначені якості мислення не виникають самі по собі. Виховувати, розвивати, вдосконалювати їх протягом навчального процесу потрібно цілеспрямовано. Питання співвідношення навчання та розвитку розроблялося багатьма видатними вітчизняними та закордонними психологами і педагогами. Формування інтелектуальних умінь учнів та студентів, критичного, дослідницького стилю мислення доцільно здійснювати на основі діяльнісного підходу до навчання (Л.С. Виготський, Г.С. Костюк, С.Л. Рубінштейн, О.М. Леонтьєв, М.Н. Скаткин, Г.І. Щукіна та ін.). Цей підхід спрямований не лише на засвоєння готових знань, а й на способи їх здобуття та використання, способи міркувань, що застосовуються при вивченні тієї чи іншої навчальної дисципліни.

С.Л. Рубінштейн вказував, що процес навчання відбувається через засвоєння знань, а розвиток мислення людини – через виконання завдань.

О.М. Леонтьєв підкреслював, що для того, щоб оволодіти знаннями і уміннями, необхідно здійснити діяльність, адекватну тій, яка втілена в цих знаннях і уміннях.

Ефективним засобом формування вказаних вище навичок є нові інформаційні технології, особливий інтерес серед яких представляють технології табличних процесорів.

Табличні процесори є зручними інструментами для відображення, обробки економічної інформації, корисними для тих фахівців, наприклад, аграріїв, діяльність яких потребує виконання складних розрахунків, а також є наявною велика кількість різноманітних чинників, що впливають на правильність рішення.

Вивчення табличних процесорів передбачене чинною програмою з інформатики, їх ефективність для інтенсифікації навчального процесу також є загально визнаною. А от ідея розглядати ці технології як засіб інтелектуального розвитку особистості, вдосконалення певних» розумових дій є порівняно новою [1].

Уміння наочно представляти інформацію у вигляді таблиць, графіків, діаграм - обов'язкова частина сучасної освіти. Уміння знаходити потрібну інформацію, перевіряти її достовірність, обробляти, зберігати, аналізувати є важливою практичною навичкою. Учні повинні володіти цими розумовими операціями в процесі власне роботи з даними, щоб відповідати на питання, як змінюються ці дані, як їх можна оцінювати. Табличні процесори – адекватні інструменти виконання таких дій [9].

Отже, метою роботи є вивчення можливостей електронних таблиць для розвитку дослідницьких навичок, формування критичного стилю мислення.

В процесі дослідження розроблено комплекс завдань для студентів 2-го курсу спеціальності “Аграрний менеджмент”, який з відповідним коректуванням можна використовувати і для навчання студентів інших спеціальностей. З метою інтеграції курсу інформатики з іншими дисциплінами завдання створено на матеріалі спеціальних предметів: “Технологія виробництва продукції рослинництва”, “Технологія виробництва продукції тваринництва”, “Механізація с/г виробництва”, “Аналіз господарської діяльності с/г підприємств”, “Організація с/г виробництва” тощо. Завдання орієнтовані на вирішення проблем практичної діяльності, демонструють можливості застосування електронних таблиць в професійній сфері, але дібрані таким чином, щоб уникнути спеціальних методик розрахунку (якими ще не володіють студенти 2-го курсу і яких не зобов'язаний детально знати викладач інформатики). Програмною реалізацією є середовище відомого пакета Microsoft Excel.

Незаперечною перевагою MS Excel є можливість застосування графічного способу рішення багатьох завдань, який дозволяє отримати результат без математичних розрахунків.

Цей метод рішення демонструє перевагу комп'ютерної графіки, причому використовуються її ілюстративна та когнітивна (що сприяє пізнанню) 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 складатиметься в тому,

що за допомогою графічного зображення можна отримати нове знання – про тенденцію показника, що досліджується, його наступні значення (наприклад, за допомогою лінії тренда). Загалом, застосовуючи графічні можливості MS Excel, студенти не тільки набувають знань про використання діаграм та графіків у табличних процесорах, побудову лінії тренда, а також отримують навички прогнозування та формулювання гіпотез відносно закономірностей, що досліджуються, уміння вирішувати задачу різними методами; вчать аналізувати одержані в обчислювальних та графічних експериментах результати, робити висновки.

Процес побудови і дослідження моделей активізує та розвиває дослідницький стиль мислення учнів. Засоби MS Excel “Подбор параметра”, “Поиск решения”, “Сценарии” – незамінні інструменти для цього, які позбавляють рутинних перерахунків, підбору варіантів “вручну” і дозволяють зосередитися на побудові гіпотез, їх аналізі, порівнянні, формулюванні висновків і прийнятті рішень за результатами досліджень.

Обґрунтування вибору оптимальних рішень із запропонованих програмою потребує вміння виявляти та враховувати вплив всіх умов та чинників. Звісно, для детального аналізу отриманих моделей і прийняття обґрунтованого рішення про доцільність їх вибору студенти 2-го курсу ще не володіють достатніми професійними знаннями, самі завдання також не надають всієї необхідної для цього інформації. Важливим є те, що під час виконання завдання студенти набувають навичок моделювання – одного з найважливіших навичок економіста, вчать висувати різні варіанти рішення, порівнювати, оцінювати, виявляти недоліки і переваги кожного, вибирати оптимальний. У процесі колективного обговорення моделей виробляються уміння аргументовано доводити свою точку зору, враховувати чужу думку, тобто формуються комунікаційні якості, навички ділового спілкування, культура мови.

Крім розвитку навичок пошуку інформації, вміння формулювати запити, визначати критерії та області пошуку, сортувати та аналізувати знайдену інформацію, виконання запропонованих завдань сприяє формуванню рефлексивного мислення, тобто усвідомлення необхідності перевірки, аналізу отриманого рішення, порівняння його з передбачуваним результатом. Ці

якості є одними з найважливіших показників дослідницького, критичного стилю мислення, навички якого, як показує практика, не вироблені у багатьох студентів.

Розроблений комплекс завдань спрямований на досягнення декількох педагогічних цілей:

- навчально-пізнавальну: накопичення фонду базових знань, умінь і навичок роботи в середовищі однієї з найбільш масових інформаційних технологій – табличних процесорів; знайомство з сучасними методами представлення, зберігання, пошуку, обробки економічної інформації;
- пропедевтичну: освоєння інструменту, знаряддя праці, що закладає основу широкого використання комп'ютерних технологій в подальшій навчальній та професійній діяльності. Реальний практичний зміст завдань забезпечує ілюстрацію практичної цінності і значущості набутих знань;
- розвивальну: підвищення інтелектуального рівня студентів, формування дослідницьких вмінь та навичок, розвиток критичного мислення.

Ефективність табличних процесорів як засобів розвитку розумових здібностей студентів зумовлена використанням інтелекту людини-користувача, а не програми. Комп'ютерам передається частина непродуктивної роботи, яку вони виконують краще за людину, наприклад, зберігання інформації, рутинні обчислення, перебір варіантів тощо. За студентом залишаються дії по організації, структуризації інформації, її аналіз, оцінювання, прийняття оптимальних рішень, осмислення отриманих результатів. При наявності декількох варіантів виникає проблема вибору, а це один з тих процесів, які погано піддаються формалізації, а отже, їх не можна доручити комп'ютеру. Засоби обробки електронних таблиць можна використовувати для формування як алгоритмічних прийомів розумової діяльності, які забезпечують рішення задач відомих типів, вчать студентів логіці міркувань, так і евристичних, що дозволяють діяти в умовах невизначеності, в принципово нових ситуаціях, полегшувати пошук розв'язання нових проблем, експериментувати, генерувати нові ідеї, гіпотези та перевіряти їх, пробуючи ті або інші варіанти, відкидати хибні, тобто дозволяють більш продуктивно використовувати та розвивати інтелектуальні здібності студентів, стимулювати та ак-

тивізувати продуктивну розумову діяльність, формувати навички дослідницького, критичного стилю мислення.

Література

1. Джонассен Д.Х. Компьютеры как инструменты познания: изучение с помощью технологии, а не из неё. // Информатика и образование. – 1996. – №4. – С. 117-125.
2. Дусавицький О.К. Система розвивального навчання: засади становлення // Початкова школа. – 1996. – №11. – С. 4-7.
3. Кларин М.В. Столкновение с проблемой: Обучение исследованию путем выдвижения и проверки гипотез. // Директор школы. – 1995. – №5. – С. 66-72
4. Коротяев Б.И. Учение – процесс творческий: Кн. для учителя: Из опыта работы. – 2-е изд., доп. и испр. – М.: Просвещение, 1989. – 159 с.
5. Орвис В. Дж. EXCEL для учёных, инженеров и студентов: Пер. с англ. – К.: Юниор, 1999. – 528 с.
6. Полат Е.С. Новые педагогические технологии и компьютерные коммуникации // Изд. центр «Академия», 2000. – 224 с.
7. Разумовский В.Г. Обучение и научное познание // Педагогика. – 1997. – №1. – С. 7-13.
8. Роберт И.В. Современные информационные технологии в образовании: дидактические проблемы, перспективы использования. – М.: Школа-Пресс, 1994. – 205 с.
9. Уваров А.Ю. Новые информационные технологии и реформа образования // Информатика и образование. – 1994. – №3. – С. 3-14.
10. Фридман А.М. Психолого-педагогические основы обучения математике в школе. – М.: Просвещение, 1983. – 158 с.

СРЕДА ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ DELPHI В УЧЕБНОМ ПРОЦЕССЕ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕ- НИЙ: ПРЕИМУЩЕСТВА И НЕДОСТАТКИ

С.В. Шаров

г. Мелитополь, Мелитопольский государственный педагогический институт

В последнее время на Украине наметилась тенденция к повышению интереса к компьютеризации процесса обучения и жизни общества в целом. Поэтому в учебных заведениях делается упор на повышение компьютерной грамотности студентов в процессе обучения.

В курсе информатики для высших учебных заведений можно использовать изучение Delphi. Delphi – это интегрированная среда визуальной разработки приложений для операционной системы Windows (ОС Windows). Эта среда позволит студентам достаточно быстро увидеть результаты своей работы на экране монитора, а также разобраться со многими свойствами и событиями объектов ОС Windows.

К преимуществам Delphi можно отнести:

- Данный программный продукт предназначен для создания приложений под ОС Windows, которая является самой распространенной и широко используемой в мире среди пользователей персональных компьютеров.
- Языком программирования в Delphi является Object Pascal, который является “родственником” Turbo Pascal и изучается во многих школах. Эти языки имеют аналогичную конструкцию написания программного кода.
- Другие визуальные среды программирования тоже можно использовать в учебном процессе, но они имеют ряд недостатков. Visual Basic он уступает Delphi по внешнему виду и по удобству использования возможностей самой среды разработки приложений. А в среде программирования Visual C++ необходимо работать на более низком уровне.
- Эта среда позволяет непосредственно наблюдать работу студента на стадии разработки приложений.
- На ознакомление с этой средой программирования пона-

добится не очень много времени, так как обычному студенту необходимо лишь ознакомиться с этой средой, а не изучать ее досконально.

- Delphi обладает достаточно простым и понятным интерфейсом, большим набором стандартных компонентов для создания приложений, а также возможностью устанавливать дополнительные компоненты, создавать новые.

- Существует большое количество учебников, как в печатном, так и в электронном виде, где можно почерпнуть необходимую информацию. Некоторые из них методически разработаны.

Конечно, как и любого другого программного продукта, у Delphi есть свои ограничения для применения, среди которых можно назвать:

- Почти каждый год выходит новая версия, которая более соответствует современным требованиям мира информационных технологий и запросам программистов. И чем новее версия, тем больше она требует ресурсов компьютера. Но не обязательно в учебном процессе использовать самые новые версии данного программного продукта. Последние версии почти ничем не отличаются друг от друга в отношении интерфейса, использования объектно-ориентированного программирования (ООП), основных возможностей этого программного продукта.

Из вышеперечисленного можно сделать вывод, что эта визуальная среда программирования может быть использована для изучения в высших учебных заведениях. К тому же Delphi является одним из самых распространенных визуальных сред среди инструментальных программ.

Студенты смогут успешно освоить основные принципы программирования на языке программирования Object Pascal, смогут понять основные принципы ООП, а также создавать более-менее сложные приложения для ОС Windows.

ФОРМУВАННЯ КУЛЬТУРИ ПРОФЕСІЙНОГО МОВЛЕННЯ МАЙБУТНІХ ВЧИТЕЛІВ ІНФОРМАТИКИ В ПРОЦЕСІ ДИФЕРЕНЦІЙОВАНОГО ВИВЧЕННЯ СПЕ- ЦІАЛЬНИХ ДИСЦИПЛІН

Г.В. Шугайло

м. Мелітополь, Мелітопольський державний педагогічний уні-
верситет

У сучасних умовах, коли шляхом конституційного та законодавчого закріплення унормовується застосування української мови як державної, кожний громадянин України має подбати про те, щоб підвищити рівень своєї мовної культури. Відповідальне завдання підвищення рівня мовно-національної вихованості народу покладається державою насамперед на педагогічних працівників – носіїв нормативної української літературної мови. З огляду на це, одним із украї необхідних завдань вищої педагогічної освіти на сучасному етапі є формування мовної культури майбутніх вчителів.

Важливою складовою мовної культури вчителя є культура професійного мовлення. Передбачається, що вчитель із високим рівнем культури професійного мовлення достоту володіє категоріальним апаратом науки, що викладається ним у рамках навчальної дисципліни; вміє перенести мовленнєві знання, уміння і навички, набуті ним в процесі опанування загальних норм української літературної мови, до сфери суто професійної викладацької діяльності; може дібрати природний, відповідний духові української мови замітник для запозиченого терміну тощо.

Аналіз літературних джерел свідчить, що процес формування культури професійного мовлення майбутнього вчителя є об'єктом дослідження науковців. Наприклад, запропонована модель формування культури професійного мовлення майбутнього вчителя, структурою якої передбачено три етапи: пропедевтичний, основний і завершальний. Кожний з них, в свою чергу, реалізується у складі відповідного навчального курсу [2]. Зазначається, що система роботи з формування культури професійного мовлення майбутніх вчителів має охоплювати усі п'ять років підготовки фахівців, розпочинаючи з першого року навчання

студента у педвузі (пропедевтичний етап) включаючи навчальні роки, що передують безпосередньому педагогічному стажуванню студентів у середніх навчальних закладах (основний етап), і завершуючи роботою за фахом під час педпрактики протягом 7–10 семестрів (завершальний етап).

Робота над формуванням культури професійного мовлення майбутнього вчителя особливо актуальна у світлі перемін, які відбуваються на даному етапі: зміни у правописі, апробація нових лексем, упорядкування української наукової термінології, перегляд виробничо-професійної лексики тощо. Надзвичайно важливою справою є формування культури професійного мовлення майбутнього вчителя інформатики. Як зазначалося в резолюціях 1-ї та 2-ї міжнародних конференцій “Проблеми української науково-технічної термінології”, що відбулися 22-25 вересня 1992 р. і 21-2 вересня 1993 р. у Львові, “у вік постіндустріальний, у вік інформатики долю будь-якої нації остаточно вирішує науково-технічне слово: житимуть нації, мовою яких говоритимуть комп’ютери” [3].

У справі формування такого типу мовлення, який повною мірою б сприяв виконанню учителем інформатики своїх професійних обов’язків, нашу увагу привернули дидактичні можливості теми “Комп’ютерна технологія створення і обробки текстової інформації. Текстові редактори і процесори”. Згідно загальноприйнятої методики, можливості текстового редактору (процесору) будь-якого призначення і складності опановують шляхом оброблювання текстової інформації засобами цього прикладного програмного засобу. Ми у рамках лабораторної роботи “Ілюстрування документу Microsoft Word” пропонуємо систему диференційованих лабораторних завдань, розроблених на двох рівнях, кожний з яких – у чотирьох варіантах. Виконуючи ці завдання, студенти, з одного боку, опановують техніку роботи із різноманітними об’єктами текстового процесору Word (таблицями, надписами, малюнками, формулами тощо), а, з іншого боку, вчать переносити знання, уміння та навички, набуті ними під час загальної мовленнєвої підготовки, до сфери їх професійного мовлення. Нижче наводяться приклади завдань (по одному варіанту) відповідно для першого і другого рівнів, узяті із [2]. За-

вдання для обох рівнів сформульовані однаково: “Оформіть вихідний текст відповідно із взірцем згідно до обраного варіанту”.



Рівень 1
Варіант 3

“Бокс для дискет”, “папка для текстових документів”, “кошик для сміття” — ненормативні вислови. Прийменник для слухний там, де говориться, що певну річ призначено для людини, наприклад: “Цей комп’ютер батьки купили для мого брата-студента”. У разі, коли мовиться, що якийсь предмет призначено на конкретні речі, тоді вживають прийменник на. Отже, треба: “бокс на дискети”, “папка на текстові документи”, “кошик на сміття”.

ДЛЯ
чи
НА

В російській мові прийменник із-за вживають у двох випадках:

ІЗ-ЗА
чи
ЧЕРЕЗ

при позначенні предметів, з протилежного боку яких відбувається дія, і при позначенні причинності. В українській мові прийменник із-за вживають тільки у першому значенні, для позначення причинності послуговуються іншими прийменниками, здебільшого прийменником через: “через відсутність” (а не “із-зі відсутності”).



Правила чергування голосних у прийменниках у та в. Рис. 1

Ненормативні і нормативні вислови

Таблиця 1.

НЕ ПРАВИЛЬНО	ПРАВИЛЬНО
Виконав це по дорученню вчителя	Виконав це за дорученням вчителя
З кожним роком	Щороку
На протязі тижня	Упродовж (протягом) тижня
При таких умовах	За таких обставин
У найближчі дні	Найближчими днями
Із-за низького розрешування екрана	Через низьку роздільну здатність екрана
Доступ к пам'яті	Доступ до пам'яті
Захист від комп'ютерних вірусів	Захист проти комп'ютерних вірусів
Звертання к процедурі	Звертання до процедури
Сортировка по спаданню	Сортування за спаданням
При обчисленні функції	Обчислюючи функцію
$y = \sqrt[3]{(x-1)^2}$	$y = \sqrt[3]{(x-1)^2}$

Рівень 2. Варіант 2

Що треба пам'ятати про іменники, аби не порушувати мовну норму?

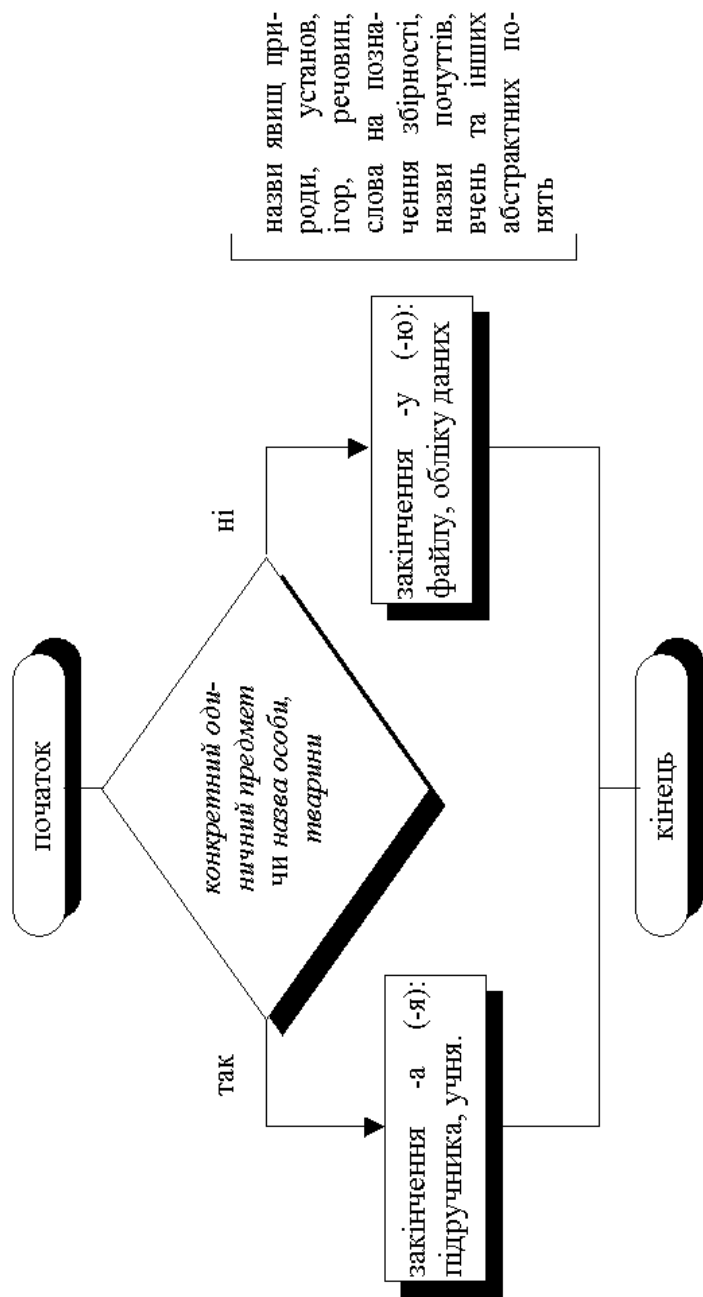


увазі одну дискету, а формою родового відмінка наголошуємо, що беремо її на деякий час)).

Ненормативними є вислови: **Збірні іменники** канцелярські приладдя, комп'ютерні обладнання, шкідливі мерехтіння екрану. Збірні іменники мають форму тільки однини, тож правильно: канцелярське приладдя, комп'ютерне обладнання, шкідливе мерехтіння екрану.

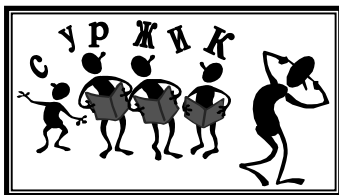
Прямий додаток коло передного дієслова ставлять у родовому відмінку, якщо присудок виражає заперечення (учень не звернув уваги на це правило), якщо підкреслюють, що дія переходить лише на частину предмету (інженер перевіряв диск на наявність вірусів (тобто не весь диск)) і якщо мова йде про обмеження часу (позич мені дискети (маємо на

Відмінок додатка



Закінчення у родовому відмінку однини іменників чоловічого роду другої відміни Рис. 1

Відповідно до широко вживаних у російській діловій мові слів отсутствие, отсутствующий, присутствие, присутствующий в українській мові також є схожі слова відсутність, відсутній, присутність, присутній: “Треба записати всіх відсутніх на уроці” (з живих уст). Проте слова відсутність, відсутній уживають навіть тоді, коли йдеться не про людей, а про речі. Нерідко можна почути: відсутність досвіду, відсутність приладу. Правильно казати по-українському: брак досвіду, немає приладу.



Ненормативні і нормативні мовленнєві моделі

Таблица 1.

НЕПРАВИЛЬНО	ПРАВИЛЬНО
Вислухали всіх без виключення	Вислухали всіх без винятку
Для наглядності	Для унаочнення (увиразнення)
Маю підготувати тези докладу	Маю підготувати тези доповіді
Дбайливе відношення до обліку	Дбайливе ставлення до обліку
Запрос аргумента	Запит аргументу
Стискування даних	Ущільнення даних
Бумага для печаті	Папір для друку
Поля сторінки	Береги сторінки
Локальна комп'ютерна сітка	Локальна комп'ютерна мережа
Привести в порядок файли	Упорядкувати файли
Руководство програміста	Посібник програміста
Модель “цвіт — яркість — насиченість”	Модель “колір — яскравість — насиченість”
Сброс лічильника	Скидання (скинення) лічильника
Цифрова связь	Цифровий зв'язок
Вислів $\frac{a^{\frac{4}{3}} \cdot x + a \cdot x^{\frac{4}{3}}}{\sqrt[3]{a} + \sqrt[3]{x}}$	Вираз $\frac{a^{\frac{4}{3}} \cdot x + a \cdot x^{\frac{4}{3}}}{\sqrt[3]{a} + \sqrt[3]{x}}$

Добираючи варіанти текстів для обох рівнів лабораторної роботи, ми, поряд із метою забезпечити диференційований під-

хід до студентів, також переслідували мету: “Сприяти підвищенню культури професійного мовлення майбутнього вчителя інформатики”. При цьому керувалися таким.

1. Вивчення можливостей текстового процесору Word може бути поєднаним із демонстрацією того, як отримані у традиційних курсах мовлення знання використовуються у педагогічно-професійній мовленнєвій практиці вчителя інформатики. Таким чином, мовленнєва теорія конкретизується. Студент оперує із знайомими йому мовленнєвими правилами і положеннями вже “не взагалі”, а на конкретному педагогічно-професійному мовленнєвому матеріалі.
2. Рекомендувати студентам до вживання слова, вислови та мовленнєві комунікативні формули, що закріпилися у літературній мові і визнані за нормативні. Водночас застерегти студентів від вживання мовних покручів, калькованих або транслітерованих термінів, неприродних фраз, штучних комунікативних формул, штапованих висловів із відтінком канцелярщини, суржика тощо.
3. Ознайомити студентів із змінами в українській науковій термінології, зокрема із поверненими суто українськими, свого часу незаслужено вилученими із мовного вжитку термінами. Показати принципи, якими слід керуватися у доборі лексем з інформатики. Попередити тавтологію, для чого ознайомити студентів із українськими заміниками транслітерованих англійських термінів.

Література

1. Кретьова О.І. Педагогічне обґрунтування навчальних курсів з культури професійного мовлення майбутніх вчителів // Вісник Черкаського університету. – 2000. – Випуск 17. – С. 49-52.
2. Методичні розробки для диференційованого навчання студентів. Інформатика: комп’ютерна технологія створення і обробки текстової інформації / Укладач Г.В. Шугайло. – Мелітополь, 2000. – 124 с.
3. Російсько-український словник з інформатики та обчислювальної техніки / В.Я. Карачун, Г.Г. Гульчук, О.О. Карачун, Ю.З. Прохур / За ред. І.А. Черненко. – К.: видавництво “Рось”, 1994. – 368 с.

Зміст

<i>С.В. Артюшенко, Н.В. Моисеєнко.</i> К вопросу о прикладных задачах в курсе событийно-ориентированного программирования в высших учебных заведениях.....	3
<i>О.П. Бесклинська, С.І. Міхневич.</i> До питання інтенсифікації процесу навчання комп'ютерним інформаційним технологіям	8
<i>М.П. Білан.</i> З досвіду викладання курсу «Основи алгоритмізації та програмування».....	15
<i>В.Г. Борищенко.</i> Використання сценаріїв електронної таблиці Excel для аналізу даних студентами вищого навчального аграрного закладу та контроль знань	19
<i>Л.С. Возняк, Б.М. Дрінь, В.В. Дудка, О.М. Дудка.</i> Тестові завдання як інструмент для порівняння знань учнів з інформатики	22
<i>М.С. Головань.</i> Проблеми навчання інформатики в школі	30
<i>Д.В. Гондюл.</i> Вимоги до користувачів програмного забезпечення персональних комп'ютерів в умовах розвитку інформаційних технологій інформаційного суспільства	35
<i>Г.А. Горщикова.</i> Використання життєвого досвіду учнів як елемент особистісно-зорієнтованого навчання на уроках інформатики.....	44
<i>В.Н. Евтеев, О.М. Ткаченко, Г.Е. Черный.</i> Конструирование вербальных тренажеров.....	47
<i>М.С. Жуков, Ю.В. Темнюк, Р.О. Постоєнко, С.В. Діордіца.</i> Динамічна комп'ютерна модель цифрових схем обчислювальної техніки.....	53
<i>Н.С. Жуков, М.В. Швець, С.А. Томилин.</i> Использование компьютерного моделирования при изучении абстрактных типов данных в вузах.....	56
<i>О.А. Жульковский, А.В. Лисовец, И.А. Жульковский.</i> Система компьютерного тестирования знаний	59
<i>М.І. Задорожній.</i> Навчання з точки зору інформаційних технологій.....	60
<i>М.М. Задорожній.</i> Вивчення чисельних методів розв'язування алгебраїчних рівнянь у школі	65
<i>Л.А. Карташова.</i> Методика використання інструментальних засобів для ефективної підготовки документів	68
<i>Н.П. Козаченко.</i> Розвиток пізнавальної активності школярів засобами комп'ютерного навчання.....	73

<i>С.Р. Красильников, І.С. Пятін, О.Ю. Рудик.</i> Актуальність вивчення мови Java у курсі “Інформатика і комп’ютерна техніка” при підготовці фахівців спеціальності “Економічна кібернетика”	81
<i>Ю.М. Красюк.</i> Можливості реалізації модульно-рейтингової технології навчання інформатики у вищих закладах освіти економічного профілю	89
<i>Б.Б. Круліковський.</i> Методичне забезпечення лабораторного практикуму з основ інформатики	95
<i>Г.И. Кулик.</i> К вопросу совершенствования методики преподавания дисциплины «Информатика» для студентов строительных специальностей.....	96
<i>С.И. Куликов, С.П. Кандзюба.</i> Методика изучения некоторых разделов информатики, основанная на использовании обучающих программ.....	99
<i>О.В. Курочкина.</i> Психолого-педагогічні аспекти викладання інформатики у 5–9 класах.....	104
<i>Є.А. Лавров, А.Т. Ашеро́в, Г.А. Смоляров.</i> Концепція комп’ютеризації аграрного університету	107
<i>В.В. Лапінський, В.М. Дем’яненко.</i> Покращення практичного спрямування фахової підготовки учителя інформатики	111
<i>В.Л. Малорян.</i> К вопросу о выборе первого языка программирования.....	116
<i>О.М. Марченко, М.С. Панченко.</i> Методичний проект «Система лекційно-практичних занять по вивченню електронних таблиц Microsoft Excel» в загальноосвітньому навчальному закладі ..	119
<i>А.А. Мясищев.</i> Использование CGI+Perl для создания простейшей базы данных.....	127
<i>С.М. Прийма.</i> Про деякі аспекти використання систем візуального програмування в освіті.....	137
<i>І.М. Пустинникова, В.В. Локтюшин.</i> Діагностика знань і вмінь студентів за допомогою експертних систем.....	142
<i>Ю.П. Рева.</i> Проблеми інформатизації сучасних технологій навчання	149
<i>О.Ю. Рудик, С.Р. Красильников.</i> Методика оцінки знань з дисципліни “Інформатика і комп’ютерна техніка”	153
<i>М.В. Сальніков, В.Г. Кудрявцев.</i> Організація обліку використання ресурсів Internet у навчальному процесі	156

<i>В.А. Сергієнко.</i> Наступність вивчення курсу “Комп’ютерна техніка”	160
<i>Г.А. Смоляров.</i> Досвід викладання дисциплін комп’ютерного циклу при підготовці спеціалістів АПК	162
<i>А.М. Стрюк.</i> Особливості викладання інформатики для некомп’ютерних спеціальностей у вищих навчальних закладах	164
<i>І.О. Теплицький, О.І. Теплицький.</i> Факультативний курс «Комп’ютерне моделювання з фізики»	169
<i>С.А. Томилин.</i> Элементы моделирования на уроках информатики в специализированных классах	180
<i>Ю.В. Филатов.</i> Методика подготовки участников олимпиад по информатике	184
<i>А.А. Хараджян.</i> Сравнение дискретного и непрерывного подхода к численному решению дифференциальных уравнений	187
<i>Т.І. Хачумян.</i> Технологія табличних процесорів – ефективний засіб розвитку критичного стилю мислення студентів-аграріїв	190
<i>С.В. Шаров.</i> Среда визуального программирования Delphi в учебном процессе высших учебных заведений: преимущества и недостатки	196
<i>Г.В. Шугайло.</i> Формування культури професійного мовлення майбутніх вчителів інформатики в процесі диференційованого вивчення спеціальних дисциплін	198

Наукове видання

**Теорія та методика навчання
математики, фізики, інформатики**

В 3-х томах

Том 3

Підп. до друку 11.04.2001
Бумага офсетна №1
Ум. друк. арк. 11,02

Формат 80x84 1/16.
Зам. №4-1103
Наклад 500 прим.

Видавничий відділ Криворізького державного педагогічного університету
КДПУ, 50086, Кривий Ріг-86, пр. Гагаріна, 54

E-mail: cc@kpi.dp.ua