

Міністерство освіти і науки, молоді та спорту України
Криворізький національний університет

Теорія та методика
навчання математики,
фізики, інформатики

Збірник наукових праць
Випуск X

Том 3

Кривий Ріг
Видавничий відділ НМетАУ
2012

Теорія та методика навчання математики, фізики, інформатики : збірник наукових праць. Випуск X : в 3-х томах. – Кривий Ріг : Видавничий відділ НМетАУ, 2012. – Т. 3 : Теорія та методика навчання інформатики. – 213 с.

Збірник містить статті з різних аспектів дидактики інформатики і проблем їх викладання у ВНЗ та школі. Значну увагу приділено питанням розвитку методичних систем навчання інформатичних дисциплін та фундаменталізації інформатичної освіти в контексті орієнтирів Болонського процесу.

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

Редакційна колегія:

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

М.І. Жалдак, доктор педагогічних наук, професор, ак. НАПН України

Ю.С. Рамський, кандидат фізико-математичних наук, професор

В.І. Клочко, доктор педагогічних наук, професор

С.А. Раков, доктор педагогічних наук, професор

Ю.В. Триус, доктор педагогічних наук, професор

П.С. Атаманчук, доктор педагогічних наук, професор

В.Ю. Биков, доктор технічних наук, професор, ак. НАПН України

О.Д. Учитель, доктор технічних наук, професор

І.О. Теплицький, кандидат педагогічних наук, доцент (відповідальний редактор)

С.О. Семеріков, доктор педагогічних наук, професор (відповідальний редактор)

Рецензенти:

Г. Ю. Маклаков – д-р техн. наук, професор, професор кафедри інформаційних технологій Кіровоградської льотної академії Національного авіаційного університету

А. Ю. Ків – д-р фіз.-мат. наук, професор, завідувач кафедри фізичного та математичного моделювання Південноукраїнського національного педагогічного університету імені К. Д. Ушинського (м. Одеса)

Друкується згідно з рішенням ученої ради Криворізького металургійного факультету Національної металургійної академії України, протокол №8 від 14 березня 2012 р.

О ФУНДАМЕНТАЛИЗАЦИИ ПОДГОТОВКИ БУДУЩИХ УЧИТЕЛЕЙ ИНФОРМАТИКИ

Е. Ы. Бидайбеков, Г. Б. Камалова

Казахстан, г. Алматы, Казахский национальный педагогический
университет им.Абая
esen_bidaibekov@mail.ru

Глобальные экономические изменения и кардинальные социальные преобразования, создание общеевропейской системы образования и задачи развития определили необходимость модернизации современного казахстанского образования с целью повышения эффективности и качества подготовки специалистов до уровня, достигнутого в развитых странах. Одним из направлений модернизации является фундаментализация образования, системной характеристикой которой служит направленность обучения на постижение и использование глубинных, сущностных, системообразующих оснований и связей между разнообразными процессами окружающего мира.

Фундаментализация образования закономерна, поскольку в быстро изменяющихся социально-экономических и информационно-технологических условиях необходимы методологические и системообразующие знания, инвариантные элементы человеческой культуры, сохраняющие свою значимость в течение всей жизнедеятельности человека, делающие его мобильным и востребованным на рынке труда.

В рамках гуманистической парадигмы, основанной на теоретических положениях гуманной педагогики и гуманитарной методологии, когда человеческая личность становится приоритетной в образовании, ее формирование и развитие является главной ценностью и важнейшим результатом образования, фундаментализация образования приобретает новое значение [1]. Сегодня – это активная деятельность субъектов образовательного процесса, направленная как на фундаментализацию содержания образования, так и на гуманизацию образовательного процесса. Поэтому современными ориентирами системы образования являются создание разнообразных возможностей для приобретения человеком подлинно фундаментальных личностно значимых знаний и формирования готовности применить эти знания в своей деятельности, выработка стиля мышления. И, прежде всего, системного, так как в современных условиях «мыслить системно становится общественно осознанной потребностью» [2]. Идея системности окружающего мира может стать концептуальной основой для интеграции научного знания в содержании образования, способствовать преодолению фрагментарности содержа-

ния обучения, становлению целостности мировосприятия в сознании обучающихся.

Очевидно, что процессы фундаментализации образования в контексте существующих ее особенностей не могут обойти стороной систему подготовки педагогов. В том числе и особенно систему подготовки будущих учителей информатики в предметной области, для которой характерен достаточно высокий темп обновления знаний.

Система подготовки будущих учителей информатики должна быть фундаментальной, чтобы обеспечить им возможность гибко перестраивать направление и содержание деятельности в связи со сменой технологий или требованиями рынка и формировать не только знания, но и потребности, умения и навыки к самообразованию в течение всей жизни. Кроме того, только при наличии соответствующих профессиональных качеств будущий учитель информатики будет в состоянии обеспечить обучение школьников фундаментальным основам информатики, сформировать у них представление об информатике как о фундаментальной науке, приобщить к системному использованию знаний и умений в области информатики при изучении других дисциплин, а также в последующей трудовой деятельности.

Следует сказать, что фундаментализацию образования целесообразно осуществлять на различных уровнях её функционирования, начиная с фундаментализации всей системы образования как ядра и конечной цели новой образовательной парадигмы. Представляется, что данный уровень фундаментализации может быть достигнут, если произойдет коренная перестройка системы высшего образования. Начало этому процессу положено – введена трехуровневая подготовка специалистов: бакалавр – магистр – доктор Ph.D; в учебный процесс введены кредитная технология обучения и современные инфокоммуникационные технологии; ведется подготовка докторов Ph.D в партнерстве с ведущими зарубежными вузами; вводится новый механизм обеспечения качества образования через создание системы институциональной и специализированной аккредитации по международной модели. В настоящее время проводится целенаправленная работа по вступлению в Болонский процесс.

Второй уровень – это фундаментализация дисциплин предметной подготовки, что подразумевает не простую совокупность традиционных курсов, а систему, объединенную общей целевой функцией, объектом исследования, методологией построения. При этом предполагается, что наряду с детальным изучением каждой дисциплины предметной подготовки необходимо приступить к наведению мостов между ними, выявлять общие стержневые методологические подходы и их отражение в каждой дисциплине.

Данный уровень фундаментальности неоднократно рассматривался в работах, посвященных предметной области «Информатика». Так, М. П. Лапчик, Е. Н. Бобонова, Т. Н. Райхерт, Е. К. Хеннер, В. В. Лаптев, Н. И. Рыжова, М. В. Швецкий [3] считают основой фундаментализации предметной подготовки будущих учителей информатики математический аппарат, отражающий основные свойства информации. К. К. Колин считает, что фундаментализация предметной области «Информатика» предполагает изучение законов природы и общества, природы и самого человека, чтобы позволить людям самостоятельно находить и принимать ответственные решения в условиях неопределённости, в критических и стрессовых ситуациях, в случаях, когда они сталкиваются с новыми, весьма сложными природными и социальными явлениями. Научные знания, в этих случаях, и высокие нравственные принципы становятся единственной надёжной опорой [4].

В Казахстане же акцент делается на объединении информационных дисциплин предметной подготовки будущих учителей информатики, связанных посредством математических и информационных моделей, которые образуют научное направление информатики, называемое вычислительной информатикой. Оно включает в себя «отображение алгоритмов на архитектуру вычислительных систем, прикладное программное обеспечение вычислительных задач и методологию численного моделирования процессов и явлений» [5]. Согласно определению предметом ее являются вычислительные алгоритмы (информационные модели), представляющие формализованное описание математических структур, и отображение алгоритмов на архитектуру вычислительных систем. Он расположен «внутри» предмета исследования информатики, которым являются общие принципы построения информационных моделей – формализованного описания информационных процессов, а также методы и средства их автоматической реализации (информационных технологий). Поэтому часть предметной области «информатика», включающая математические основания, формируемые теорией представления числовой информации в памяти ЭВМ, алгоритмы, как формы записи методов вычислений, теорию и практические приемы реализации вычислительных алгоритмов, математическое моделирование и формализацию, относится к направлению вычислительная информатика. Включение его в содержание подготовки будущих учителей информатики должно способствовать усилению ее фундаментализации, поскольку одним из возможных способов обеспечения фундаментализации подготовки по информатике может быть «перенос акцентов на применение формальных методов и соответствующего математического аппарата».

Ясно, что вопросы теоретических оснований алгоритмов численных

вычислений, включая проблематику численного моделирования, «составляя одну из основ информатики, теснейшим образом примыкают и к вычислительной математике, и к программированию, а в плане построения математических и информационных моделей – ко всем наукам, где возможно применение ЭВМ. В связи с этим налицо прямая связь и некоторый непрерывный переход вычислительной информатики (четкой границы установить невозможно) с вычислительной математикой, с одной стороны, и с многочисленными предметными науками (через математические и информационные модели) – с другой» [5]. Термины «вычислительная математика» и «вычислительная информатика» в связи с этим используются даже как синонимы. Хотя если быть более точным, можно сказать, что вычислительная информатика занимается вопросами исследования и реализации вычислительных алгоритмов в информатике и объединяет в себе комплекс дисциплин, связанных посредством математических и информационных моделей, существенной частью, ядром которого является вычислительная математика [6]. Часть ее вопросов, таких как математические основания представления чисел в памяти ЭВМ, вопросы, связанные с информационным, в частности математическим, моделированием, процесс решения вычислительной задачи, а главное методология математического моделирования и вычислительного эксперимента находят свое отражение в составных частях «ядра» современной информатики, а именно в теоретических вопросах разделов алгоритмизации, программирования, информационного моделирования.

Курс «Архитектура компьютера» дает представление об архитектуре вычислительных систем, способах обработки информации. Имеет большое значение для раскрытия сущности отображения абстрактных вычислительных алгоритмов на архитектуру вычислительных (компьютерных) систем.

Систематизировать и закрепить знания, полученные на этих курсах, позволяет дисциплина «Численные методы». Предметом ее изучения являются вычислительные алгоритмы и определение критериев для оценки их качества. Она играет основную роль в процессе формирования современного научного представления о теоретических основаниях алгоритмов численных вычислений, а также вопросах их реализации на ЭВМ, включая проблематику численного моделирования. Более того, как ядро вычислительной информатики, эта дисциплина, с одной стороны, наряду с изучением математической теории методов вычислений призвана продемонстрировать широкие применения математического аппарата для изучения различных процессов и явлений, с другой – показать богатство возможностей компьютерных информационных подходов к действительности и их принципиальную ограниченность. Она может

стать важнейшей связующей частью между различными видами подготовки учителя информатики и выполнять следующие функции: междисциплинарную, интегративную по отношению к математической, естественнонаучной и специальной подготовке в области информатики; способствовать осознанию методологии моделирования в целом как одной из ведущих в познании окружающего мира; развития и углубления навыков в области информационного моделирования, алгоритмизации, программирования и использования ЭВМ для решения различных задач.

Рассматриваемое направление информатики не исчерпывается перечисленными дисциплинами. Безусловно, основные ее положения находят развитие дальше в других курсах учебного плана подготовки учителей информатики («Методы оптимизации», «Исследование операций» и др.). В Казахском национальном педагогическом университете им. Абая, например, дальнейшее развитие вычислительная информатика получает также при изучении дисциплин по выбору, таких как «Параллельные вычисления», «Машинная арифметика и вопросы устойчивости вычислительных алгоритмов», «Теория разностных схем», «Обратные задачи для дифференциальных уравнений», которые занимают важное место в системе учебных мероприятий по дополнению и углублению профессиональных знаний будущего учителя информатики.

Усиление фундаментализации подготовки в предметной области через вычислительную информатику объясняется не только достаточно огромным ее потенциалом, оно еще оправдано и исторически.

При становлении информатики в Казахстане, также как и в других странах СНГ, а она создавалась преимущественно математиками, традиционно отдающими предпочтение фундаментальным знаниям, и в ранние годы ее развития, в ней, в основном, превалировало данное направление. Значительную долю ее исследований составляли тогда наряду с вопросами разработки ЭВМ и их программного обеспечения, проблемы решения вычислительных задач на компьютере. Вообще, в ранние годы информатика во многом сводилась к вычислительной науке (computer science) и преимущественно была вычислительной информатикой. И внедрение ее в систему образования еще в рамках первых факультативных курсов начиналось именно с элементов вычислительной информатики (элементов алгоритмизации, вычислительной математики, программирования решения вычислительных задач и т.д.). По мере того, как компьютеры становились способными решать все более сложные задачи, данное направление информатики приобретало все большее значение и важность. И сегодня, как много лет назад, вычислительная информатика занимает особое место в решении задач, возникающих в различных прикладных областях, поскольку и сегодня «самые совершен-

ные и дорогие компьютеры используются для решения задач с отчетливой математической подоплекой», называемых задачами математического моделирования. Об этом свидетельствуют исследования ученых, в том числе казахстанских: А. Т. Лукьянова, У. М. Султангазина, Ш. С. Смагулова и др. в космической, нефтегазовой и других отраслях именно в ракурсе моделирования реальных явлений и процессов, разработки и исследования вычислительных методов, включая отображение разработанных алгоритмов на архитектуру ЭВМ.

Использование компьютера при решении подобных задач и умение правильно истолковывать полученные результаты предъявляет особые требования к пониманию сути, как самой решаемой вычислительной задачи, так и выбранного метода решения. И предполагает включение в содержание подготовки специалистов в области информатики, в том числе и будущих учителей информатики, теоретических оснований алгоритмов численных вычислений, включая точность представления чисел в памяти ЭВМ, наряду с методологией математического моделирования и вычислительного эксперимента. Не вызывает сомнений, что это обеспечит специалистов указанной категории необходимым инструментарием, рассчитанным на длительную перспективу и достаточно инвариантным по отношению к возможным изменениям в области информационных технологий и вычислительной техники, усиливая фундаментализацию их подготовки. Одним из немаловажных аспектов подготовки по вычислительной информатике является, имеющаяся при соответствующей методике обучения, реальная возможность научить будущего учителя информатики системному подходу к осмыслению всего, что происходит вокруг него. Такая возможность обеспечивается использованием новой технологии исследования вычислительных задач, основанной на построении и анализе с помощью ЭВМ математических моделей изучаемого объекта, так называемой методологии математического моделирования и вычислительного эксперимента, в основе которой лежит системный подход. Соединяя в себе достоинства теоретического и экспериментального методов исследования объектов и явлений, она, охватив всю проблему в целом, позволяет произвести системный анализ объекта (предмета) и является своего рода программой деятельности, которая проектирует те познавательные действия и операции, которые последовательно открывают в предмете разные аспекты системно-структурной организации и выстраивают структуру знаний о нем как о целостности. Использование ее приводит к привычке мыслить системно. Одним словом, в вычислительной информатике заложена уникальная возможность: обеспечить фундаментальность подготовки с одной стороны, и с другой стороны, формировать системное мышление и видение

мира. Это очень важно в современных условиях фундаментализации образования и свидетельствует о необходимости подготовки будущих учителей информатики по рассматриваемому направлению.

Третий уровень рассмотрения фундаментализации образования – это уровень отдельной дисциплины. В качестве одной из них в предметной подготовке будущих учителей информатики выступает дисциплина «Теоретические основы информатики», играющая роль системообразующей науки внутри предметного цикла [7]. Содержание ее составляют такие фундаментальные понятия информатики, как информационные процессы, информационные модели и информационные основы управления, задающие основные направления развития предметной подготовки будущих учителей информатики

Представленные подходы к предметной подготовке будущих учителей информатики, на наш взгляд, позволяют наиболее полно реализовать идеи фундаментальности.

Литература

1. Левченко И. В. Развитие системы методической подготовки учителей информатики в условиях фундаментализации образования вузах : дис. на соискание ученой степени доктора пед. наук : 13.00.02 – теория и методика обучения и воспитания (информатика) / Левченко Ирина Витальевна. – М., 2009. – 46 с.
2. Новик И. Б. Системный стиль мышления: Особенности познания и управления в сложных системах / И. Б. Новик. – М. : Знание, 1986. – 64 с.
3. Лаптев В. В. Методическая теория обучения информатике. Аспекты фундаментальной подготовки / Лаптев В. В., Рыжова Н. И., Швецкий М. В. – СПб. : Изд-во С.-Петербур. ун-та, 2003. – 352 с.
4. Колин К. К. О структуре и содержании образовательной области «Информатика» / Колин К. К. // Информатика и образование. – 2000. – №10. – С. 5–10.
5. Ильин В. П. Вычислительная информатика: открытие науки / В. П. Ильин. – Новосибирск : Наука, Сиб. отделение, 1991. – 194 с.
6. Бидайбеков Е. Ы. Вычислительная информатика как основное направление фундаментализации подготовки учителей информатики / Е. Ы. Бидайбеков, Г. Б. Камалова // Педагогика және психология=педагогика и психология. – 2010. – №2. – С. 60-63
7. Балыкбаев Т. О. Роль курса «Теоретические основы информатики» в фундаментализации подготовки учителей информатики / Балыкбаев Т. О., Бидайбеков Е. Ы., Киселева Е. А. // Вестник КазНПУ им. Абая, серия «Физико-математические науки». – 2010. – № 2. – С. 98-103.

«ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ» В СИСТЕМЕ ФУНДАМЕНТАЛЬНОЙ ПОДГОТОВКИ УЧИТЕЛЕЙ ИНФОРМАТИКИ

Е. Ы. Бидайбеков, Е. А. Киселёва

Казахстан, г. Алматы, Казахский национальный педагогический
университет имени Абая
esen_bidaibekov@mail.ru

Республика Казахстан признана мировым сообществом как государство с рыночной экономикой, как страна, достигшая за короткий промежуток времени, после приобретения независимости, значительно роста экономики. Вхождение в мировое экономическое сообщество потребовало от системы образования кардинальных изменений. Поэтому образование и профессиональная подготовка кадров отнесены к числу основных приоритетов государственной политики. В связи с этим все большую значимость приобретают вопросы фундаментализации образования. Сегодня по-настоящему востребованным может быть только широко образованный человек, способный гибко перестраивать направление и содержание своей деятельности в связи со сменой технологий или требований рынка.

Проблемы фундаментализации предметной подготовки по информатике в педагогических вузах рассматривались в работах Е. Ы. Бидайбекова, Е. Н. Бобоновой, И. В. Левченко, И. Б. Готской, А. Р. Есяяна, К. К. Колина, Э. И. Кузнецова, В. В. Лаптева, М. П. Лапчика, Н. И. Рыжовой, А. Я. Фридланда, М. В. Швецкого и других ученых. Очевидно, что процессы фундаментализации образования не могут обойти стороной педагога. Только при наличии у него соответствующих профессиональных качеств учитель будет в состоянии обеспечить обучение школьников фундаментальным основам информатики, сформировать у них представление об информатике как о фундаментальной науке, приобщить к системному использованию знаний и умений в области информатики при изучении других дисциплин, а также в последующей трудовой деятельности. Таким образом, фундаментализация образования в соответствии с гуманистической парадигмой требует качественных изменений в профессиональной подготовке учителей информатики. Фундаментализация образования осуществляется на различных уровнях её функционирования, при этом имеет место следующая иерархия [1]:

Первый уровень – это фундаментализация всей системы образования как ядро и конечная цель новой образовательной парадигмы.

Второй уровень – это фундаментализация дисциплин предметной

подготовки, что подразумевает не простую совокупность традиционных курсов, а систему, объединённую общей целевой функцией, объектом исследования, методологией построения.

Анализ научно-методической литературы показал, что фундаментализация предметной подготовки будущих учителей информатики необходима по следующим причинам:

Нет чёткого разграничения между дисциплинами математического цикла, предмет которых не включает в себя природу систем и процессов, и информационными дисциплинами, связанными с изучением систем и процессов определённой природы – биологической, технической или социальной.

Учебный материал быстро теряет актуальность и постоянно требует замены более современным, при этом устаревает не только содержание, но и структура его изложения.

Отсутствие единого научного подхода к принципу отбора содержания дисциплин предметной подготовки, которое приводит к невозможности становления методики обучения информатике классическим путем, как для других учебных дисциплин.

Учитывая вышесказанное, считаем, что созрела необходимость в создании принципиально новых учебных курсов, ориентированных на формирование целостных современных представлений о научной картине мира в рамках предметной области «Информатика»; эти изменения должны быть внесены в ГОСО РК.

Третий уровень рассмотрения фундаментализации образования – это уровень отдельной дисциплины. При этом внутри предметного цикла она играет роль системообразующей науки, а на уровне дисциплины – самостоятельная область знания. В качестве такой дисциплины в предметной подготовке будущих учителей выступают «Теоретические основы информатики» (ТОИ).

Анализ публикаций по проблеме фундаментализации на уровне отдельной дисциплины в предметной подготовке будущих учителей информатики подвел нас к выводу, что фундаментальное знание – это знание, которое начинается с основ, то есть с самых глубоких содержательных обобщений. В качестве таких обобщённых элементов в области «Информатика» выступают информационные процессы, информационные модели и информационные основы управления. Именно эти понятия задают основные направления развития предметной подготовки будущих учителей информатики и должны быть реализованы в курсе «Теоретические основы информатики». На сегодняшний день имеется тенденция к преподаванию информатики как фундаментальной интегральной дисциплины, способствующей формированию научно-

информационного мировоззрения, стабильных навыков работы с информацией, способности и стремления адаптироваться к быстро меняющейся информационной среде деятельности. Включение в содержание образовательной области «Информатика» самостоятельного специализированного курса «Теоретические основы информатики» является необходимым условием для реализации идей фундаментализации вузовского и школьного образования по информатике.

Считаем, что в общем виде цель изучения данной дисциплины можно сформулировать как ознакомление студентов с набором фундаментальных концепций информатики, содействие развитию когнитивных моделей для этих концепций, поощрение развития у студентов навыков, необходимых для применения концептуальных знаний, которыми должен обладать будущий учитель информатики, а точнее, освоение общеобразовательного потенциала информатики, отражающего вклад в формирование современной системно-научной картины мира.

При формировании структуры и содержания дисциплины учитывались следующие принципы:

1) *принцип научности* обеспечивает достаточную глубину, корректность и научную достоверность изложения содержания учебного материала, с учетом последних научных достижений;

2) *принцип доступности* предполагает необходимость определения степени теоретической сложности и глубины изучения учебного материала сообразно возрастным и индивидуальным особенностям учащихся. Недопустима чрезмерная усложненность и перегруженность учебного материала, при которой овладение этим материалом становится непосильным для обучаемого;

3) *принцип фундаментализации содержания образования* требует осознания сущности познавательной и практической преобразующей деятельности. Обучение в этой связи предстает не только как способ получения знания и формирования умений и навыков, но и как средство вооружения методами добывания новых знаний, самостоятельного приобретения умений и навыков;

4) *систематичности и последовательности* означает обеспечение последовательного усвоения учащимися определенной системы знаний в изучаемой предметной области. Необходимо, чтобы знания, умения и навыки формировались в определенной системе, в строго логическом порядке и находили применение в жизни;

5) *принцип соответствия содержания образования требованиям развития общества, науки, культуры и личности*, предполагающей интеграцию традиционно необходимых знаний, умений и навыков и отражающих современный уровень развития общества, научного знания и

возможности личностного роста;

6) *принцип гуманизации содержания*, связанный с созданием условий для активного творческого и практического освоения студентами общечеловеческой культуры;

7) *принцип единой содержательной и процессуальной стороны обучения*, который предполагает учет особенностей конкретного учебного процесса. При отборе содержания дисциплины учитывались принципы и технологии передачи материала, уровни его усвоения и связанные с этим действия;

8) *принцип структурного единства содержания образования на разных уровнях* его формирования предполагает согласованность таких составляющих, как теоретическое представление, учебный предмет, учебный материал, педагогическая деятельность, личность обучаемого.

Учитывая все вышесказанное, предлагается следующее содержание дисциплины «Теоретические основы информатики»:

1. Предмет и задачи теоретической информатики.
2. Основы теории информации: кодирование информации, измерение информации; передача сигнала по каналу связи
3. Информационные системы и технологии.
4. Автомат как основной элемент информационных систем. Абстрактные автоматы.
5. Конечные автоматы; машины Поста и Тьюринга; легко и трудно разрешимые задачи; невычислимые функции; проблема останова; следствия невычислимости.
6. Функции, отношения и множества; основы логики; методы доказательства; основы вычислений; дискретная вероятность.
7. Представление информации в цифровых автоматах. Системы счисления.
8. Выполнение арифметических операций на двоичных сумматорах.
9. Цифровая логика и цифровые системы: обзор и история архитектуры ЭВМ, базовые аппаратные блоки, логические выражения.
10. Понятие алгоритма. Основные понятия теории алгоритмов.
11. Алгоритмы и поиск решений задач: стратегии поиска решений задач; роль алгоритмов в процессе поиска решений задач; понятие и свойства алгоритмов.
12. Алгоритмы и структуры данных.
13. Анализ эффективности и сложности алгоритмов. O-нотация, стандартные классы сложности.
14. Анализ алгоритмов поиска и выборки.
15. Анализ алгоритмов сортировок.
16. Анализ рекурсивных алгоритмов.

17. Недетерминированные алгоритмы.
18. Вероятностные алгоритмы.
19. Информационное моделирование.
20. Семантические основы информатики.

Представленное содержание курса ТОИ разработано в соответствии с международными рекомендациями по обучению информатике, учитывает опыт российских коллег, что обеспечивает возможность интеграции казахстанского образования в международную образовательную систему. Спецификой предложенной дисциплины является то, что она преподается как вводный курс. В соответствии с требованиями кредитной системы обучения рабочие учебные планы должны включать три группы дисциплин по степени обязательности и последовательности усвоения содержания образовательной программы: группа А – дисциплины, изучаемые обязательно и строго последовательно по времени; группа В – дисциплины, изучаемые обязательно, но, возможно, не последовательно; группа С – дисциплины, которые студенты изучают по выбору в любой академический период [2].

К группе А относятся вводные курсы, являющиеся типичными курсами начального уровня, которые предлагаются на первом или втором курсах обучения. Вводные курсы по информатике должны знакомить студентов с главными интеллектуальными аспектами дисциплины. Когда мы рассматриваем информатику как дисциплину, нам нужно отвлечься от распространенного восприятия информатики как инструмента и сконцентрироваться на ее концептуальных основах. Теоретическая информатика изучает общие закономерности получения, обработки и использования информации, принципы организации информационных процессов в технических и социальных системах, методологии создания и использования информационных моделей (независимо от природы информационных данных). Вводный курс, при кредитной форме обучения, дает студентам представление о целом ряде интересных и важных тем, вместо того, чтобы сразу погружать их в детали одной конкретной области. Студенты, заинтересовавшиеся той или иной областью, могут двигаться дальше к любому другому циклу, более подробно излагающему свой предмет.

Курс «Теоретические основы информатики» должен обеспечить студентам развернутое введение в информатику, связывающее воедино идеи теории информации, программирования, дискретной математики, основ архитектуры ЭВМ, теории алгоритмов, теории вычислимости, анализа алгоритмической сложности и др.

При составлении курса, ориентированного на максимальный охват материала, существуют проблемы, которые необходимо учитывать. Пе-

рвая заключается в том, что дискретная математика должна рассматриваться как неотъемлемая часть курса. При таком подходе студенты смогут лучше понять и оценить важность дискретной математики в информатике. Например, булевская логика может быть рассмотрена в процессе обсуждения операторов языка, алгоритмы счета могут рассматриваться при обсуждении эффективности итеративных алгоритмов, а рекуррентные соотношения вполне естественно изучать применительно к вопросу эффективности рекурсивных алгоритмов. Цель состоит в том, чтобы студенты получили представление о математических понятиях в контексте их использования при решении важных вычислительных задач.

Вторая проблема состоит в том, что многие малосвязанные темы, обычно встречаемые в программе курса, ориентированного на максимальный охват материала, должны быть интегрированы в единое целое. Необходимо развить в студентах понимание сложных взаимосвязей между основными областями информатики. Эта цель может быть достигнута путем демонстрации того, как каждая из тем курса использует идеи из уже пройденного материала и на их основе получает новые и более мощные абстракции. Такой «спиральный» подход является важной составляющей успеха курса.

Специфика подготовки будущих учителей информатики заключается в необходимости формирования, прежде всего, системообразующих концептуальных знаний, которые давали бы общее видение научно-образовательной области «Информатика». Вводный курс «Теоретические основы информатики» лучше всего подходит для решения этой проблемы.

Литература

1. Леднев В. С. Научное образования : развитие способностей к научному творчеству / Леднев В. С. – М. : МГАУ, 2002. – 120 с.
2. Организация образовательного процесса на основе кредитной технологии / Сост. : Н. А. Асанов, Д. А. Калдияров, Г. С. Минажева. – Алматы : Қазақ университеті, 2004. – 59 с.

ЗМІСТОВНО-СТРУКТУРНА МОДЕЛЬ МЕДИЧНОЇ ІНФОРМАТИКИ НА ДОДИПЛОМНОМУ ЕТАПІ ПІДГОТОВКИ МАЙБУТНЬОГО ЛІКАРЯ

Л. П. Войтенко

Україна, м. Київ, Національний медичний університет
імені О. О. Богомольця
Latl@meta.ua

Зміст освіти належить до найважливіших педагогічних категорій. Саме тому питання визначення компонентів, критеріїв і процедур відбору змісту навчальної дисципліни відноситься до числа традиційно актуальних питань світової та вітчизняної професійної освіти.

Зміст дисципліни «Медична інформатика», як і будь-якої іншої дисципліни, детермінується кінцевими цілями підготовки майбутнього лікаря, представлених у формі освітньо-кваліфікаційної характеристики та освітньо-професійної програми підготовки фахівців напряму «Медицина». Разом з тим він підлягає інтенсивному впливу стрімкого розвитку інформаційно-комунікаційних технологій в умовах інформатизації суспільства, що спричиняє необхідність розроблення структури і змісту медичної інформатики, яка б здійснила фундаментальний внесок в науково-теоретичну та практичну підготовку студента-медика.

У попередніх публікаціях ми виокремили основні передумови введення медичної інформатики у зміст професійної освіти майбутніх лікарів, дослідили місце і значення цієї навчальної дисципліни в системі вищої медичної освіти, обґрунтували окремі компоненти її методичної системи, зокрема, цілі та зміст, дослідили еволюцію змісту медичної інформатики [1; 2; 3; 4].

Узагальнюючи ці дослідження, вважаємо, що змістовно-структурну модель дисципліни «Медична інформатика» можна представити у вигляді концентричних кіл (рис. 1). Ядром цієї моделі (внутрішнє кільце) ми визначили фундаментальні основи інформатики, адже успішне засвоєння дисципліни «Медична інформатика» студентами-медиками можливе лише за наявності відповідних базових знань з основ інформатики, отриманих під час вивчення в загальноосвітній школі однойменної навчальної дисципліни «Інформатика» (освітня галузь «Технології»). Фундаментальні основи інформатики у цій моделі включають наступні домени знань: формалізація та алгоритмізація медичних задач, формальна логіка у вирішенні задач діагностики, лікування та профілактики захворювань та бази даних у медицині.

Наступне коло моделі – клінічна інформатика, розвиток якої спри-

чинений впровадженням засобів ІКТ в клінічну медицину для діагностики, визначення тактики лікування, трактування результатів обстежень тощо. У запропонованій моделі підготовки клінічна інформатика об'єднує такі домени: кодування і класифікація даних у медицині, візуалізація медико-біологічних даних, аналіз біосигналів та методи їх обробки, індивідуальні медичні картки.



Рис. 1. Змістовно-структурна модель дисципліни «Медична інформатика»

Зовнішнє коло моделі – інформатика в галузі охорони здоров'я, що включає знання майбутнього лікаря з доказової медицини, методів підтримки прийняття рішень, госпітальних інформаційних систем, інформаційних ресурсів в системі охорони здоров'я, етичні та правові принципи управління інформацією в системі охорони здоров'я.

Експериментальна перевірка запропонованої моделі здійснювалася з метою встановлення її відповідності вимогам галузевих стандартів вищої (медичної) освіти (ГСВО) на основі дослідження кількісних оцінок показників інформатичної підготовки та фахової компоненти базової вищої освіти майбутнього лікаря.

Вважатимемо, що:

– кількісним показником інформатичної підготовки майбутнього

лікаря є оцінка з дисципліни «Медична інформатика»;

– кількісним показником фахової компоненти базової вищої освіти майбутнього лікаря є результат складання медичного ліцензійного інтегрованого іспиту «Крок 1. Загальна лікарська підготовка»;

– відповідність змісту інформатичної підготовки майбутнього лікаря вимогам ГСВО – це показник, який чисельно характеризує ступінь прогнозу успішності оволодіння фаховою компонентою базової вищої освіти майбутнього лікаря на основі успішності вивчення дисципліни «Медична інформатика»;

– відповідність змісту інформатичної підготовки майбутнього лікаря вимогам ГСВО – це прогностична валідність успішності вивчення дисципліни «Медична інформатика».

Експеримент проводився протягом 2006-2008 рр. у Національному медичному університеті імені О. О. Богомольця на чотирьох медичних факультетах. Експериментом було охоплено 596 студентів-медиків другого року навчання (IV навчальний семестр), що вперше у 2006 р. вивчали оновлену за змістом дисципліну «Медична інформатика» відповідно до нової навчальної програми дисципліни [5] та 12 викладачів кафедри медичної інформатики та комп'ютерних технологій навчання університету, які проводили заняття з дисципліни.

Основним аналітичним методом, який використовувався у дослідженні, є кореляційний аналіз статистично значимих вибірок даних.

Характеристика методики дослідження. Методика експерименту передбачала дослідження прогностичної валідності успішності вивчення оновленої за змістом дисципліни «Медична інформатика».

Прогностична валідність оцінюється коефіцієнтом кореляції між результатом поточної методики оцінювання вимірюваної властивості та результатом, визначеним за певним зовнішнім критерієм (зовнішньою змінною), що характеризуватиме вимірювану властивість у майбутньому [6; 7; 8]. Саме тому в нашому дослідженні ми оцінили кореляцію показника успішності студентів-медиків з оновленої за змістом дисципліни «Медична інформатика», що вивчалася студентами медичних факультетів у IV навчальному семестрі з показником успішності складання медичного ліцензійного інтегрованого іспиту «Крок 1. Загальна лікарська підготовка» цими ж студентами у VII навчальному семестрі.

При інтерпретації результатів кореляційного аналізу ми скористалися практичним правилом, запропонованим Коеном (Cohen, 1988) та використали у дослідженнях якості системи вступу до ВНЗ України на основі зовнішнього незалежного оцінювання випускників загальноосвітніх навчальних закладів [9]:

– сильна стохастична залежність, якщо коефіцієнт кореляції біль-

ше 0,5;

- достатня стохастична залежність, якщо коефіцієнт кореляції знаходиться в межах інтервалу [0,3; 0,5];

- низька стохастична залежність, якщо коефіцієнт кореляції менше 0,3.

База даних дослідження. База даних дослідження отримана в результаті інтеграції результатів складання медичного ліцензійного іспиту «Крок 1. Загальна лікарська підготовка» Центру тестування професійної компетентності при МОЗ України з базою даних успішності студентів з дисципліни «Медична інформатика» кафедри медичної інформатики та комп'ютерних технологій навчання Національного медичного університету імені О. О. Богомольця.

Медичний ліцензійний іспит (МЛІ) «Крок 1. Загальна лікарська підготовка» («Крок 1. ЗЛП») вимірює показник якості фахової компоненти базової вищої освіти майбутнього лікаря та є засобом її діагностування. Іспитом оцінюється вміння використовувати знання та розуміння ключових понять фундаментальних біомедичних наук, головну увагу приділяючи принципам та механізмам, що лежать в основі здоров'я, хвороби та моделей лікування [10, 11]. Структура іспиту сформована на основі ОКХ спеціаліста напряму підготовки «Медицина». Тестові завдання складаються із ситуаційної задачі, запитання або твердження [12].

При оцінюванні поточної навчальної діяльності з дисципліни використовувалися стандартизовані методи оцінки теоретичної підготовки студентів (тестовий контроль) та контроль практичної підготовки. При проведенні підсумкового модульного контролю з медичної інформатики ми використовували стандартизовані методи оцінки: для перевірки теоретичної підготовки студентів з дисципліни – тестування, для перевірки практичної підготовки – структурований за процедурою контроль практичних навичок з використанням комп'ютерної техніки.

Підготовка даних дослідження. На основі журналів обліку успішності студентів з дисципліни ми створили базу даних успішності студентів з медичної інформатики.

Наступним кроком створення бази даних дослідження стала інтеграція бази даних про успішність студентів з дисципліни «Медична інформатика» із звітами за результатами складання МЛІ «Крок 1. ЗЛП», які нам були надані Центром тестування професійної компетентності при МОЗ України.

Інтегрована база дослідження включає в себе:

- особисті дані студента (факультет, група, стать, форма навчання (бюджет, контракт), середній бал успішності навчання у ВНЗ);

- вхідний (базовий) рівень знань студентів-медиків з загальноосвіт-

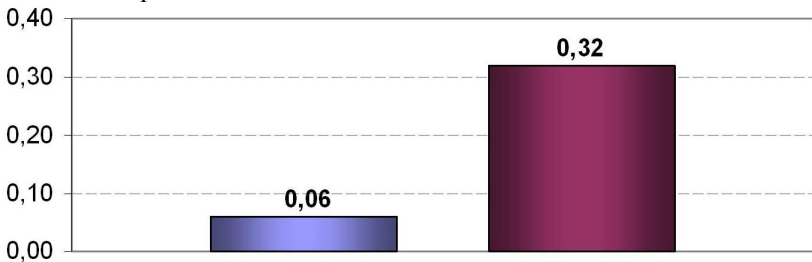
нього предмету «Інформатика», методика визначення якого описана у [13];

– успішність вивчення дисципліни «Медична інформатика» (поточний контроль, підсумковий модульний контроль, сума балів за модуль, оцінка з дисципліни);

– результати складання медичного ліцензійного інтегрованого іспиту «Крок 1. ЗЛП».

Результати дослідження. Кореляційний аналіз даних проводився з використанням засобів статистичного аналізу даних «Пакет аналізу» електронних таблиць Excel. Коефіцієнт кореляції розраховувався за формулою Пірсона, оскільки вибірки, які досліджувалися, розподілені за нормальним законом. Висновки про статистичну достовірність розрахованого коефіцієнту кореляції робились за t -критерієм Стьюдента на рівні значущості $p = 0,05$.

На рис. 2 наочно представлені значення коефіцієнтів кореляції успішності з дисципліни «Медична інформатика» та результату складання МЛІ «Крок 1. ЗЛП» та значення коефіцієнтів кореляції вихідного рівня знань з загальноосвітнього предмету «Інформатика» та результату складання МЛІ «Крок 1. ЗЛП».



- Значення коефіцієнта кореляції між оцінкою базового рівня знань з загальноосвітнього предмету "Інформатика" та результатом МЛІ "Крок 1. ЗЛП"
- Значення коефіцієнта кореляції між успішністю з дисципліни "Медична інформатика" та результатом МЛІ "Крок 1. ЗЛП"

Рис. 2. Значення коефіцієнта кореляції

Результати кореляційного аналізу засвідчують:

– наявність достатнього стохастичного зв'язку між оцінкою з дисципліни «Медична інформатика» та результатами складання МЛІ «Крок 1. ЗЛП»;

– відсутність стохастичного зв'язку між вхідним рівнем знань з загальноосвітнього предмету «Інформатика» та результатами складання

МЛП «Крок 1. ЗЛП».

Інтерпретуючи результати кореляційного аналізу даних, констатуємо достатню прогностичну валідність показника успішності вивчення дисципліни «Медична інформатика» на показник фахової компоненти базової вищої освіти.

Продовжимо наше дослідження у розрізі спеціалізацій підготовки – за факультетами (табл. 1).

Таблиця 1

Значення коефіцієнтів кореляції

Факультет	Значення коефіцієнта кореляції між оцінкою з дисципліни «Медична інформатика» та результатом МЛП	Значення коефіцієнта кореляції між результатом базового рівня знань з загальноосвітнього предмету «Інформатика» та результатом МЛП
Медичний №1	0,21	0,19
Медичний №2	0,32	0,15
Медичний №3	0,33	0,07
Медичний №4	0,52	0,08

Інтерпретуючи результати кореляційного аналізу, можемо констатувати:

1. Наявність високої кореляції між показником успішності з дисципліни «Медична інформатика» та показником успішності складання МЛП «Крок 1. ЗЛП.» на медичному факультеті №4.

2. Наявність достатньої кореляції між показником успішності з дисципліни «Медична інформатика» та показником успішності складання МЛП «Крок 1. ЗЛП.» медичних факультетах №2 та №3.

3. Наявність низької кореляції між показником успішності з дисципліни «Медична інформатика» та показником успішності складання МЛП «Крок 1. ЗЛП» на медичному факультеті №1.

Отримавши результати за факультетами, які досить сильно різняться, зазначимо, що всі чотири факультети навчаються за програмою загальної лікарської підготовки, але за різними спеціальностями: медичний №1, 2 – «лікувальна справа», медичний № 3 – «педіатрія», медичний №4 – «медико-профілактична справа». Це зумовлює різний престиж факультетів, відповідно до майбутньої професії, різний конкурс при вступі, різні дисципліни, які складаються абітурієнтами при вступі, і, як наслідок, різний контингент студентів.

Для подальшого дослідження ми вибрали факультети, які показали діаметрально-протилежні результати – медичний факультет №1 та №4. Ми вважаємо, що отримані результати можна пояснити наявністю ста-

тистично-достовірних відмінностей між вибірками.

Для перевірки наявності статистично-достовірних відмінностей між вибірками формуємо нульову гіпотезу H_0 про те, що різниця між генеральними параметрами порівнюваних вибірок студентів медичного факультету №1 (M1) та студентів медичного факультету №4 (M4) дорівнює нулю і що відмінності, які ми спостерігаємо між вибірковими характеристиками, носять не систематичний, а виключно випадковий характер.

Для перевірки висуненої гіпотези, а отже, і достовірності оцінки генеральних параметрів за вибірковими даними, використаємо у якості критерію достовірності t -критерій Стьюдента на рівні значущості $p \geq 0,5$ [14].

Дані статистичної обробки та перевірки гіпотези ймовірних розбіжностей результатів студентів медичного факультету №1 та №4 відображені у таблиці табл. 2. Обробка даних проводилася з використанням засобів Пакету статистичного аналізу даних «Двухвыборочный t -тест с разными дисперсиями» електронних таблиць Excel.

Таблиця 2

Статистичні характеристики порівнюваних вибірок

Контингент	M1	M4
Статистичні параметри		
Об'єм вибірки (n)	170	114
Середнє значення (\bar{x})	156,0	144,4
Дисперсія (σ)	16,3	16,2
Значення t -критерію (t_d)	5,90	

Для числа ступенів свободи $\nu = 282$ значення стандартного критерію Стьюдента на рівні значущості $p \geq 0,5$ становить $t_{st} = 1,96$. Оскільки $t_d > t_{st}$, то нульова гіпотеза H_0 спростовується з рівнем значущості $p \geq 0,5$. Таким чином, ми довели, що відмінності між вибірками є статистично достовірними.

Ймовірним поясненням наявності статистично-достовірних відмінностей між вибірками може бути відмінності між контингентом студентів на кожному з факультетів: контингент студентів медичного факультету №4 формується значною мірою з цільового набору осіб з сільської місцевості, де рівень підготовки з загальноосвітнього предмету «Інформатика» є нижчим в порівнянні з іншими студентами. Саме тому при вивченні медичної інформатики ці студенти фактично вирівнюють свої знання з базової інформатики, опановуючи паралельно медичну інформатику. Аргументом на користь висловленого припущення може виступати той факт, що середня оцінка з дисципліни «Медична інформатика»

у студентів медичного факультету №4 нижча за середню оцінку студентів медичного факультету №1 з цієї ж дисципліни (рис. 3).

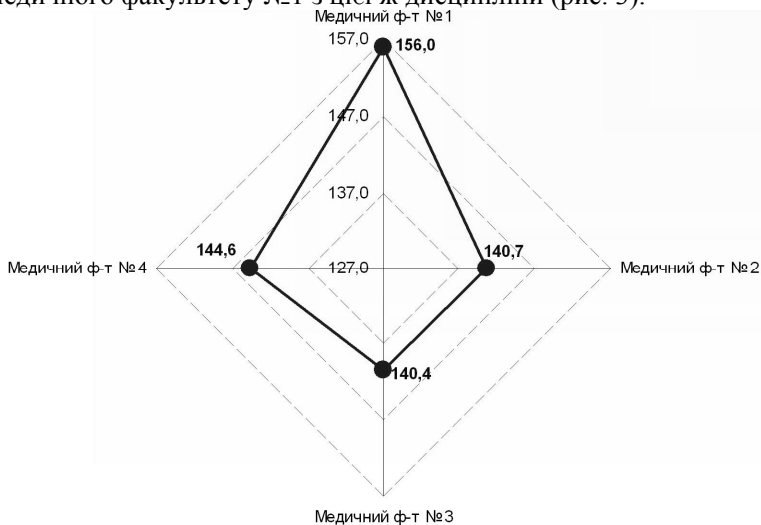


Рис. 3. Діаграма успішності з дисципліни «Медична інформатика» за факультетами

Отримані результати статистичного аналізу підтверджують правильність гіпотези щодо ефективності змістовно-структурної моделі змісту дисципліни «Медична інформатика» та її відповідність вимогам Галузевим стандартам вищої (медичної) освіти.

Проведене дослідження дозволяє зробити висновок про педагогічну доцільність апробованої моделі та зумовлює її подальшу реалізацію при вивченні медичної інформатики на до дипломному етапі підготовки лікарів.

Література

1. Войтенко Л. П. Медична інформатика у змісті професійної освіти майбутніх лікарів / Л. П. Войтенко // Теорія та методика навчання математики, фізики, інформатики: збірник наукових праць. Випуск ІХ. – Кривий Ріг : Видавничий відділ НМетАУ, 2011. – С. 430-436.
2. Войтенко Л. П. Обґрунтування окремих компонентів методичної системи навчання медичної інформатики у ВМНЗ / Л. П. Войтенко // Вища освіта України у контексті інтеграції до європейського освітнього простору / гол. ред. В. Андрущенко. – К. : Інститут вищої освіти АПН України. – Том ІІІ (21). Новітні засоби навчання: проблеми впровадження та стандартизації. – К., 2009. – С. 42-50.
3. Булах І. Є. Еволюція інформатичної підготовки лікарів на доди-

пломному етапі навчання / І. Є. Булах, М. Р. Мруга, Л. П. Войтенко // Перший Всеукраїнський з'їзд «Медична та біологічна інформатика і кібернетика» з міжнародною участю : збірник праць, м. Київ, НМАПО імені П. Л. Шупика, 23-26 червня 2010 р. – К., 2010. – С. 95.

4. Войтенко Л. П. Становлення та розвиток інформатичної освіти у системі вищої медичної освіти: історичний аспект проблеми / Л. П. Войтенко // Теорія та методика навчання математики, фізики, інформатики: збірник наукових праць. Випуск VIII : в 3-х томах. – Кривий Ріг : Видавничий відділ НМетАУ, 2010. – Т. 3. – С. 26-32.

5. Медична інформатика. Програма навчальної дисципліни для студентів навчальних закладів III-IV рівнів акредитації України. Спеціальності 7.110101 «лікувальна справа», 7.110104 «педіатрія», 7.110105 «медико-профілактична справа», 7.110106 «стоматологія». – К. : Друкарня НМУ, 2005. – 27 с.

6. Короткий текстологічний словник-довідник / упорядник Л. Т. Коваленко – К. : Грамота, 2008. – 160 с.

7. Словарь-справочник по психологической диагностике / Бурлачук Л. Ф., Морозов С. М. ; отв. ред. Крымский С. Б. – К. : Наук. думка, 1989. – 200 с.

8. Булах І. Є. Теорія комп'ютерного тестування / І. Є. Булах. – К. : ЦМК МОЗ України, 1994. – 59 с.

9. Моніторингове дослідження якості системи формування контингенту студентів ВНЗ України на основі ЗНО. Звіт за проектом 400777 / Міжнародний фонд «Відродження», Центр тестових технологій і моніторингу якості освіти // Вісник ТІМО. – 2010. – №4-5. – С. 14-47.

10. Моніторинг якості медичної освіти / Під заг. ред. І. Є. Булах. – К. : Центр тестування при МОЗ України, 2005. – 143 с.

11. Наказ МОЗ України від 31.01.2005 р. № 52 «Про затвердження та введення нового навчального плану підготовки фахівців освітньо-кваліфікаційного рівня «спеціаліст» кваліфікації «лікар» у вищих навчальних закладах III-IV рівнів акредитації України за спеціальностями «лікувальна справа», «педіатрія», «медико-профілактична справа»».

12. «Крок 1» – Лікувальна справа : навчально-методичний посібник. – К. : ЦМК МОЗ України, 1998. – 34 с.

13. Войтенко Л. П. Наступність інформатичної підготовки в системі вищої медичної освіти / Л. П. Войтенко // Вища освіта України – Додаток 2 до № 3, том II (27) – 2010. – Тематичний випуск «Вища освіта України у контексті інтеграції до європейського освітнього простору». – К. : Інститут вищої освіти НАПН України. – С. 123-129.

14. Лакин Г. Ф. Биометрия. – Издание четвертое, переработанное и дополненное / Лакин Г. Ф. – М. : Высшая школа, 1990. – 352 с.

3D-МОДЕЛЮВАННЯ ЯК ЗАСІБ ПІДВИЩЕННЯ ПІЗНАВАЛЬНОЇ АКТИВНОСТІ ШКОЛЯРІВ

В. О. Воронов

Україна, м. Шепетівка, Шепетівський навчально-виховний комплекс №1
voron@lenta.ru, val_voron@mail.ru

Основною проблемою сучасною школи є підвищення пізнавальної активності школярів, виховання та розвиток творчої особистості. Сучасний спеціаліст – це людина, яка володіє сучасними інформаційними технологіями. Вивчення технологій у старшій школі спрямоване на підготовку молодої людини до вступу в доросле життя, що потребує не лише наявності знань шкільних наук, але й умінь та здатності використовувати знання на практиці, самостійно вирішувати нестандартні життєві проблеми [3].

Різноманітні аспекти впровадження інформаційно-комунікаційних технологій у навчальний процес привертала увагу багатьох дослідників. Дидактико-педагогічні й методичні проблеми інформатизації навчального процесу вивчали В. Г. Болтянський, В. П. Беспалько, М. І. Жалдак, В. М. Монахов, Ю. С. Рамський, В. Г. Розумовський, О. В. Співаковський та ін.

Незважаючи на дослідження засобів комп'ютерного моделювання багатьма дослідниками, питання використання засобів інформаційних технологій у формуванні професійного спрямування та використання навиків об'ємного моделювання у формуванні пізнавальної активності школярів поки що не знайшли належного осмислення й трактування в нинішніх дослідженнях та вимагають додаткового вивчення.

Сучасні інформаційні технології не розкривають повною мірою свого навчального потенціалу в традиційній освітній системі, де домінують дидактичні лінійні технології передавання готових знань, оскільки стрімке зростання інформаційних потоків об'єктивно не дозволяє повністю реалізувати принцип передавання всіх накопичених знань у процесі навчання. У зв'язку з цим інформаційні технології спрямовуються на нелінійну структурування навчального процесу, яка створює умови для розвитку в учнів умінь та навичок постановки задач, моделювання, оптимізації, прийняття рішень в умовах невизначеності, вміння самостійно здобувати знання.

Зокрема, до нелінійних технологій належить комп'ютерне моделювання як основний метод пізнання. Головною особливістю нелінійних технологій є постановка навчальних цілей і задач у предметній галузі, що дозволяють педагогу сформулювати експертний шлях їх досягнення і

розв'язання, запропонувати необхідний інструментарій, методичний матеріал, інструкції, досвід.

Моделювання як метод пізнання є центральним в освітньому процесі, де використовуються комп'ютерні технології [4].

3D-графіка – це створення віртуальної тривимірної моделі за допомогою 3D-програм. Керуючись технічним завданням на основі побажань замовника, 3D-модельєр створює об'ємну модель будь-якого об'єкта. У спеціальній 3D-програмі об'єкт можна вивчити із усіх ракурсів і помістити його в будь-яке навколишнє середовище. 3D-моделювання стає усе більш популярним у сучасній житті. Якщо раніше було важко уявити, як буде виглядати той або інший об'єкт, то з появою 3D-графіки стало можливим створення повноцінного об'ємного об'єкта майбутньої споруди. Сучасні технології в 3D-графіці дозволяють домагатися вражаючої реалістичності.

3D-моделювання й 3D-графіка широко застосовуються в багатьох сферах. 3D-графіка широко використовується при проектуванні будинків і споруд. Це може бути модель майбутнього будинку, як приватного котеджу, так і багатоповерхового будинку. Крім того, візуалізація активно застосовується в дизайн проектах інтер'єрів.

Програми й технології тривимірного моделювання широко застосовуються й у промисловості, наприклад, у виробництві корпусних меблів. Багато конструкторів і дизайнери вже давно перейшли від використання лінійки й олівця до сучасних тривимірних комп'ютерних програм, таким як 3ds Max, Lightwave, Arcon і Bryce. Поступово нові технології освоюють і інші компанії, насамперед, виробничі й торговельні [1].

Темпи технологічного розвитку у світі спонукають активне впровадження сучасних технологій і змушують усвідомлювати, що також відбувається принципова зміна технологій розумової праці на основі комп'ютеризованих інформаційних систем. Професійна культура працівника 21 століття має уже включати складову інформаційної культури.

Інформаційна культура особистості є інтегральним показником рівня її досконалість в інформаційній сфері діяльності. Серед різноманітних форм та методів інноваційного навчання особливе місце належить дидактичним іграм. Концептуальними основами запровадження ігрових технологій у навчанні є психологічні механізми ігрової діяльності, що спираються на фундаментальні потреби особистості у самовираженні, самоутвердженні, самовизначенні, саморегуляції, самореалізації. Дидактичні ігри розрізняють за цільовими орієнтаціями, характером педагогічного процесу, ігровою методикою тощо.

При використанні комп'ютера в основній школі ігрову діяльність можна використовувати, щоб реалізувати такі функції:

- спонукальну (викликає інтерес);
- комунікативну (засвоєння елементів культури спілкування);
- самореалізацій (кожен учень реалізує свої можливості);
- розвивальну (розвиток уваги, волі та інших психічних якостей);
- діагностичну (виявлення прогалин у знаннях);
- корекційну (внесення позитивних змін у структуру особистості учнів та їх систему знань);
- розважальну (отримують задоволення).

Звернемо увагу на той факт, що інноваційні навчальні технології спрямовані не лише на засвоєння школярами нових фактів, а на їх відкриття, на пошукову та дослідницьку діяльність учнів. Організацію такої діяльності в процесі вивчення комп'ютерного моделювання можна забезпечити на основі використання ППЗ bCAD Студент.

Тому при виборі варіативного модуля навчання для технологій було вибрано «Основи комп'ютерного моделювання». Програмне забезпечення, яке було вибрано для практичного опрацювання, – bCAD Студент.

bCAD – це інтегрований пакет для двохмірного креслення, об'ємного моделювання та реалістичної візуалізації. Дане програмне забезпечення широко розповсюджене в мебельному виробництві та дизайні інтер'єрів. В даному пакеті є чудова безкоштовна локалізація bCAD Студент, яка дозволяє початківцям отримати досить серйозні навички в 3D-моделюванні. Враховуючи доступність програмного забезпечення та регіональну потребу в професійних кадрах даної направленості, було зупинено вибір на даному інтегрованому пакеті.

Досить докладну інформацію можна отримати на сайті компанії «ПроПро Група» (<http://www.propro.ru>), яка займається розробкою даного програмного забезпечення. На сайті можна знайти різні версії даної програми та скачати bCAD Студент, що є повністю безкоштовним програмним продуктом. Дана програма має різноманітні мовні інтерфейси, в тому числі російською мовою, тому для сучасної молоді не становить труднощів в освоєнні інтерфейсу даної програми. Встановлення та запуск програми досить прості [2].

Як показує практика роботи з даною програмою, через 2-3 ознайомчі заняття учні успішно засвоюють основні навички роботи і створюють досить непогані дво- та тривимірні моделі. На нашу думку, запропонований пакет сприяє певному вибору майбутньої сфери діяльності.

bCad надає можливість забезпечувати кресленнями виробів учнів, виконувати та розробляти різноманітні графічні завдання. При потребі можна легко виправити помилки або змінити деякі конструктивні елементи деталей. Під час практичної діяльності в учнів автоматично зні-

маються багато запитань по формі та конструктивних елементах деталей, хоча це все і обговорювалося раніше. Учні, які пропустили заняття, мають можливість самостійно ознайомитися з кресленням виробу, а також отримати консультацію у своїх товаришів. Якщо і далі виникають запитання, то на них відповідає учитель. Таким чином, учні працюють більш самостійно під час фронтальної практичної роботи при опануванні технологічних прийомів роботи.

Основне завдання використання засобів 3D-моделювання – розвинути в учнів творчі здібності, логічне мислення і свободу у використанні інформаційних технологій. Все це пропонується розвинути шляхом залучення учнів до процесу створення різних моделей.

Саме можливість створювати різні проекти засобами 3D-моделювання дозволяє залучати учнів до використання проектної технології у різних галузях виробництва та сферах життєдіяльності, у будь-якому виді технологічної діяльності, бізнесу і обслуговування. Під час виконання творчих проектів у учнів формуються вміння творчо та критично мислити, працювати з різними інформаційними джерелами, Інтернет-технологіями, здійснювати дослідницьку роботу. Учні навчаються свідомо планувати власну діяльність для досягнення поставлених завдань.

Література

1. Головна - Henergy [Електронний ресурс]. – Режим доступу : <http://www.henergy.kiev.ua/index.html>
2. bCAD Студент. Бесплатная версия для обучения [Электронный ресурс] // ПроПро Група 1992-2012. – 2012. – Режим доступа : <http://www.propro.ru/ru/Products/bCAD/student.aspx>
3. Технології. 10 клас : підручник / О. М. Коберник, А. І. Терещук, О. Г. Гервас, О. Б. Авраменко, С. М. Ящук, В. В. Бербец ; загальна редакція – А. І. Терещук. – К. : Літера, 2010. – 160 с.
4. Шмиголь Ю. В. Інформаційні технології в розвитку пізнавальної активності студента / Ю. В. Шмиголь, А. В. Калініченко, А. К. Костоглод // Матеріали V Міжнародної науково-технічної конференції «Комп'ютерні технології в будівництві» : Київ–Севастополь, 18-21 вересня 2007 р. – Кривий Ріг, 2008. – С. 106-107.

ДЕЯКІ АСПЕКТИ МОБІЛЬНОГО НАВЧАННЯ ПРОГРАМУВАННЯ У VBA

Я. М. Глинський, В. А. Рязьська

Україна, м. Львів, Національний університет «Львівська політехніка»
ya_hlynsky@mail.lviv.ua

Мобільну освіту розглядають як частину відкритої дистанційної освіти, що реалізується за допомогою мобільних технічних засобів, темпи поширення яких інтенсивно нарощуються [1]. В мобільній освіті виокремлюють апаратну мобільність та програмну мобільність [2]. Програмна мобільність часто базується на концепції «хмарних обчислень», що реалізується за допомогою різних web-сервісів. Прикладом веб-сервісу, який ефективно можна застосувати в мобільному навчальному процесі, є сервіс Microsoft Skydrive. Студентам від тепер не потрібні флешки для зберігання і просторового переміщення своїх файлів, оскільки їх можна зберігати на «хмарі» Skydrive. Більше того, тут вони можуть доопрацювати свої текстові документи, розрахункові та курсові роботи і звіти засобами вільнодоступних, але дещо урізаних легальних версій MS Word чи MS Excel. Це особливо актуально, якщо на серверах навчального закладу таких офісних програм немає з відомих причин.

Іншим прикладом ефективного застосування мобільного навчання є розміщення навчального програмного забезпечення на відеосервісі YouTube. У YouTube на каналі hlynsky1 ми розташували навчальне програмне забезпечення у форматах mpeg4 та avi (відеофільми) з метою навчання учнів та студентів основам алгоритмізації і програмування у середовищі VBA і отримали значний педагогічний ефект.

Відеофільми дали змогу автоматизувати процес навчання шляхом перерозподілу навчального часу на користь позааудиторної самостійної роботи учнів і студентів. Оскільки тематика відеофільмів пов'язана з прагматичними питаннями роботи в програмних середовищах, вдалося вивільнити аудиторний час на користь висвітлення теоретичних питань алгоритмізації і програмування, що націлено на підвищення рівня фундаменталізації навчання інформатики в загальноосвітній і вищій школах.

За допомогою сучасних технічних засобів (ноутбуків, нетбуків і смартфонів) навчання за методикою відеофільмів може здійснюватися як у стаціонарному, так і, що особливо важливо, у мобільному режимах. Безсумнівною перевагою такого навчання є можливість повторного перегляду відеофільмів як в географічно іншій точці, так і в зручний час, що забезпечує принципи мобільності навчання в часі та в просторі.

Комп'ютерні відеофільми, як і класичні навчальні фільми з різних

предметів, які показували в кінозалах у навчальних закладах 40-50 років тому, є ефективною формою демонстраційного навчання і важливим елементом наповнення сучасного електронного навчально-методичного комплексу. Вдало створені відеофільми можуть привести до перегляду ролі і значення традиційного електронного навчального посібника в мобільному навчанні на користь твердої копії. Якщо в системі дистанційної освіти концепція електронного навчального посібника актуальна, то в системі мобільної освіти вона має дещо менше значення. У мобільній освіті важливо, щоб час взаємодії суб'єкта, що навчається, з мобільним засобом був обмеженим і максимально ефективним. Інтерактивність навчального засобу стає менш важливою, оскільки в умовах мобільності не має змоги концентрувати на ній увагу. Сучасні електронні підручники, створені за всіма вимогами теорії [3], часто цій вимозі не відповідають, а ось відеофільми відповідають якнайкраще, оскільки їх тривалість зазвичай обмежена часом 6-10 хв. Якщо відеофільми достатньо повно охоплюють теми навчання, то вони можуть бути ефективним доповнювальним засобом до твердої копії навчального посібника (підручника), що веде до зростання ролі класичного підручника. Не можна не погодитися з твердженням, що традиційний підручник є ядром системи засобів навчання [3]. Підручник – це складна багатofункціональна система, що поєднує відтворення змісту навчання і реалізацію методики його опанування. Підручник має забезпечувати сходження особистості до все більш високих ступенів розвитку власного потенціалу, розширення поля пізнавальних проблем, актуальних для неї і доступних для самостійного вивчення [3].

Однак останнім часом спостерігається невміння, а часто просто небажання, учнів та студентів, працювати з підручниками і навчальними посібниками. Причиною цього явища є бурхливий розвиток сучасних інформаційно-комунікаційних технологій і різних альтернативних засобів і способів навчання. Власне відеофільми і є такими альтернативними засобами, які можуть виправдати покладені на них очікування.

Розглянемо зміст наших відеофільмів. Ми розвиваємо тему, підняту в [4], а саме: акцентуємо на поверненні мови Visual Basic в навчальний процес під час вивчення розділу алгоритмізації та програмування в курсі базової інформатики. Ми зосереджуємо увагу читачів і глядачів на сердовище VBA і це не випадково. Це результат багаторічних спостережень за тенденціями в освіті й головно за інтересами і можливостями сучасних учнів і студентів. Для студентів, які планують займатися програмуванням на професійному рівні, C# і Java – без сумніву кращий вибір. Тому як суб'єктів навчання за нашою методикою ми розглядаємо студентів некомп'ютерних напрямів підготовки (зокрема студентів педагогіч-

них ВНЗ), а також учнів загальноосвітніх шкіл.

Власне VBA з найменшими затратами дає змогу реалізувати основні стилі програмування: процедурне, візуальне, об'єктне. Більше того, можна говорити про четвертий стиль, притаманний тільки VBA, – це різновид візуального програмування безпосередньо в документі офісної програми, наприклад, MS Word чи MS Excel тощо.

Продемонструємо методику ознайомлення учнів та студентів з цими стилями програмування на прикладі розв'язування модельної задачі.

Відкриємо програму MS Word і запишемо умову задачі.

Задача про клумбу. Задано радіус клумби. Обчислити її площу довшину огорожі.

Щоб побудувати математичну модель задачі введемо позначення:

r – радіус кола (дано),

L – довжина кола (треба обчислити),

S – площа круга (треба обчислити).

Модель і метод розв'язування задачі – дві відомі формули:

$$L = 2\pi r$$

$$S = \pi r^2$$

Алгоритм розв'язування задачі такий:

<Ввести r >

$$L = 2\pi r$$

$$S = \pi r^2$$

<Вивести L, S >

Код мовою Visual Basic має такий вигляд:

```
Dim r As Single, L As Single, S As Single
```

```
Const pi = 3.1415
```

```
<Ввести r>
```

```
L = 2 * pi * r
```

```
S = pi * r ^ 2
```

```
<Вивести L, S>
```

Необов'язковою командою Dim оголошуємо тип змінних r, L, S як дійсний числовий. Командою Const оголошуємо константу π і запам'ятовуємо, що грецьких букв в кодах використовувати не можна. У кодах у числах замість десяткової коми треба писати крапку. Множення позначають зірочкою, а піднесення до степеня – символом «дашок».

Команди у кутових дужках несправжні. Замість них треба записати конкретні команди. Тут можливі різні конкретні команди, оскільки є різні способи введення даних і виведення результатів. Розглянемо деякі способи.

Ввести значення радіуса можна за допомогою команди присвоєння одним із трьох способів:

- 1) безпосередньо в коді, наприклад, так: $r = 8.5$;
- 2) в діалоговому режимі під час виконання коду за допомогою функції InputBox, наприклад, так: $r = \text{InputBox}(\text{"А тепер введіть значення радіуса"})$;
- 3) у візуальному програмуванні за допомогою елемента керування класу «текстове поле» на формі, наприклад, так: $r = \text{TextBox1}.\text{Text}$.

Вивести на екран результати обчислень L та S можна трьома способами:

- 1) у консольне вікно негайного виконання Immediate командою Debug.Print так: `Debug.Print "Довжина=" L, "Площа =" S ;`
- 2) у вікно повідомлень такою командою:
`MsgBox "Довжина=" & L & " Площа =" & S ;`
- 3) у візуальному програмуванні на форму двома способами: у текстові поля, наприклад, так:

```

Textbox2.Text = L
TextBox3.Text = S,

```

або в елементи керування класу «написи», наприклад, так:

```

Label1.Caption = L
Label2.Caption = S .

```

Продемонструємо усі описані способи.

Відкриємо середовище VBA комбінацією клавіш Alt + F11. Виконаємо команди з головного меню Insert \Rightarrow Module. Цим самим ми відкриємо вікно модуля Module1 для написання коду. Введемо текст sub klumba і натиснемо Enter. Отримаємо заготовку процедури

```

Sub klumba()
<Сюди вводять код>
End Sub
У цю заготовку введемо текст коду
Dim r As Single, L As Single, S As Single
Const pi = 3.1415
r = 8.5
L = 2 * pi * r
S = pi * r ^ 2
Debug.Print "Довжина=" L, "Площа =" S
MsgBox "Довжина=" & L & " Площа =" & S

```

Тут значення радіуса (нехай 8,5) надаємо командою присвоєння. Результати виводитимемо у консольне вікно Immediate і одночасно у вікно повідомлень.

Запустимо створений код на виконання. Курсор має знаходитися в тексті коду klumba. Натиснемо на зеленому трикутнику на панелі інструментів, що позначає команду Run, і отримуємо результати як у вікні Immediate, так і у вікні повідомлень.

Проведемо експеримент. Закриємо вікно повідомлень, натиснувши на кнопку ОК, і змінимо значення радіуса в кодї на 10,5. Виконаємо код повторно. Отримаємо нові результати.

Висновок. Щоб отримати результати для іншого значення радіуса, треба внести зміни в код і повторно його виконати.

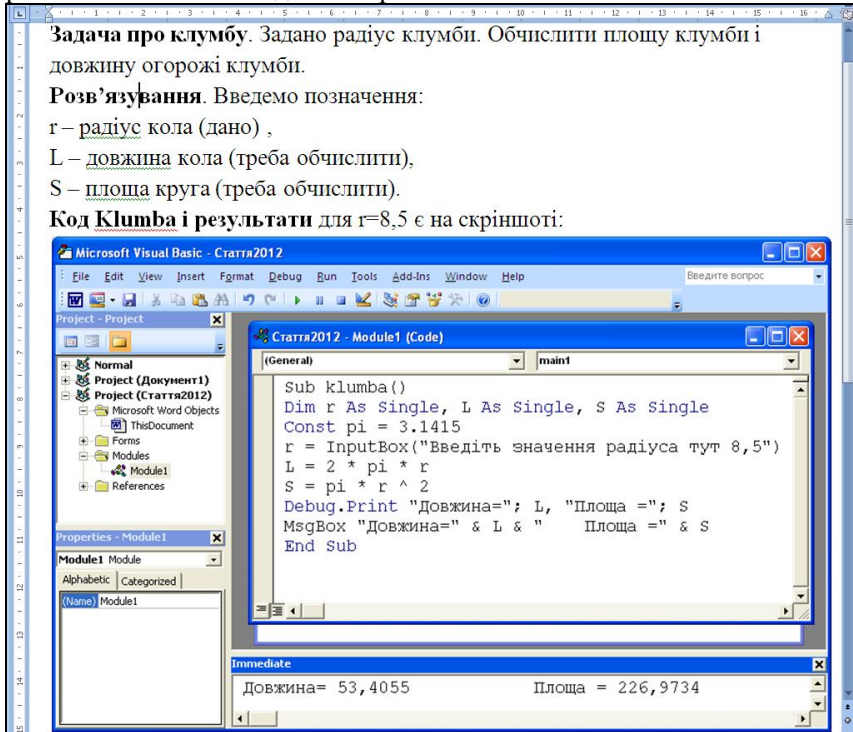


Рис. 1. Процедурне програмування у VBA

Виконаємо завдання на модифікацію коду: потрібно вводити значення радіуса у режимі діалогу.

Повертаємося в середовище VBA і в кодї замість команди $r = 10.5$ пишемо команду $r = \text{InputBox}(\text{"Введіть значення радіуса"})$. Запустимо код на виконання. Отримаємо діалогове вікно, куди введемо потрібне число, наприклад 8,5, чи інше. *Зверніть увагу!* У діалогових вікнах у десяткових числах зазвичай використовується кома, а не крапка. Натиснемо ОК і отримаємо результати.

Висновок. Використання режим діалогу дає змогу вводити вхідні дані на етапі виконання коду без внесення змін у код.

Підготуємо звіт про проведену роботу. Для цього натисканням клавіш Alt + PrintScreen скопіюємо зображення активного вікна, що містить

текст коду і результати роботи коду в буфер обміну, і вставимо це зображення комбінацією клавіш Ctrl + V у документ1, що містить умову задачі. Звіт матиме вигляд як на рис. 1. Підготовка звіту демонструє органічне поєднання програмування з роботою з текстовим документом, чого не можна досягнути під час використання професійних середовищ програмування на кшталт Visual Basic 2010 тощо. Ми продемонстрували використання VBA у програмі MS Word для реалізації концепції процедурного програмування.

Тепер розв'яжемо задачу про клумбу в стилі візуального програмування. Для цього використаємо об'єкт форму і такі візуальні елементи керування: три написи, три текстові поля, дві кнопки.

Запишемо в MS Word умову задачі, як вище. Увійдемо у середовище програмування VBA і вставимо в проєкт форму UserForm1 командами Insert ⇒ UserForm.

Підпишемо форму, задавши значення її властивості Caption як «Задача про клумбу». Для тла форми використаємо будь-яку картинку. За допомогою властивості Font задамо великий шрифт текстів для всіх елементів керування, що будуть на формі.

Розташуємо на формі (вертикально) три елементи керування класу «написи». Для цього клацнемо мишею на значку елемента Label в Toolbox і у формі. Вирівняємо елементи і підпишемо їх так: «Радіус», «Довжина», «Площа». Поряд з ними вставимо за допомогою Toolbox три текстові поля TextBox1, TextBox2, TextBox3 для відповідних числових величин задачі про клумбу. Вставимо дві кнопки і підпишемо їх за допомогою вікна властивостей як «Обчислити» і «Кінець роботи».

Сценарій проєкту такий: після запуску проєкту користувач має ввести в перше текстове поле значення радіуса, клацнути на кнопці «Обчислити» і у двох інших полях отримати результати. Після експериментів з різними значеннями радіуса і виконання всіх обчислень, треба закінчити роботу з проєктом, клацнувши на кнопці «Кінець роботи».

Створимо код кнопки «Обчислити». Для цього треба в конструкторі двічі клацнути на кнопці – відкриється вікно коду проєкту із заготовкою коду для кнопки CommandButton1_Click. У цю заготовку впишемо чи вставимо за допомогою буфера обміну код, що є у вікні коду на рис. 2:

```
Dim r As Single, L As Single, S As Single
Const pi = 3.1415
r = TextBox1.Text
L = 2 * pi * r
S = pi * r ^ 2
TextBox2.Text =L
TextBox3.Text = S
Debug.Print "Довжина=" L, "Площа ="S
```

Задача про клумбу. Задано радіус клумби. Обчислити її площу і довжину огорожі.

Розв'язування в стилі візуального програмування

UserForm1

Радіус	8,5
Довжина	53,4055
Площа	226,9734

Обчислити Кінець роботи

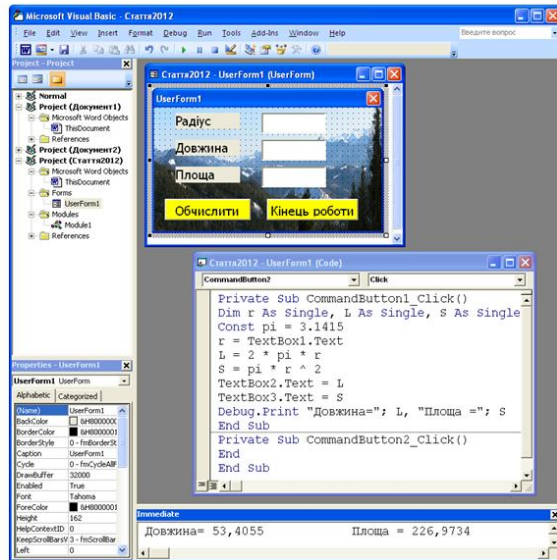


Рис. 2. Візуальне програмування у VBA

Значення радіуса вводимо з текстового поля, а результати виводимо у два текстові поля і дублюємо у вікні Immediate.

Створимо код кнопки «Кінець роботи», ввівши в заготовку коду кнопки одну команду End.

Запустимо проект на виконання. Отримаємо вікно працюючого проекту. Введемо у перше текстове поле значення радіуса 8,5 через десяткову кому, клацнемо на кнопці «Обчислити» і отримаємо результати. Тепер введемо нове значення радіуса 12,5. Натиснемо кнопку «Обчислити» і отримаємо інші результати.

Висновок. Ми маємо змогу обчислювати результати для різних вхідних значень без перезапуску проекту. Потрібні результати отримуються не в момент запуску проекту, а після настання події клацання на кнопці. Такий стиль програмування називають візуальним чи подійно-

орієнтованим.

Створимо звіт. Скопіюємо вікно проекту в буфер обміну (Alt + PrintScreen) і вставимо його в документ. Скопіюємо код кнопки «Обчислити» або ціле вікно з кодом проекту і вставимо його в документ (рис. 2).

Розв'яжемо задачу про клумбу в стилі об'єктного програмування. Запишемо в MS Word умову задачі і перейдемо в середовище VBA. Вставимо за допомогою головного меню в проект модуль класу Class1, який відразу перейменуємо на klumba. Складемо код класу, використовуючи поняття інкапсуляції, тобто створимо клас, що має властивість r (радіус) і два поля-методи: pl (для обчислення площі круга) і dov (для обчислення довжини кола). Вставимо модуль Module1 і запишемо головний код проекту, де командою Dim k As klumba спочатку оголосимо про наміри створити об'єкт k класу klumba, а командою Set k = New klumba реалізуємо намір створити об'єкт k. Доступ до полів об'єкта дають складені імена k.r, k.pl(), k.dov(). Виконаємо проект і отримаємо результати (рис. 3).

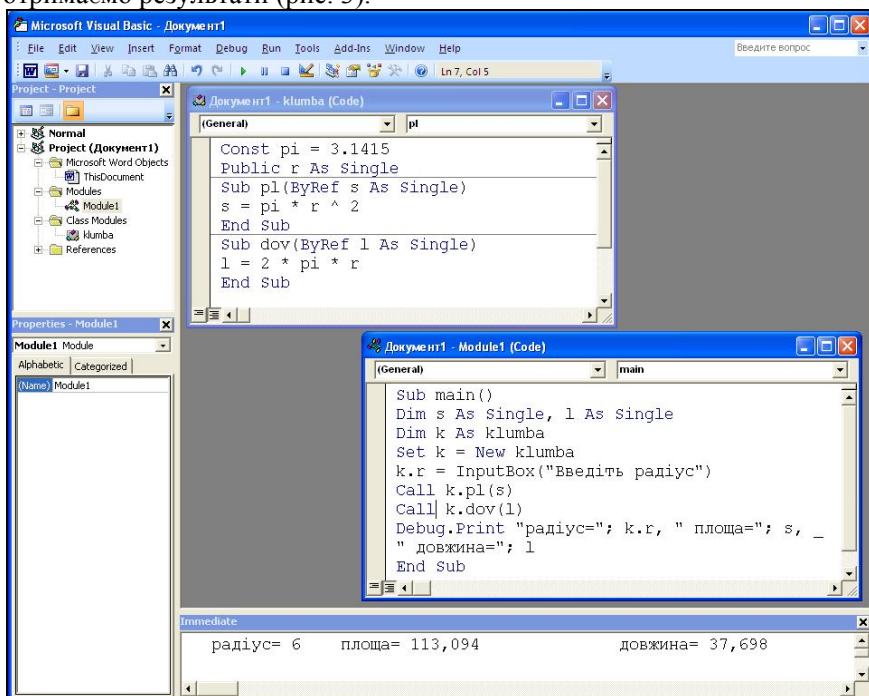


Рис. 3. Об'єктне програмування у VBA

Висновок. Ми продемонстрували спосіб розв'язування модельної

задачі з застосуванням елементів об'єктного програмування, формально створивши клас, що володіє методами, які є розв'язками задачі. Головний код не містить команд перетворення вхідних даних.

Зауважимо, що можливості повноцінного об'єктного програмування у VBA дещо обмежені, тому VBA не можна рекомендувати для професійного вивчення концепції об'єктно-орієнтованого програмування. Цих недоліків не мають сучасні версії мови Visual Basic 2010 та ін.

Розглянемо четвертий спосіб програмування – розв'яжемо задачу про клумбу безпосередньо в документі MS Word із застосування візуальних елементів (рис. 4).

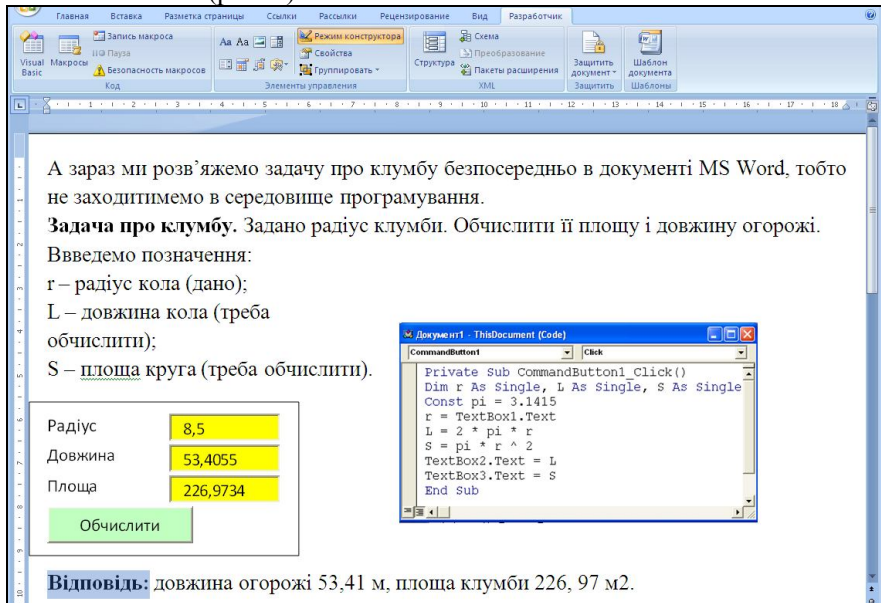


Рис. 4. Візуальне програмування в документі MS Word

На стрічку MS Word 2007 додамо закладку «Разработчик», відкриємо режим конструктора, панель елементів керування Visual Basic (тут елементи керування називаються елементами ActiveX) і вставимо елементи класів «написи», «текстові поля» і «кнопки» безпосередньо в документ під умовою задачі. Двічі клацнемо на кнопці «Обчислити» і у заготовку процедури введемо той же код, що розглядався раніше. Закриємо режим конструктора, введемо у перше текстове поле значення радіуса 8,5 і після клацання на кнопці «Обчислити» отримаємо результати у двох інших полях (див. рис. 4).

Розглянемо п'ятий стиль програмування, який можна назвати «негайним» за назвою вікна Immediate, в якому він реалізується.

Запишемо в MS Word умову задачі, перейдемо в середовище VBA і відкриємо консольне вікно Immediate командами View ⇒ Immediate Window. Введемо код розв’язування задачі і після введення кожної команди Print відразу ж отримаємо результати без запуску коду на виконання (рис. 5). Зауважимо, що таке програмування має обмежені можливості. Можна складати найпростіші лінійні, розгалужені і циклічні коди. Тут не можна застосовувати команди Dim, масиви, складені команди (наприклад «розгалуження» в «циклі») тощо. Програмування підтримується однопрохідним інтерпретатором, де виконання коду відбувається рядок за рядком відразу після введення рядка, тобто після натискання на клавішу Enter.

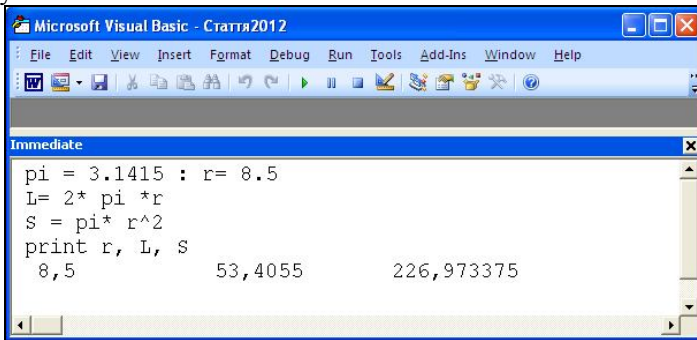


Рис. 5. Негайне програмування у VBA

Зауважимо, що наміри початківців розпочати роботу у VBA часто розбиваються після отримання повідомлення про неможливість роботи з макросами «macros disable», яке можна отримати під час першої спроби запустити проект на виконання. Ця проблема розв’язується так: у випадку роботи з давнішими версіями офісних програмах потрібно понизити рівень безпеки комп’ютера до середнього чи низького, а в нових версіях треба виконати такий алгоритм: послідовно слід натиснути головну офісну кнопку, кнопки «Параметри Word», «Центр керування безпекою», «Параметри центру керування безпекою», «Параметри макросів», увімкнути радіокнопку «Увімкнути всі макроси» і перемикач «Довіряти доступ до об’єктної моделі проектів VBA». Документ треба закрити зі збереженням і відкрити його повторно.

Погоджуючись з аргументами, що наведені в [5], і з важливістю навчання ООП засобами C#, ми і далі вважаємо, що шлях до ООП (як це видно з рис. 3) пролягає через процедурне програмування. Візуальне програмування – це лише засіб створення графічного інтерфейсу до програм користувача, що на стадії навчання учнів і студентів є вагомим мотивуючим фактором, оскільки розробка сценаріїв проектів і форми дає

змогу реалізувати їхні креативні задуми і є цікавим видом діяльності, що компенсує незацікавлення чи небажання займатися процедурним програмуванням. Для підтримки цієї позиції створено посібник [6], де достатньо повно описано принципи розв'язування задач від моделювання до програмування різними стилями у VBA і паралельно у Visual Basic 2010, а також розроблено серію навчальних відеофільмів для початківців.

Література

1. Куклев В. А. Становление системы мобильного обучения в открытом дистанционном образовании / Куклев В. А. // Международная научно-практическая заочная конференция «Интернет в образовании», 12 октября 2009 – 01 апреля 2010 г. ; Москва : Современная гуманитарная академия ; Московская финансово-промышленная академия. – Режим доступа к статье : http://www.conf.muh.ru/091012/thesis_Kuklev.htm
2. Поліщук О. П. Сучасні тенденції розвитку теорії і методики навчання інформатики / О. П. Поліщук, І. О. Теплицький, С. О. Семеріков // Теорія та методика навчання математики, фізики, інформатики : збірник наукових праць. – Вип. ІХ. – Кривий Ріг : Вид. відділ НметАУ, 2011. – С. 509-514.
3. Білоусова Л. І. Електронний навчальний посібник у системі засобів навчання інформатики в школі / Л. І. Білоусова, Н. В. Олефіренко // Теорія та методика навчання математики, фізики, інформатики : збірник наукових праць. – Вип. ІХ. – Кривий Ріг : Вид. відділ НметАУ, 2011. – С. 424-429.
4. Глинський Я. М. Бейсик повертається / Я. М. Глинський, В. А. Рязьська // Теорія та методика навчання математики, фізики, інформатики: збірник наукових праць. – Вип. ІХ. – Кривий Ріг : Вид. відділ НметАУ, 2011. – С. 443-449.
5. Шевчук П. Г. Навчання об'єктно-орієнтованому програмуванню в загальноосвітніх навчальних закладах засобами мови С# / П. Г. Шевчук // Теорія та методика навчання математики, фізики, інформатики: збірник наукових праць. – Вип. ІХ. – Кривий Ріг : Вид. відділ НметАУ, 2011. – С. 595-601.
6. Глинський Я. М. Інформатика. Основи алгоритмізації і програмування мовою Visual Basic : навч. посібн. / Глинський Я. М. – Львів : СПД Глинський, 2011. – 272 с.

ФУНКЦІЇ КУРСУ ОСНОВ ПРОГРАМУВАННЯ ДЛЯ МАЙБУТНІХ ПРОГРАМІСТІВ ТА ПРИНЦИПИ НАВЧАННЯ

Л. В. Гришко

Україна, м. Черкаси, Черкаський національний університет
імені Богдана Хмельницького
from4lu@mail.ru

А. М. Пишкало визначив методичну систему навчання як структуру, що складається з наступних основних елементів: цілей навчання, змісту навчання, методів, форм і засобів навчання [1]. Ефективність методичної системи навчання визначається ступенем досягнення поставлених перед нею цілей.

У роботі С. У. Гончаренка [2] дано означення методики навчального предмету: «Методика конкретного навчального предмета – це галузь педагогічної науки, що досліджує зміст навчального предмета й характер навчального процесу, який сприяє засвоєнню учнями необхідного рівня знань, умінь та навичок, розвитку мислення школярів, формуванню світогляду і виховання якостей громадянина своєї країни. До завдань методики входить дослідження змісту навчання, процесу викладання й процесу навчання». Також автор відзначає, що «методика глибоко пов'язана з відповідною наукою, оскільки в навчанні треба відображати особливості відповідної науки, її зміст і методи дослідження». На наш погляд, таке розуміння методики навчання предмету є прийнятним і для дисциплін, що вивчаються у ВНЗ, у тому числі й для курсу з основ програмування.

Цілі навчання основ програмування впливають із загальних цілей освіти, а також з особливостей програмування, як дисципліни, його ролі й місця в системі навчання програмістів. Головною метою навчання є різнобічний розвиток особистості, її здібностей, формування фахівця з комп'ютерних наук, конкурентоздатного на міжнародному ринку інтелектуальної праці, громадянина України, здатного працювати на користь своєї батьківщини.

Програмування – перша серед професійно-орієнтованих дисциплін, яку вивчають майбутні програмісти. Від змісту курсу, системи знань, умінь і навичок, що буде сформована у процесі навчання цього курсу, залежить якість засвоєння ними дисциплін, заснованих на знаннях із програмування, успішність у майбутній професійній діяльності.

Головним завданням при навчанні основ програмування майбутніх програмістів повинно бути навчання методів побудови ефективних алгоритмів, їх аналізу та реалізації сучасними засобами програмування на

комп'ютері.

Для досягнення поставленої мети в курсі з основ програмування передбачено ознайомлення студентів з історією розвитку мов програмування; парадигмами програмування; формування знань синтаксису та семантики мов програмування Pascal та C; об'єктів даних (змінні та константи, атрибути та зв'язування); елементарних типів даних (числові типи, символьний тип, булевий тип, покажчик), засобів їхнього представлення та алгоритмів обробки; структурованих типів даних (рядки, масиви, асоціативні масиви, записи, об'єднання, множини, файли, списки, АД), засобів їхнього представлення та алгоритмів обробки; контролю та перетворення типів; структур управління на рівні операторів (оператор присвоювання, складений оператор (блок), оператор розгалуження, оператор циклу, безумовний перехід); методів побудови алгоритмів; основ аналізу алгоритмів; основ технологій побудови програмних продуктів. Курс повинен забезпечити оволодіння студентами методологією структурного програмування. Із цією метою розглядаються засоби деконпозиції програм, методологія покрокової розробки програм та проектування «з гори до низу».

Після вивчення курсу студенти повинні:

- *знати* етапи обробки програм на ПЕОМ, такі як: редагування, трансляції, компонування; структуру програми мовою Pascal; структуру програми мовою C; методи представлення алгоритмів; синтаксис та семантику мов програмування Pascal та C; управляючі конструкції алгоритмів: послідовність, вибір, повторення; властивості алгоритмів; класифікацію операцій (арифметичні операції, операції відношення, логічні операції, порозрядні бітові та операції зсуву, операції над рядками, операції над множинами); пріоритет виконання операцій; структурну методологію розробки програм; класифікацію структур даних; скалярні типи даних; структуровані типи даних: масиви, рядки, записи, множини, файли, списки; оператори (прості, структуровані); методи побудови алгоритмів; бібліотечні підпрограми мови Pascal; графічні засоби мови Pascal та C; бібліотечні функції мови C; типи підпрограм мови Pascal, механізм передачі параметрів-значень та параметрів-змінних; механізми використання функцій у мові C; класи пам'яті мови C; форми рекурсивних алгоритмів; засоби препроцесорної обробки програми; модульний принцип розробки програм; методи оцінки швидкодії алгоритмів; методи аналізу алгоритмів; програмування модулів мови Pascal; парадигми програмування; технології проектування програмних комплексів;

- *уміти* формалізувати задачу; визначати структури даних для розв'язування даної задачі; будувати інформаційну модель задачі; розробляти алгоритми методом покрокового уточнення; будувати декілька різ-

них алгоритмів для розв'язування однієї задачі; подавати алгоритми в різних формах; складати лінійні програми, програми з розгалуженням, циклічні програми мовою Pascal та C; використовувати скалярні типи даних: цілі, дійсні, символьні, логічні; налагоджувати програму в відповідних середовищах розробки програмних продуктів (наприклад, Borland Pascal, Turbo Pascal, Free Pascal, Borland C, Turbo C, C Builder та ін.); користуватися довідковою системою середовища програмування; підключати процедури та функції стандартної бібліотеки модулів мови Pascal; підключати функції стандартної бібліотеки мови C; використовувати структуровані типи даних при складанні програм; ділити складну задачу на елементарні складові і задавати їхнє комбінування; використовувати власні процедури та функції при складанні програм мовою Pascal; використовувати власні функції при складанні програм мовою C; оцінювати швидкість алгоритмів; будувати алгоритми з використанням мінімального об'єму пам'яті; застосовувати і комбінувати відомі прийоми програмування і типові алгоритми; застосовувати методи побудови алгоритмів; реалізувати багатомодульні програми; складати і використовувати макрозасоби та засоби препроцесорної обробки програм у C;

- *здійснювати пошук відомостей*, необхідних для розв'язування задачі; будувати алгоритми і писати програми для розв'язування навчальних задач, які правильно і добре документовані; аналізувати власні помилки правильно і однозначно висловлювати думку в зрозумілій співбесіднику формі; розуміти готові програми, що написані іншими програмістами; розробляти невеликі програмні продукти індивідуально, мікрогрупами та колективно.

Треба зазначити неприпустимість зведення навчальних цілей тільки до засвоєння предметних знань і вмінь. Вважаємо за необхідне в системі цілей включати вміння студентів самостійно навчатися як у навчальному закладі, так і в процесі наступної професійної діяльності.

Виходячи із зазначеного, визначимо основні функції навчання основ програмування [3].

Загальноосвітня функція навчання основ програмування полягає в тому, щоб забезпечити в студентів:

- *формування знань* про: фундаментальні концепції і парадигми програмування; методологію побудови інформаційних моделей; базові принципи побудови алгоритмів і програм; модульний принцип розробки програм; типові алгоритмічні конструкції: слідування, розгалуження, повторення; синтаксис та семантику мов програмування високого рівня; етапи обробки програм на ПК, таких як: редагування, трансляція, компонування; типи і структури даних та алгоритми їх обробки; типи підпрограм, механізм передачі параметрів-значень та параметрів-змінних;

методи структурного низхідного програмування; рекурсивні алгоритми та способи їх побудови; методи динамічного розподілу пам'яті; критерії ефективності алгоритмів і програм; засоби для розробки, супроводу та організації багатофайлових прикладних програм;

- *формування вмінь і навичок* практичного програмування мовами програмування високого рівня Pascal та C, а саме: складати лінійні, розгалужені та циклічні програми з використанням стандартних та структурованих типів даних; розробляти алгоритми і програми методом покрокового уточнення; налагоджувати й тестувати навчальні програми в середовищі розробки мови програмування; використовувати підпрограми при складанні програм мовою Pascal та складати програми мовою C, які містять одну або декілька функцій; використовувати й адаптувати стандартні алгоритми; будувати алгоритми, ефективні за часом та пам'яттю; реалізовувати багатомодульні програми; виконувати роботу, що включає опис проблеми, її аналіз, проектування, розробку й модифікацію невеликих програмних проектів, а також підготовку необхідної документації.

Важливу роль відіграє курс з основ програмування у формуванні пізнавальних, організаційних, комунікативних умінь і навичок студентів, а також у формуванні і розвитку в них потреби постійно розширювати і поглиблювати свої знання. Тому, розвиваюча функція навчання основ програмування полягає в тому, щоб сприяти

- *розвитку*: алгоритмічного стилю мислення; інтелектуальних якостей і творчих здібностей, потрібних для розуміння програм; критичного та творчого мислення; інформаційної культури; професійної інтуїції; здатності до порівняльного / критичного аналізу;

- *формуванню здатності*: бачити задачу одночасно на різних рівнях деталізації; узагальнювати типові ситуації; до вивчення нових мов програмування; використовувати знання і навички в новій ситуації;

- *формуванню*: розуміння зв'язку теорії програмування із практикою програмування; навичок самостійної роботи з навчальною та науковою літературою при розв'язуванні завдань з програмування; розуміння галузі застосування здобутих знань в інших професійно-орієнтованих і спеціальних дисциплінах та в сфері майбутньої професійної діяльності студентів; наукового світогляду; готовності до навчання протягом усього життя.

Виховна функція навчання основ програмування полягає у формуванні в студентів потреби і уміння використовувати науковий зміст основ програмування для усвідомлення цілісної картини процесу розв'язування пізнавальних і професійних задач або проблем, а також, пов'язана з формуванням у студентів якостей, які притаманні фахівцям

із комп'ютерних наук та програмістам, у тому числі: педантичність, дисциплінованість, акуратність; наполегливість; внутрішня керованість; низька нетерпимість до невизначеності; усвідомлення особистої відповідальності за результати своєї роботи; прагнення до самоутвердження через творчу діяльність; уміння працювати як індивідуально, так і як лідер або член команди; бережливе відношення до техніки; системність і цілеспрямованість у професійній діяльності.

Виходячи з мети та функцій навчання основ програмування майбутніх програмістів визначимо принципи навчання. Принципи навчання – це керівні ідеї, нормативні вимоги до організації і здійснення освітнього процесу [4]. Ми вважаємо, що для успішної структуризації процесу навчання слід керуватися загальнодидактичними принципами навчання.

У вищій школі принципи навчання розділяють на дві групи [5]. До першої групи відносять принципи, які відображають закономірності зв'язку мети і змісту навчання: розвиваючого і виховного характеру навчання; науковості навчання; активності і самостійності студентів у навчанні; систематичності навчання; зв'язку змісту навчання з життям, практикою. До другої групи відносять принципи моделювання умов продуктивного навчання: принципи наочності в навчанні; врахування вікових та індивідуальних особливостей студентів; міцності знань; об'єднання колективних, групових і індивідуальних форм організації навчання; зв'язку лекційної і поза лекційної навчально-виховної роботи зі студентами; керівної ролі викладача, кафедри та ін.

У процесі організації навчальної діяльності студентів при навчанні основ програмування нами визначено наступні *принципи навчання*:

Принцип спрямованості процесу навчання на гармонійний розвиток особистості. Згідно з цим принципом, процес навчання основ програмування покликаний формувати у студента різносторонні знання, уміння і навички, які відповідають рівню розвитку сучасних комп'ютерних наук і комп'ютерної техніки; розвивати волюву, інтелектуальну, емоційну сфери особистості; формувати суспільно значущі мотиви діяльності та інтереси, які істотним чином впливають на майбутню професійну діяльність і широкий розумовий кругозір.

Досвід роботи показує, що навчання основ програмування сприяє формуванню і розвитку пізнавальних інтересів студентів у тому випадку, якщо існує взаємозв'язок між теоретичним матеріалом і практичними завданнями; лекційні заняття чергуються із лабораторними заняттями; здійснюється підбір завдань з урахуванням індивідуальних здібностей студентів.

Принцип науковості навчання. Слід зазначити, що комп'ютерні науки і сьогодні знаходяться в стадії розвитку і що в цій галузі, як ні в якій

інший, студенти зустрічаються з новими науковими і технічними термінами, з новими поняттями.

Власний досвід підтверджує, що для реалізації принципу науковості навчання основ програмування доцільно використовувати метод наукового пізнання; знайомити студентів з історією розвитку обчислювальної техніки і комп'ютерних наук; надавати можливість студентам переживати радість власного відкриття при розв'язуванні задач; привчати студентів читати періодичну літературу, науково-популярні журнали і робити повідомлення про прочитане; сприяти участі студентів в олімпіадах з програмування і наукових студентських конференціях.

Разом із принципом науковості при навчанні основ програмування слід дотримуватися принципу доступності, суть якого полягає в тому, що зміст, форми і методи навчання повинні враховувати реальні можливості студентів.

Принцип створення відповідних умов для функціонування процесу навчання. Цей принцип є особливо важливим при навчанні комп'ютерним наукам і, у тому числі, основ програмування, оскільки комп'ютерна техніка, що використовується в процесі навчання, повинна відповідати санітарним нормам. Крім того, у процесі навчання слід дотримуватися норми температурного режиму, освітленості, провітрювання в навчальних приміщеннях і в комп'ютерних класах.

Згідно з цим принципом, на викладача лягає відповідальність забезпечення сприятливої морально-психологічної атмосфери під час проведення занять, передбачених навчальним планом дисципліни, зокрема, і в студентському колективі в цілому.

Принцип систематичності і послідовності навчання. За цим принципом передбачається формування знань, умінь і навичок [6] «в певному порядку, в системі: кожний елемент навчального матеріалу логічно пов'язується з іншими, наступне спирається на попереднє і готує до засвоєння нового».

Для навчання інформатики А. П. Єршов запропонував реалізацію принципу послідовності у формі циклічності. Реалізація цього принципу передбачає розподілення навчального матеріалу на окремі модулі, які, у свою чергу, повинні складатися з окремих тем, логічно пов'язаних між собою; визначення теми на лекційних заняттях, побудову логічно обгрунтованої опорної схеми подання змісту навчального матеріалу; формулювання мети заняття на лабораторних роботах, демонстрацію можливості і способів досягнення мети заняття на конкретних прикладах і задачах.

Принцип свідомості, активності і самостійності студентів у навчанні при керівній ролі викладача. Цей принцип спрямований на фор-

мування особистості, яка матиме соціально значущу життєву позицією. Свідоме здобування знань студентами залежить від таких чинників як організація навчально-виховного процесу; від методів і засобів навчання; від рівня пізнавальної активності студентів, від мотивів навчання. Для реалізації принципу свідомості при навчанні основ програмування викладачу слід пояснювати значення і важливість навчального матеріалу кожного заняття; забезпечувати розуміння студентом кожного поняття, що вводиться в процесі навчання; навчати так, щоб студент розумів, які і у якій послідовності треба виконувати дії для розв'язування поставленої перед ним задачі, запобігаючи механічному, неусвідомленому виконанню завдань; навчати студентів знаходити декілька різних алгоритмів розв'язування задачі; використовувати можливості взаємонавчання студентів; забезпечувати розвиток інтелектуальних здібностей студентів через їх усвідомлене ставлення до здобування знань.

Принцип активності підкреслює функцію студента як суб'єкта навчання, а не пасивного учасника процесу. У методиці навчання інформатики розрізняють інтелектуальну та емоційну активність [7]. Інтелектуальна активність у навчанні основ програмування забезпечується постановкою проблемних задач.

Емоційна активність студента передбачає його зацікавленість предметом. Для досягнення успіху процесу навчання необхідно сформувати позитивне емоційне відношення студента до предмету, а саме, потрібно чітко позначити студентам перспективу їхньої професійної діяльності, забезпечити їх необхідними для навчання методичними матеріалами, привчати студентів до підтримки в порядку навчальних приміщень (лекційні аудиторії, комп'ютерні класи). Не менш важливим є формування в студентів позитивного емоційного відношення до викладача, що забезпечується уважним, поважним, доброзичливим, строгим і справедливим відношенням викладача до студентів.

Для реалізації принципу активності викладачу слід навчати студентів ставити питання; контролювати чинники, які відволікають увагу студентів від об'єкту навчання, і усувати причини їх виникнення; вивчати і розвивати індивідуальні інтереси студентів до програмування; підтримувати ініціативу студентів, яка спрямована на досягнення навчальних цілей, а саме, уважно відноситься до гіпотез, версій, варіантів розв'язування поставленої перед студентами задачі.

Принцип самостійності виражається в умінні студента управляти своєю навчальною діяльністю. У зв'язку із цим важливі дії викладача, спрямовані на формування у студентів умінь і навичок самостійної навчально-пізнавальної діяльності, які полягають в умінні знаходити потрібні відомості в навчальній, науковій літературі, у мережі Internet і

використовувати їх для розв'язування завдань; результативно працювати в комп'ютерному класі в час, який відведено для самостійної роботи.

Принцип індивідуалізації і колективності навчання. Принцип індивідуалізації впливає із необхідності врахування індивідуально-психологічних особливостей кожного студента. Викладач повинен ураховувати індивідуальні особливості студентів, їхні здібності до навчання при виборі темпу навчання, підготовці завдань для самостійного виконання. Це забезпечується спостережливістю, чуйним й поважним відношенням викладача до кожного студента. Викладач повинен знати кого зі студентів варто похвалити за найменший успіх; кому потрібно вчасно зробити зауваження; коли зробити вигляд, що не помітив помилки студента; кому дати індивідуальне навчальне завдання.

Принцип колективності враховує необхідність колективної роботи студентів, з одного боку, для зміцнення колективістських якостей студентів і для посилення виховної дії навчання, з іншого боку, для формування професійних якостей спеціаліста в галузі комп'ютерних наук.

Принцип ефективного поєднання методів навчання. В основі цього принципу знаходиться теза про те, що «процес засвоєння знань, умінь і навичок відбуватиметься найбільш ефективно, якщо в міру можливості в ньому братимуть участь всі органи чуття людини – зір, слух, дотик та ін.» [6]. Набуття студентами практичних навичок розв'язування задач за допомогою комп'ютера в курсі з основ програмування є реалізацією принципу поєднання практичних і наочних методів навчання, що забезпечує досягнення освітніх цілей у підготовці майбутніх інженерів-програмістів.

Принцип міцності і дієвості знань, умінь і навичок. Згідно з цим принципом, викладач повинен, з одного боку, обмірковувати зміст предмету з погляду виділення в ньому найважливіших, істотних знань, які повинні бути обов'язково осмислені і засвоєні студентами і, з іншого боку, сформувані у студентів уміння і навички використання цих знань для розв'язування практичних навчальних задач програмування. Для реалізації цього принципу викладачу слід сприяти розвитку мислення; вчити і привчати студентів користуватися довідниками з інформаційних технологій, мов програмування і т.п.; формувати позитивне відношення до науки програмування; стежити за логікою подання навчального матеріалу; застосовувати технічні засоби, навчальні дискусії, яскраве емоційне подання навчального матеріалу; обмірковувати структуру завдань до кожного заняття; за можливістю підбирати завдання, що мають практичне застосування; при повторенні навчального матеріалу активізувати у студентів не тільки пам'ять, але й мислення.

Принцип забезпечення оперативного контролю і самоконтролю в

навчанні. Контроль рівня знань, умінь і навичок студентів в процесі навчання основ програмування, необхідний для забезпечення зворотного зв'язку в процесі навчання; отримання відомостей про ступінь ефективності процесу навчання; для здійснення функцій освіти, навчання, діагностики, стимулювання, виховання, розвитку; формування у студентів навичок самоконтролю; можливого коректування процесом навчання під впливом одержаних відомостей про хід засвоєння навчального матеріалу.

Для реалізації цього принципу слід використовувати різноманітні форми контролю знань, умінь і навичок студентів: усне опитування, тестування, тематичні диктанти, самостійні роботи, письмові контрольні роботи, перевірка виконання домашніх завдань, модульний контроль, екзамен (іспит).

У ході контролю викладачеві слід забезпечувати сприятливу психологічну атмосферу, виключити можливості виникнення у студентів стресових станів, які згубно позначаються на їх емоційній сфері і можуть викликати гальмування розумової діяльності.

Досвід показує, що реалізація функцій навчання і дотримання означених принципів навчання, надають можливість забезпечити досягнення головних цілей навчання основ програмування майбутніх програмістів.

Література

1. Пышкало А. М. Методическая система обучения геометрии в начальной школе : автореф. дис. ... док. пед. наук. – М., 1975. – 60 с.
2. Гончаренко С. У. Методика як наука / Гончаренко С. У. – Хмельницький : Вид-во ХГПК, 2000. – 30 с.
3. Триус Ю. В. Особливості створення методичної системи навчання основ програмування для підготовки майбутніх інженерів-програмістів / Триус Ю. В., Богатирьов О. О., Гришко Л. В. // Вісник Черкаського університету, серія «Педагогічні науки». Випуск 35. – Черкаси, 2002. – С. 135-141.
4. Морзе Н. В. Методика навчання інформатики : навч. посіб. : у 4 ч. / За ред. акад. М. І. Жалдака. – К. : Навчальна Книга, 2003. – Ч. 1: Загальна методика навчання інформатики. – 254 с.
5. Алексюк А. М. Педагогіка вищої освіти України / А. М. Алексюк. – К. : Либідь, 1998. – 560 с. – (Історія. Теорія: Підручник).
6. Бабанский Ю. К. Оптимизация процесса обучения / Бабанский Ю. К. – М. : Педагогика, 1977. – 256 с.
7. Могилев А. В. Информатика : учеб. пособие для пед. вузов / А. В. Могилев, Н. И. Пак, Е. К. Хённер ; под ред. Е. К. Хённера. – М., 1999. – 816 с.

ОФІСНИЙ ПАКЕТ В КУРСІ ІНФОРМАТИКИ ВИЩОГО НАВЧАЛЬНОГО ЗАКЛАДУ

Г. Г. Злобін

Україна, м. Львів, Львівський національний університет
імені Івана Франка

zlobin@electronics.wups.lviv.ua

Сьогодні неможливо уявити випускника ВНЗ, який не вміє працювати з текстовими документами, електронними таблицями та презентаціями на ПЕОМ. Кількість офісних пакетів, якими можна скористатись з цією метою, немала: Microsoft Office; Corel WordPerfect Office; IBM Lotus SmartSuite; IBM Lotus Symphony; iWork; OpenOffice.org; LibreOffice; ThinkFree Office; Лексикон 5.1.

До офісних пакетів, які встановлюються на локальні робочі станції, слід додати онлайнові офісні пакети: Zoho Office Suite; Google Docs; Microsoft Office 365; Oracle Cloud Office; ThinkFree Office Online; OpenGoo.

В зв'язку з цим важко передбачити, з яким офісним пакетом зустрінеється випускник ВНЗ на своєму робочому місці.

Разом з тим, після перегляду навчальних програм і підручників з інформатики для студентів ВНЗ, може скластись враження, що у світі є лише одна операційна система Microsoft Windows і єдиний офісний пакет Microsoft Office [1–3]. Почасти це зумовлено політикою видавництва (українських і російських), які масово друкують підручники і навчальні посібники з описом лише пропрієтарного програмного забезпечення, почасти політикою Міністерства освіти і науки, молоді та спорту України, яке далі розмов про впровадження вільного програмного забезпечення в навчальний процес не йде. Свою частку в замовчування факту існування конкурентоспроможного вільного програмного забезпечення вносять викладачі ВНЗ, які не бажають освоювати нові програмні засоби і не переймаються ліцензійністю програм, використовуваних в навчальному процесі.

В умовах існування великої кількості альтернативних офісних пакетів орієнтація більшості викладачів ВНЗ на Microsoft Office знижує рівень професійної компетентності випускників. Для забезпечення готовності випускників ВНЗ до використання будь-якого сучасного офісного пакету потрібно перейти від вивчення інтерфейсу Microsoft Office (тим більше, що Microsoft, починаючи з Microsoft Office 2007, докорінно змінив попередній інтерфейс) до формування алгоритмів виконання основних операцій з офісним пакетом:

- створити новий документ: Файл (File) – Новий (New) або відповідний інструмент в панелі інструментів;
- відкрити існуючий документ: Файл (File) – Відкрити (Open) або відповідний інструмент в панелі інструментів;
- зберегти створений документ: Файл (File) – Зберегти/Зберегти як (Save) або відповідний інструмент в панелі інструментів;
- вставити в текст малюнок: Вставка (Insert) – Малюнок (Picture) – З файлу (From File...);
- вставити в текст таблицю: Таблиця (Table) – дії з таблицею і її елементами;
- і т.д.

Для формування таких алгоритмів достатньо розглянути на лекційних заняттях повністю основний офісний пакет, за аналогією – один стаціонарний і один мережевий офісний пакет. На лабораторних заняттях в основному офісному пакеті виконуються усі завдання, передбачені програмою курсу, в додаткових офісних пакетах – відкриття і редагування документів, створених в основному офісному пакеті. Такий підхід автор використав у Львівському банківському інституті (2003-2006 рр.) та Львівському державному університеті внутрішніх справ (2008-2009 рр.): основний офісний пакет – Microsoft Office XP, додатковий офісний пакет – OpenOffice.org.ukr.

На думку автора, найкращим претендентом на роль основного офісного пакету є OpenOffice/LibreOffice, оскільки вони достатньо добре локалізовані та вільно поширювані; як додатковий офісний пакет можна використовувати Microsoft Office 365 (якщо він буде доступним для ВНЗ безоплатно). Найкращим претендентом на роль основного мережевого офісного пакету є Google Docs.

Література

1. Інформатика та комп'ютерна техніка : навч.-метод. посібник для самост. вивч. дисц. / Рзаєв Д. О., Шарапов О. Д., Ігнатенко В. М., Дибкова Л. М. – К. : КНЕУ, 2002. – 486 с.
2. Доценко В. І. Медична інформатика : навч. посібник / Доценко В. І., Сілкова О. В. – Полтава, 2005. – 164 с.
3. Інформатика. Комп'ютерна техніка. Комп'ютерні технології : підручник / за ред. Баженова В. А., Шинкаренко Г. А. – К. : Каравела, 2011. – 592 с.

ОБУЧЕНИЕ ЯЗЫКУ JAVA КАК ОДНОМУ ИЗ ФУНДАМЕНТАЛЬНЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ ДЛЯ ОБУЧЕНИЯ ИНЖЕНЕРОВ-ПРОГРАММИСТОВ

Ф. С. Ильясова

Украина, г. Симферополь, Крымский инженерно-педагогический
университет

fatime.ilyasova@gmail.com

Для студентов направления подготовки «Информатика» важное место занимают компьютерные дисциплины, одной из которых является дисциплина «Язык программирования Java». Как известно, сегодня язык Java является стандартом описания программных систем [1]. Программа курса состоит из 12 лекций и 15 лабораторных работ. Лабораторные работы содержат задания и примеры, демонстрирующие базовые возможности языка Java. Лекционные занятия содержат материалы из различных источников и мультимедийное сопровождение, которые знакомят с возможностями объектно-ориентированного языка программирования Java.

Овладение содержанием дисциплины «Язык программирование Java» предусматривает разнообразные формы и методы практической подготовки студентов к будущей профессиональной деятельности [2].

Программа курса рассчитана на один семестр и обеспечивает адаптацию студентов к специфическим условиям обучения в вузе, формирование личности будущего специалиста в области компьютерных технологий. Язык Java является передовым языком в сфере разработок новейших компьютерных технологий, поэтому изучение данной дисциплины является необходимым для повышения профессионального уровня подготовки студентов специальности информатика [3], [4].

Целью данной статьи является предложение одной из методик обучения языку Java.

При изучении учебной дисциплины «Язык программирования Java» студент должен **знать**: 1) основы объектно-ориентированного программирования; 2) лексику языка; 3) типы данных; 4) понятия об именах и пакетах; 5) методы объявления классов; 6) преобразования типов; 7) объектную модель Java.

Студент должен **уметь** реализовывать простейшие задачи: 1) с применением условных и циклических операторов; 2) с применением методов ввода/вывода в интерактивном режиме; 3) создания классов.

Дисциплина «Язык программирования Java» связана с дисциплинами образовательно-квалификационного уровня «бакалавр» направления

подготовки «Информатика»: «Программирование для начинающих» (1 семестр), «Программирование» (1, 2, 3 семестры), «Программирование на языке Python» (4 семестр), «Объектно-ориентированное проектирование» (6 семестр), «Основы математического моделирования и системный анализ» (7 семестр).

Из таблицы 1 видно, что дисциплина разделена на 2 модуля, которые включают шесть содержательных модулей.

Таблица 1

Структура содержания программы

<i>Модуль 1. Синтаксис языка Java (Структурное программирование)</i>	<i>Модуль №2. Классы (Объектно-ориентированное программирование)</i>
Содержательный модуль 1.	Содержательный модуль 4
ТЕМА 1.1. Идентификаторы и зарезервированные слова	ТЕМА 2.1. Члены класса
ТЕМА 1.2. Выражения и операторы	ТЕМА 2.2 Создание и инициализация объекта
ТЕМА 1.3. Прimitives типы данных	ТЕМА 2.3. Подклассы и наследование
Содержательный модуль 2	Содержательный модуль 5
ТЕМА 1.4. Методы	ТЕМА 2.4. Скрытие и инкапсуляция данных
ТЕМА 1.5. Ссылочные типы	ТЕМА 2.5. Абстрактные классы и методы
Содержательный модуль 3	ТЕМА 2.6. Интерфейсы
ТЕМА 1.6. Массивы	Содержательный модуль 6
ТЕМА 1.7. Структура файла Java	ТЕМА 2.7. Пакет java.awt
ТЕМА 1.8. Пакеты и пространства имен	ТЕМА 2.8. Пакет java.util

Ниже приведен пример простейшей лабораторной работы для студентов направления подготовки «Информатика»:

План лабораторной работы №7

По дисциплине: Язык программирования Java

Тема занятия: **Вложенные классы**

Цель занятия: Научить студентов осуществлять вызов одного класса в другом.

Количество часов: 2

Содержание работы:

Создадим класс «third_class»

```
public class third_class {
    public void simpleMessage() {
```

```

        System.out.println("This is another class");
    }
}

```

Обратите внимание на метод этого класса. Когда он откомпилируется, в консольном режиме его невозможно будет запустить, т.к. отсутствует метод «main».

Для того, чтобы увидеть результат выполнения класса «third_class», создадим еще один класс, использующий метод «simpleMessage». Назовем его «Main_class»

```

public class Main_class {
    public static void main(String[ ] args) {
        third_class third_classObject=new third_class();
        third_classObject.simpleMessage();
    }
}

```

Класс создает объект класса «Main_class», который наследует метод «simpleMessage». Откомпилируйте и запустите программу.

Пример интерактивного ввода с клавиатуры с использованием «вложенных классов»

Изменим уже созданный класс «third_class» на следующее:

```

public class third_class {
    public void simpleMessage(String name) {
        System.out.println("Hello"+name);
    }
}

```

Обратите внимание на параметры метода «simpleMessage».

Изменим уже созданный класс «Main_class» на следующее:

```

import java.util.Scanner;
public class Main_class {
    public static void main(String[ ] args) {
        Scanner input = new Scanner(System.in);
        third_class third_classObject=new third_class();
        System.out.println("Enter you name here:");
        String name=input.nextLine();
        third_classObject.simpleMessage (name);
    }
}

```

Задание для самостоятельного выполнения:

1. Измените классы так, чтобы при вводе числа с клавиатуры выводилось только то, которое <=25.

2. Измените классы так, чтобы можно было ввести любые 10 чисел с плавающей точкой, а вывод осуществлялся только тех, которые <=25.

Примечание: для выполнения задания воспользуйтесь строкой для приведения типов

```
double x=Double.parseDouble (name) ;
```

Объявление массива

```
double [] A=new double [10];
```

Выводы: 1) овладение содержанием дисциплины «Язык программирование Java» предусматривает разнообразные формы и методы практической подготовки студентов в будущей профессиональной деятельности; 2) полученные знания, умения, навыки студенты специальности «Информатика» могут использовать для работы инженерами-программистами.

Литература

1. Флэнаган Д. Java в примерах : справочник, 2-е издание / Дэвид Флэнаган ; пер. с англ. – СПб. : Символ-Плюс, 2003. – 664 с.

2. Флэнаган Д. Java : справочник, 4-е издание / Дэвид Флэнаган ; пер. с англ. – СПб. : Символ-Плюс, 2004. – 1040 с.

3. Электронная библиотека фирмы IBM [Электронный ресурс]. – Режим доступа : ibm.com/developerworks/ru/library/j-javafx/index.html

4. TIOBE Programming Community Index [Electronic resource]. – Mode of access : tiobe.com/index.php/content/paperinfo/tpci/index.html

МОДЕЛЬ ФУНДАМЕНТАЛІЗАЦІЇ НАВЧАННЯ ІНФОРМАТИЧНИХ ДИСЦИПЛІН МАЙБУТНІХ БАКАЛАВРІВ ІНФОРМАТИКИ НА ОСНОВІ МІЖПРЕДМЕТНИХ ЗВ'ЯЗКІВ

У. П. Когут

Україна, м. Київ, Інститут інформаційних технологій і засобів навчання
НАПН України
ulyana_kogut@mail.ru

Актуальність. В умовах ринкової економіки відбувається посилення конкуренції серед випускників ВНЗ, що зумовлює високі вимоги до якості підготовки фахівців. Досягнення високого ступеня професіоналізму майбутніх бакалаврів інформатики можливе лише за умови відповідної фундаментальної освіти, тому для якісної підготовки фахівців необхідно посилення її математичної складової, оскільки математика є основою фахової підготовки: вона сприяє розвитку логічного мислення, просторової уяви, алгоритмічної культури, вміння встановлювати причинно-наслідкові зв'язки, обґрунтовувати твердження, моделювати процеси та явища [6].

Постановка проблеми. Характерною рисою сучасної наукової картини світу є визнання фундаментальної ролі інформаційного фактору, інформаційних процесів в природі та суспільстві. У зв'язку з цим зростає значення глибоких та комплексних фундаментальних знань з інформатичних дисциплін, що мають отримати студенти в процесі освіти.

Метою дослідження є аналіз моделі фундаменталізації навчання інформатичних дисциплін на основі міжпредметних зв'язків, виявлення тенденцій використання СКМ для викладання інформатичних дисциплін.

Ступінь розробки проблеми. Застосування засобів СКМ у навчанні для фундаментальної підготовки майбутніх бакалаврів інформатики розглядали М. І. Жалдак, Т. П. Кобильник, Н. В. Морзе, Ю. С. Рамський, С. О. Семеріков.

Виклад основного матеріалу. Одним із напрямів у фундаменталізації навчання є впровадження компетентнісного підходу, який спрямований на врахування індивідуальних особливостей студентів, а також максимальне використання всього арсеналу профорієнтаційних можливостей навчально-педагогічного процесу, створення та впровадження педагогічних та інформаційних технологій, орієнтація не тільки на підвищення рівня знань, але й на розвиток професійного самовизначення.

Для цього необхідно побудувати модель фундаменталізації навчання інформатичних дисциплін майбутніх бакалаврів інформатики, яка

характеризується наступними **критеріями**:

- особистісна орієнтація змісту навчання;
- затребуваність результатів навчання в житті;
- діяльнісний характер навчання;
- забезпечення варіативності та свободи вибору в навчанні;
- цілісність змісту навчання, забезпечення міжпредметних зв'язків;
- орієнтація на компетентнісний підхід.

Модель навчання майбутніх бакалаврів інформатики ефективно буде функціонувати, якщо базуватиметься на певних принципах [4]:

- інтегрованості;
- диференційованості;
- міжпредметності;
- індивідуальності освітньої траєкторії студента;
- неперервності теоретичного і практичного навчання;
- гуманізації та демократизації та ін;

Компетентнісний підхід у навчанні на противагу концепції засвоєння знань передбачає опанування студентами вмінь, які допоможуть їм в майбутньому діяти ефективно в різних життєвих ситуаціях, а особливо в таких критичних ситуаціях, для яких неможливо наперед розробити стратегію поведінки. Фактично при такому підході трактуванню *знання* як накопиченню суми предметної інформації та відомостей протиставляється *знання* як комплекс вмінь, здатностей щодо використання відомостей, які дозволяють діяти і отримувати необхідний результат в невідзначених ситуаціях.

Методика формування фахових компетентностей в умовах діяльності навчального закладу включає наступну сукупність організаційно-змістових **компонентів**:

- форми (лекція, практичне заняття, самостійна та лабораторна робота, наукові конференції);
- методи (використання інформаційних технологій, математичне моделювання, тестування, захист проєктів і розробок);
- засоби (навчальні книги (посібники, методичні рекомендації, збірники задач, опорні конспекти лекцій));
- принципи (науковості, систематичності, доступності, свідомості, творчої активності, фахової спрямованості навчання, між предметності, імітаційне моделювання).

Переважно фахові компетентності розглядаються в двох аспектах: 1) як ціль освіти; 2) як проміжний результат, який характеризує спеціаліста, що має здійснювати професійну діяльність.

Ступінь сформованості компетентностей спеціаліста оцінюється наступними **критеріями**:

- сформованість системи професійних знань (когнітивний компонент);
- сформованість системи професійних вмінь, навиків та досвіду (діяльнісний компонент);
- ступінь вираження професійної мотивації (мотиваційний компонент);
- організаційні і комунікативні якості (організаційно-комунікативний компонент);
- якості мислення (психологічний компонент).

Також доцільно визначити **рівні сформованості** компонентів фахових компетентностей у студентів за п'ятибальною шкалою:

– нульовий рівень – 1 бал (не завжди може відтворити отримані знання; частково виконує репродуктивні дії; не має уяви про роль ІКТ в професійній підготовці; не проявляє творчого мислення);

– низький рівень – 2 бали (відсутність мотивації до навчальної діяльності; пізнавальна інертність; епізодичний інтерес до навчання; мінімальна самостійна діяльність);

– середній рівень – 3 бали (немає чітко вираженої професійної спрямованості; пізнавальна активність потребує постійних мотивацій; залежність процесу самостійної діяльності від викладача; відсутня здатність до творчої самостійності та науково-дослідної роботи);

– просунутий рівень – 4 бали (створює завдання, які необхідні в професійній діяльності; усвідомлює необхідність інформатичної підготовки для майбутньої професійної діяльності; проявляє здібності до планування діяльності засобами формального моделювання; демонструє достатній рівень творчого мислення);

– високий рівень – 5 балів (позитивна професійна направленість; висока пізнавальна активність; творча самостійна активність; участь в науково-дослідній роботі; чітко виражена професійно-психологічна направленість на досягнення успіхів у професійній діяльності).

Всі рівні взаємопов'язані. І кожен попередній обумовлює наступний і входить до нього. [3]

Нами розроблено модель фундаменталізації навчання інформатичних дисциплін бакалаврів інформатики засобами систем комп'ютерної математики на основі міжпредметних зв'язків (рис. 1). Основними компонентами запропонованої моделі є: цільовий (завдання фундаментально підготовки); змістовий (інформатичні дисципліни, принципи, форми, методи та засоби навчання); критеріальний (критерії визначення рівня сформованості фахових компетентностей). Об'єднуючим фактором є мета підготовки, що забезпечує взаємозв'язок між усіма її елементами.

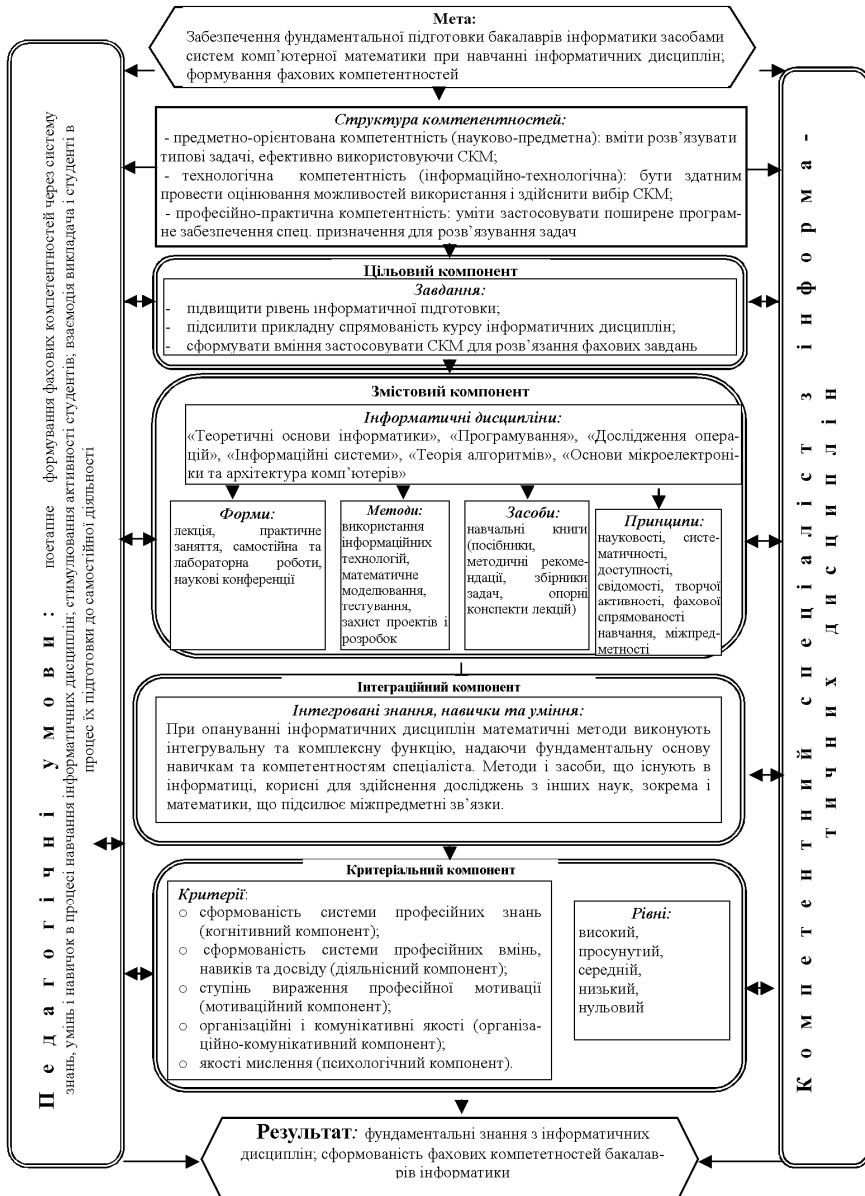


Рис. 1. Модель фундаменталізації навчання інформатичних дисциплін бакалаврів інформатики засобами СКМ

Другий напрям у фундаменталізації навчання полягає у тому, що

розглядається проблема добору змісту освіти на основі міжпредметних зв'язків загальнонаукових, загальнопрофесійних та інформатичних дисциплін.

Міжпредметні зв'язки – це вміння використовувати отримані знання та навички з однієї дисципліни при вивченні іншої. Фундаментальні знання визначаються різними внутрішніми та зовнішніми зв'язками, розкривають зміст певної предметної галузі, саме формування таких знань неможливе без застосування мимовільного або довільного формування міжпредметних зв'язків [1].

У навчальному процесі, з використанням міжпредметних зв'язків, у студентів розвиваються узагальнені інтелектуальні вміння, що характеризують певні види діяльності, загальні для ряду предметів.

Міжпредметні зв'язки дозволяють вплинути на розвиток творчої діяльності (використовувати отримані знання й уміння у новій ситуації, висувати нові гіпотези, вміння звернути увагу на різні характеристики об'єкта вивчення та ін.).

За допомогою міжпредметних зв'язків у навчально-виховному процесі можна стимулювати та спонукати майбутніх спеціалістів до професійного самовдосконалення. Використавши багатосторонні міжпредметні зв'язки, можливо виховувати та розвивати студентів, розв'язувати завдання навчання, подати основу для комплексного вирішення складних проблем.

Міжпредметні зв'язки дозволять забезпечити [5]:

- однакові вимоги до знань, вмінь та навичок з різних дисциплін;
- логічну послідовність при формуванні основних та неосновних понять;
- узгодженість навчальних планів нормативних дисциплін для взаємодопомоги при формуванні складних понять;
- використання отриманих раніше знань з інших дисциплін;
- економію часу при усуненні надлишкового дублювання знань з інших дисциплін;
- використання спільних законів та методик розв'язку задач та досліджень в різних дисциплінах;
- формування у студентів вмінь та навичок використання комп'ютерних технологій.

Фундаментом освіти повинно бути єдине ціле, тому різні дисципліни подаються не як окремі автономні дисципліни, а об'єднуються в певні фундаментальні дисципліни, об'єднані загальною функцією та міжпредметними зв'язками [7]. Для здійснення досліджень у різних науках можуть бути тісно пов'язані методи інформатики і математики. При опануванні інформатичних дисциплін математичні методи виконують

інтегровальну та комплексну функцію, надаючи фундаментальну основу навичкам та компетентностям спеціаліста. Методи і засоби, що існують в інформатиці, корисні для здійснення досліджень з інших наук, зокрема і математики, що підсилює міжпредметні зв'язки.

В інформатиці використовують такі міжпредметні методи та процедури, як аналіз і синтез, індукція і дедукція, візуалізація та формалізація, алгоритмізація і програмування, інформаційно-логічне, математичне та комп'ютерне моделювання, програмне керування, експертне оцінювання, ідентифікація та т.д. [2].

До сучасних програмних засобів, що дають змогу забезпечити міжпредметні зв'язки математики та інформатики, автоматизувати обчислювальний процес розв'язування задач прикладної спрямованості, зосередившись на побудові моделі та інтерпретації результатів обчислювального експерименту, є системи комп'ютерної математики (СКМ). **Системи комп'ютерної математики** – це поліфункціональні, універсальні програмні засоби, призначені для ефективного виконання математичних операцій з даними як у символічній, так і в числовій формі, візуалізації математичних закономірностей, проведення навчальних та наукових досліджень, а також моделювання процесів та явищ в різних предметних галузях.

Використання СКМ у навчальному процесі надає можливість за декілька секунд здійснити необхідні обчислення, дозволяючи студентам більшу частину навчального часу використати для розв'язання змістовних задач, набуття навичок побудови математичних моделей, інтерпретації та аналізу результатів обчислювального експерименту. Таким чином, СКМ виступають як засіб підвищення фундаментальності навчання інформатичних дисциплін, середовище для математичного моделювання та основа для побудови інформаційних навчальних середовищ. На базі СКМ можна розробити цілісні комп'ютерні курси, зовсім нові і орієнтовані на новітні інтерактивні технології. Ці курси можуть сильно відрізнятися від існуючих як по формі і змісту, так і по ролі вчителя.

Висновки. Використання СКМ надасть можливість забезпечити повноцінну навчальну, методичну та науково-дослідну діяльність, вводити інновації в навчальний процес, реалізувати принцип міжпредметності, поєднувати індивідуальний підхід з різними формами колективної діяльності. Тому, проблема розробки методик навчання фундаментальних дисциплін з використанням СКМ, гармонійне поєднання традиційних методичних систем навчання з ІКТ, створення на їх основі інформаційних навчальних середовищ залишається актуальною. Їх науково-математичне та методичне опрацювання є предметом подальших науково-педагогічних досліджень.

Література

1. Дем'яненко В. М. Методика навчання майбутніх вчителів інформатики апаратних і системних програмних засобів : дис. ... канд. пед. наук : 13.00.02 – теорія та методика навчання інформатики / Дем'яненко Віктор Михайлович ; Національний педагогічний університет імені М. П. Драгоманова. – К., 2003. – 195 с.
2. Казиев В. М. Некоторые системные и методологические аспекты информатики и информатизации [Электронный ресурс] / В. М. Казиев. – Режим доступа до статті : http://www.auditorium.ru/conf/conf_fulltext/kaziev.pdf
3. Калашникова Л. М. Взаимодействие учебного и производственного процессов как педагогическое условие организации преподавания специальных дисциплин / Л. М. Калашникова // Стандарты и мониторинг в образовании : научно-методический и информационный журнал. – 2011. – №4 (79) (июль-август). – С. 17-22.
4. Коньков Е. В. Компетентносный подход к обучению в общеобразовательной школе / Е. В. Коньков // Стандарты и мониторинг в образовании : научно-методический и информационный журнал. – 2011. – №4 (79) (июль-август). – С. 57-60.
5. Покришень Д. А. Програмно-педагогічне забезпечення міжпредметних зв'язків інформатики з математикою і фізикою у навчанні майбутніх інженерів : дис. ... канд. пед. наук : 13.00.02 – теорія та методика навчання (інформатика) / Покришень Дмитро Анатолійович ; Національний педагогічний університет імені М. П. Драгоманова. – К., 2010. – 240 с.
6. Раков С. А. Формування математичних компетентностей учителя математики на основі дослідницького підходу в навчанні з використанням інформаційних технологій : автореф. дис. ... доктора пед. наук : 13.00.02 – теорія та методика навчання (інформатика) / Раков Сергій Анатолійович ; Національний педагогічний університет імені М. П. Драгоманова. – К., 2005. – 44 с.
7. Суханов Б. М. Интеграция естественнонаучного и технологического знания / Борис Михайлович Суханов. – Л. : Изд-во ЛГУ, 1987. – 96 с.

РЕЗУЛЬТАТИ ВПРОВАДЖЕННЯ ЕЛЕМЕНТІВ МЕТОДИЧНОЇ СИСТЕМИ ФОРМУВАННЯ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ КОМПЕТЕНТНОСТЕЙ МАЙБУТНІХ ВЧИТЕЛІВ ІНФОРМАТИКИ В ПРОЦЕСІ НАВЧАННЯ ПРОГРАМУВАННЯ

О. М. Кривонос

Україна, м. Житомир, Житомирський державний університет
імені Івана Франка

Alexander.Kryvonos@zu.edu.ua

У контексті приєднання до Болонського процесу національна система вищої освіти переживає суттєві структурні та змістовні зміни. Процес навчання повинен мати на меті поетапне формування у студентів відповідної системи знань, певних умінь та навичок, ключових і професійних компетентностей, у тому числі інформаційно-комунікаційних. Сучасній школі, як вищій, так і середній, потрібні педагоги, котрі оперативно реагують на зміни соціального середовища, ефективно застосовують у навчанні досягнення науково-технічного прогресу, компетентно спрямовують особистісний розвиток учнів або студентів, творчо працюють над удосконаленням навчального процесу, виступають носіями та трансляторами сучасних технологій.

Актуальність проблеми та її недостатня теоретична й практична розробленість зумовили вибір теми нашого дослідження «Формування інформаційно-комунікаційних компетентностей майбутніх учителів інформатики в процесі навчання програмування».

Мета дослідження полягає в теоретичному обґрунтуванні, розробці та експериментальній перевірці елементів методичної системи формування інформаційно-комунікаційних компетентностей майбутніх вчителів інформатики у процесі навчання програмування.

Дослідження здійснювалося впродовж 2008-2012 рр. й охоплювало три етапи науково-методичного пошуку.

I етап (теоретико-узагальнювальний, 2008-2009 рр.) – теоретичне осмислення проблеми дослідження; вибір та обґрунтування теми; визначення об'єкта, предмета, мети; формулювання гіпотези та завдань дослідження, вивчення теоретичного стану проблеми аналізу філософської, педагогічної, психологічної літератури з теми дослідження, накопичення матеріалу дослідження; аналіз досвіду викладання програмування. Механізм реалізації компетентнісного підходу була вибрана методична система.

II етап (аналітико-синтетичний, 2009-2010 рр.) – проведено конста-

тувальний етап експерименту з метою виявлення загального стану навчально-виховного процесу, а саме: встановлення орієнтовного рівня знань, умінь і навичок студентів з інформатики, необхідних для успішного засвоєння дисциплін з комп'ютерних дисциплін та використання в майбутній професійній діяльності; проаналізовано зміст теоретичного матеріалу курсу з програмування, виходячи з цілей навчання у ВНЗ відповідних напрямів підготовки; проаналізовано зміст лабораторного практикуму з програмування і його роль в системі підготовки студентів – майбутніх вчителів інформатики; визначено найбільш раціональні методи, форми та засоби навчання програмування в системі підготовки майбутніх вчителів інформатики. Під час формувального етапу педагогічного експерименту в повному обсязі були використані елементи методичної системи, що були розроблені. Це дозволило провести необхідні спостереження та педагогічні виміри.

III етап (експериментально-підсумковий, 2010-2011 рр.) – на основі кількісних та якісних показників було проведено аналіз результатів педагогічного дослідження, співставлення з гіпотезою дослідження та формулювання висновків. Проведено порівняльний аналіз одержаних проміжних та кінцевих результатів педагогічного експерименту. Достовірність результатів дослідження забезпечувалась науковою обґрунтованістю вихідних теоретичних положень; адекватність методів, що використовувалися, меті та задачам дослідження; використанням математичних методів опрацювання результатів дослідницької роботи тощо.

Дослідницько-експериментальна робота проводилась на базі фізико-математичного факультету Житомирського державного університету імені Івана Франка.

Аналіз психолого-педагогічної літератури надав можливість визначити сучасні підходи до формування інформаційно-комунікаційних компетентностей на основі особистісно-орієнтованого та діяльнісного підходів в поєднанні з методами проблемно-розвиваючого навчання. Встановлено, що природа компетентності така, що вона може проявлятися тільки в органічному поєднанні з цінностями особистості за умови глибокого усвідомлення зацікавленості у створенні власного кінцевого продукту, тобто власній діяльності [3].

Під час дослідження розглянуто різні підходи виокремлення рівнів сформованості компетентності: визначено основні компоненти інформаційно-комунікаційних компетентностей: мотиваційно-ціннісний, змістово-проектувальний, когнітивно-операційний та особистісно-рефлексивний; окреслено рівні сформованості інформаційно-комунікаційних компетентностей; взявши за основу системне розуміння компетентностей, визначено критерії інформаційно-комунікаційних

компетентностей, що описують структурні і функціональні компоненти, дозволяють розглядати компетентності як стан, як процес і як результат. У якості критеріїв інформаційно-комунікаційних компетентностей майбутнього вчителя інформатики нами було виділено наступні – цілі та мотиви, фахові знання, фахові уміння, самооцінка та прагнення до самоосвіти [2].

Процес формування інформаційно-комунікаційних компетентностей майбутнього вчителя інформатики має інформаційну та діяльнісну складові. Інформаційна складова базується на засвоєнні навчального матеріалу та формуванні знань, знаходиться на початковому етапі формування кожної компоненти. Діяльнісна складова базується на інформаційній (на базі знань формуються уміння, навички, досвід діяльності). Рівень діяльнісної складової залежить від кількості виконаних практичних завдань. Підвищення складності виконуваних завдань сприяє формуванню досвіду діяльності та підвищенню рівня сформованості інформаційно-комунікаційних компетентностей в цілому.

Спроекована модель розглядалась нами з позиції системного, особистісного та діяльнісного підходів до процесу формування зазначених компетентностей, як сукупність закономірних, функціонально пов'язаних компонентів, що утворюють певну цілісну систему [1]. Виокремлення компонентів в моделі дозволило розбити її на блоки: цільовий, змістовний, організаційний, функціональний та результативний; які дають можливість уявити цілеспрямований процес формування відповідних компетентностей майбутніх вчителів інформатики.

З метою перевірки ефективності запропонованих елементів методичної системи формування інформаційно-комунікаційних компетентностей майбутнього вчителя інформатики під час навчання програмування нами було розроблено програму експериментальної роботи, що містила чотири етапи – попередньої дослідницької роботи, констатувальний, формувальний та аналітико-узагальнюючий.

Аналізуючи результати констатувального етапу експерименту, нами визначено стан сформованості інформаційно-комунікаційних компетентностей майбутніх учителів інформатики на основі розроблених рівнів. Передували формувальному етапу спостереження за діяльністю студентів на лекційних та лабораторних заняттях, аналіз анкет.

Визначений розподіл студентів за рівнями сформованості інформаційно-комунікаційних компетентностей дав нам можливість стверджувати, що близько двох третин студентів (68%) мають низький або середній рівень сформованості інформаційно-комунікаційних компетентностей.

Результати констатувального етапу експериментально підтвердили

необхідність розробки й впровадження у навчальний процес елементів методичної системи формування інформаційно-комунікаційних компетентностей майбутнього вчителя інформатики на основі компетентнісного підходу.

Метою розроблених нами елементів методичної системи є створення умов для реалізації інтересів і потреб майбутнього вчителя інформатики, вільного й усвідомленого вибору ним способів власної навчальної діяльності та їх удосконалення, підвищення рівня інформаційно-комунікаційних компетентностей, здатності до творчої самореалізації у майбутній професійній діяльності [4].

Дані експериментального дослідження (рис. 1, таблиця 1) засвідчили, що розроблені елементи методичної системи формування інформаційно-комунікаційних компетентностей майбутнього вчителя інформатики на основі особистісно-діяльнісного підходу забезпечує створення умов для більш ефективної підготовки фахівців.

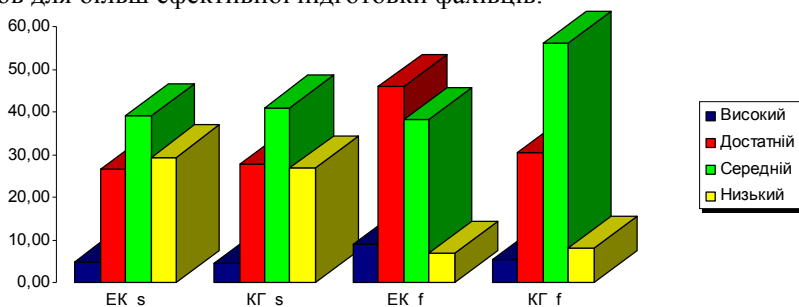


Рис. 1. Результати визначення рівня сформованості інформаційно-комунікаційних компетентностей на початку та в кінці експерименту

Таблиця 1

Рівень сформованості компетентностей	На початку експерименту				В кінці експерименту			
	Експериментальна група		Контрольна група		Експериментальна група		Контрольна група	
	К-ть	%	К-ть	%	К-ть	%	К-ть	%
Високий	5	4,90	5	4,46	9	8,82	6	5,36
Достатній	27	26,47	31	27,68	47	46,08	34	30,36
Середній	40	39,22	46	41,07	39	38,24	63	56,25
Початковий	30	29,41	30	26,79	7	6,86	9	8,04

Результативність впровадження запропонованих елементів методичної системи визначено через показники критеріїв інформаційно-комунікаційних компетентностей майбутнього вчителя інформатики за допомогою методів спостереження, анкетування, опитування, тестування

ня, моделювання тощо. Результати підсумкового зрізу з використанням методів статистичної обробки та порівняльного аналізу підтвердили позитивну динаміку формування інформаційно-комунікаційних компетентностей майбутніх учителів інформатики. Зміна рівнів інформаційно-комунікаційних компетентностей студентів експериментальних груп у кінці експерименту має таку тенденцію: відбувся перехід студентів з низького рівня на вищі рівні та збільшилася кількість студентів, у яких компетентність сформована на достатньому рівні (достатній – 46,08 %).

Порівняння показників експериментального дослідження свідчить про наявність вираженої зміни рівнів інформаційно-комунікаційних компетентностей в експериментальних групах і підтверджує ефективність запропонованих нами елементів методичної системи.

Проведене дослідження не претендує на всебічне розв'язання проблеми підготовки компетентних учителів інформатики. Передбачається продовження наукового пошуку за такими напрямками: вдосконалення змісту, форм, методів і технологій підготовки майбутніх учителів інформатики, залучення викладачів і студентів до використання нових методів навчання програмування, розробки навчально-методичного забезпечення з урахуванням особливостей реалізації компетентнісного підходу.

Література

1. Активные методы обучения педагогическому общению и его оптимизация / Под ред. А. А. Бодалева, Г. А. Ковалёва. – М. : Педагогика, 1983. – 136 с.
2. Кривонос О. М. Інформаційно-комунікаційні компетентності майбутніх вчителів інформатики // Вісник Київського національного університету імені Тараса Шевченка. – Серія «Соціологія. Психологія. Педагогіка» – Додаток 4, том III (15) – 2009 р. – Тематичний випуск №3. «Вища освіта України у контексті інтеграції до європейського освітнього простору» – С. 131–137.
3. Кривонос О. М. Компетентнісний підхід. Аналіз основних понять / Кривонос О. М. // Формування інформаційної компетентності студентів у процесі навчання природничо-математичних дисциплін : матеріали міжвузівського регіонального науково-методичного семінару 23 січня 2009 року. – Житомир, 2009. – С. 34–39.
4. Кривонос О. М. Педагогічні умови формування інформаційно-комунікаційних компетентностей / Кривонос О. М. // Освіта в інформаційному суспільстві: до 25-річчя шкільної інформатики : матеріали Всеукраїнської науково-практичної конференції. – К. : Київськ. ун-т імені Бориса Грінченка, 2010. – С. 55–56.

ФОРМИРОВАНИЕ ИССЛЕДОВАТЕЛЬСКИХ НАВЫКОВ У СТУДЕНТОВ МЛАДШИХ КУРСОВ ПРИ ИЗУЧЕНИИ ПРЕДМЕТА «ИНФОРМАТИКА» В ТЕХНИЧЕСКОМ ВУЗЕ

Г. И. Кулик

Украина, г. Днепрпетровск, Приднепровская государственная академия
строительства и архитектуры

Kulik.galina@mail.ru

Возможности использования информационных технологий в различных областях существенным образом изменили вид деятельности человека. Качественные изменения, связанные с активным использованием компьютера, внесли коррективы в подготовку специалистов различного профиля.

В значительной мере это касается инженерно-технических специальностей. Внедрение информационных технологий на всех этапах проектных и технологических работ предполагает, что инженер владеет специальными знаниями, профессионально использует компьютерные технологии в производственной деятельности.

Известно, что качественное высшее образование базируется на прочных школьных знаниях. Входные тесты, первичное анкетирование студентов показывает различный уровень подготовки выпускников средних школ в области точных наук и информатики. Кажущееся очевидным преимущество выпускников элитных школ не всегда гарантирует им высокие результаты в вузе. Анализ результатов при изучении информатики отражает проблемы как средней, так и высшей школы и требует решения этих проблем.

При проведении практических занятий по информатике наиболее результативным оказался следующий подход. По рассмотренному теоретическому материалу формулируется практическое задание и фиксируется время его выполнения. По истечении указанного времени, когда студенты самостоятельно выполняли это задание, проводится анализ результатов и отмечаются лучшие работы. После этого у доски разбирается решение задачи и прорабатываются типовые ошибки для такого рода задач (рис. 1).

Такой подход дает возможность дифференцировать работу студентов с различным уровнем подготовки. Более сильные студенты получают возможность проявить себя и заработать дополнительные баллы на модульном контроле, более слабые студенты при выполнении работы пытаются найти решение и проанализировать свои ошибки, допущенные в ходе работы.

Такое построение практических занятий позволяет активизировать работу всех студентов, заинтересовать их в конечном результате, выявить лидеров и проработать наиболее сложные вопросы рассматриваемой темы.

Подход, вовлекающий в активную работу основную массу аудитории, применим и при решении исследовательских задач. Бытует мнение, что студентов младших курсов неэффективно привлекать к выполнению научно-исследовательской работы. В качестве аргументов приводится недостаточный уровень знаний, квалификации и т.п.

Однако, всем известно, что с такими навыками человек не рождается, а приобретает их непосредственно в ходе выполнения работы.

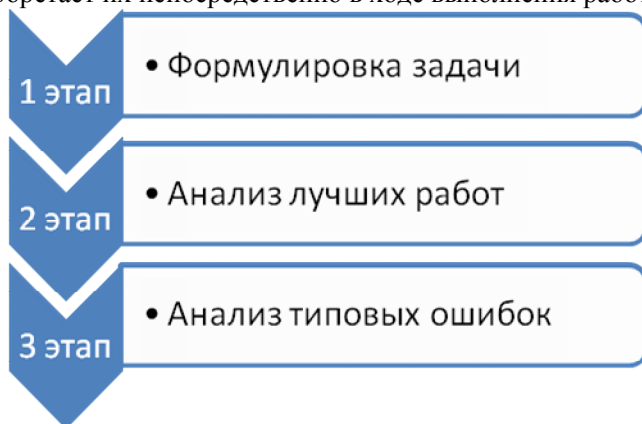


Рис. 1. Схема проведения практических занятий

Различное количество часов в школьном курсе информатики и различное наполнение этих часов создает проблему, с которой сталкиваются ежегодно преподаватели дисциплин, связанных с информационными технологиями.

Основной задачей преподавателя в сложившейся ситуации является проведение занятий, которые, с одной стороны, обеспечивают необходимый уровень знаний студентов, с другой стороны, не позволяют утратить интерес той части аудитории, которая обладает более прочными знаниями.

Активные методы проведения занятий и изучения программного материала позволяет не только повысить успеваемость студентов, что само по себе очень важно. Привычка к активному восприятию материала дает и отложенный во времени результат. Постепенно у студента формируются навыки, необходимые для решения более сложных задач, выполнения самостоятельных исследований.

Среди заданий, способствующих активизации знаний студентов, выделяются деловые игры, где одни выступают в качестве разработчиков программы, а другие в качестве экспертов. Также пользуются популярностью тестовые задания, в которых приводятся фрагменты программ, содержащих ошибки. Задача студента – найти и устранить эти ошибки за отведенный промежуток времени. В отличие от тестовых заданий, предполагающих выбрать правильный ответ из нескольких представленных вариантов, такой вид работы позволяет качественно изменить ход занятий.

Из пассивных слушателей, «поглощающих» информацию, студенты становятся оппонентами, критически воспринимающих информацию.

Это, пожалуй, самое важное, что должен научиться делать будущий специалист. И такой подход к знаниям, образованию в целом, позволит активно использовать полученные знания.

Известно, что самым эффективным способом активизации полученных знаний является участие студента в решении задач, непосредственно связанных с исследованиями в предметной области.

Активное применение знаний, непосредственное участие в НИРС, реальных курсовых и дипломных проектах позволяет развиваться и реализоваться творческим способностям студента, приблизить его к решению специальных задач, научиться быстро реагировать на смену условий деятельности и адаптироваться к новым, все возрастающим требованиям к современному специалисту.

В современных условиях технологии обучения, ориентированные на «транслирование» и усвоение готовых знаний, уступают место новым, активным методам в обучении, развивающим аналитические и творческие способности студента.

Идея активного обучения не нова – впервые она была сформулирована в начале XX века американским философом и педагогом Джоном Дьюи. Он утверждал, что традиционной (пассивной) системе образования, основанной на приобретении и усвоении знаний, нужно противопоставить активную систему, чтобы новые знания извлекались человеком из практической деятельности и личного опыта.

Получившая развитие идея активного обучения позволила сформировать две концепции: «пирамида обучения» (learning pyramid) и «конус опыта Эдгара Дейла» (Dale's cone of experience).

«Конус опыта» наглядно иллюстрирует, каких разных образовательных результатов можно добиться, используя различные средства обучения.

Концепция «пирамиды обучения» также демонстрирует зависимость между методами обучения и степенью усвоения материала. Так,

классическая лекция (монолог преподавателя, который не сопровождается слайдами и другими иллюстрациями) обеспечивает освоение в среднем около 5% содержания.

Предмет «Информатика» в учебном процессе занимает особое место, поскольку обучение этому предмету дает возможность творческого подхода к решению ряда проблем, стоящих перед преподавателем и студентом в современных условиях, предоставляет возможность подготовить студента к научно-исследовательской работе.

Поскольку предмет «Информатика» читается на первых курсах, студенты не обладают необходимыми навыками для самостоятельного исследования. Для вовлечения наиболее способных студентов к научным исследованиям формируются группы по 2-3 человека и осуществляется совместная работа над проектом.

Такой подход позволяет распределить усилия и более эффективно организовать работу начинающих исследователей в соответствии с их наклонностями и уровнем подготовки.

При изучении определенных разделов курса информатики студенты приобретают начальные навыки НИР и могут реализовать на практике концепцию активного обучения, участвуя в исследованиях под руководством преподавателя.

Таковыми разделами являются следующие:

- «Обработка текстовой информации»;
- «Подготовка презентаций с использованием возможностей анимационных эффектов»;
- «Обработка экспериментальных данных по результатам исследований»;
- «Моделирование и анализ данных»;
- «Эффективный поиск информации в сети Интернет».

Результатом научно-исследовательской работы студентов является участие в студенческих конкурсах научных работ, выступление с докладами на студенческих научных конференциях. На ежегодной студенческой научной конференции, которая проводится в нашем вузе, традиционно представлены доклады, подготовленные студентами по кафедре прикладной математики.

Тематика выполненных студенческих докладов достаточно разнообразна. Это работы связанные с историей развития предмета, отдельными эпизодами становления того или иного раздела информатики, обзоры последних достижений в области информационных технологий. Подготовка таких докладов позволяет сформировать навыки эффективного поиска и анализа источников по выбранной теме. Работы, носящие характер реферативных, обычно представляют студенты первых курсов.

Работы студентов старших курсов чаще представлены как авторские разработки, предполагающие получение и обсуждение самостоятельных результатов исследования.

В обязательном порядке студенческие доклады сопровождаются презентациями, по итогам работы конференции издается сборник тезисов студенческих работ, определяются лучшие доклады в различных номинациях и отмечаются авторы наиболее интересных докладов.

Участие студентов в НИРС по кафедре прикладной математики



Рис. 2. Участие студентов в НИРС

Информация о восприятии студентов получена из анкет, которые осуществляют обратную связь студентов-участников и организаторов конференций, что позволяет отметить возрастающий интерес студентов к подобным мероприятиям, понимание ими полезности такой работы как способа активизации полученных знаний.

Большинство преподавателей начинали свой профессиональный путь именно с решения исследовательских задач в студенческие годы. Интерес, который появляется при решении поставленной задачи, творческий азарт, который приходит одновременно с первыми успехами, может существенным образом повлиять на выбор дальнейшего приложения сил молодого специалиста.

Очень многие ученые с благодарностью вспоминают своих первых наставников в научной работе. Результат совместных приложенных усилий привлекает молодые кадры в ряды исследователей, которые про-

должат научную работу по выбранной специальности.

Необходимость повышения эффективности проведения аудиторных занятий и активные методы обучения, позволяющие реализовать в исследовательских задачах базовые знания, позитивно сказываются и на процессе обучения и на дальнейшем карьерном продвижении выпускников.

Анализ результатов научной работы со студентами за последние несколько лет демонстрирует положительную динамику как по количеству участников студенческих конференций, олимпиад, так и по уровню выполненных работ.

Литература

1. Dale E. Audiovisual methods in teaching, third edition / Edgar Dale. – New York : The Dryden Press; 1969. – 719 p.

2. Бусигін Б. С. Прикладна інформатика : підручник для студентів комп'ютерних спеціальностей / Бусигін Б. С., Коротенко Г. М., Коротенко Л. М. – Дніпропетровськ : Національний гірничий університет, 2004. – 559 с.

3. Гессен С. И. Теория воспитания Джона Дьюи / С. И. Гессен // Педагогические сочинения. – Саранск, 2001. – С. 399.

4. Информатика : базовый курс / С.В. Симонович и др. – СПб. : Питер, 2001. – 640 с.

5. Кулик Г. И. Интернет: новые возможности в формировании научного кругозора студентов / Кулик Г. И. // «Интернет – освіта – наука – 2010», сьома міжнародна конференція ІОН-2010 : збірник матеріалів конференції. – Вінниця : ВНТУ, 2010. – С. 13.

ВІДПРАЦЮВАННЯ ЗНАНЬ І ВМІНЬ З ІНФОРМАТИКИ

І. М. Лукаш

Україна, м. Чернігів, Чернігівський національний педагогічний
університет імені Т. Г. Шевченка
lukash2006@mail.ru

Під відпрацюванням розуміють удосконалення знань і вмінь, що виникли у процесі засвоєння. Якщо у процесі засвоєння знання і вміння набувають повноти, глибини, правильності, то у процесі відпрацювання форма дії від матеріальної (з опорою на джерело) стає розумовою (без опори на джерело), розгорнута дія (покрокова) стає згорнутою, свідомою дія перетворюється у підсвідому (автоматизовану), повільна дія становиться швидкою, а важка – легкою [2]. Це досягається поєднанням запам'ятовування знань і тренуванням умінь через застосування відповідних вправ.

Процеси відпрацювання знань і вмінь поділяються на поетапні і непоетапні. Поетапний процес відпрацювання здійснюється за такими етапами: матеріальним (виконання вправ з опорою на джерело знань), вербальним (відновлювання знань стосовно розв'язування задачі в усній або письмовій формі перед застосуванням їх на практиці), розумовим (згорнута дія). Непоетапний процес відпрацювання знань і вмінь передбачає розв'язування вправ без опори методом «проб і помилок», що з одного боку може призвести до формування неправильної основи орієнтовних дій, а з іншого – спонукати до самостійної, дослідницької діяльності.

Одним із видів поетапного відпрацювання знань і вмінь є класичне поетапне формування дії за П. Я. Гальперінім, коли здійснюється поетапне тренування при розв'язуванні задач з опорою на схему орієнтовної основи дії [1]. Звісно, що в межах сучасної педагогічної науки розробляються загальні дидактичні методи навчання, які по різному застосовуються при вивченні різних предметів та окремих тем з цих предметів. Виникає необхідність адаптації загальних дидактичних методів навчання до певної тематики предмету, до певних умов викладання.

Вчителі (зокрема, вчителі інформатики) намагаються самостійно вирішувати цю проблему, про що свідчить значна кількість публікацій конспектів уроків у педагогічній пресі. Але відсутність систематизації адаптованих методів навчання ускладнює роботу вчителя, не дає загального розуміння методики навчання особливо для вчителя-початківця. Значні доробки з методики навчання інформатики наведені в роботах Н. В. Морзе, М. П. Лапчика, І. Г. Семакіна, Є. К. Хеннера та ін. Але в

більшості випадків вчителям надаються рекомендації, що при вивченні певних тем доцільно застосовувати відповідні методи навчання без наведення їх конкретних реалізацій.

Метою статті є розгляд прикладів адаптації деяких загальних дидактичних методів поетапного і непоетапного відпрацювання знань і вмінь у процесі викладання інформатики, які застосовуються при вивченні студентами курсу методики навчання інформатики.

Розглянемо кілька методів поетапного відпрацювання знань і вмінь.

Метод відпрацювання у вправах при *розв'язуванні задач за зразком*. Розв'язування задач однакової дидактичної складності.

Визначення дидактичної складності задачі ґрунтується на визначенні кількості операцій, які необхідно виконати при її розв'язуванні. Задача вважається складнішою, якщо її розв'язування потребує виконання більшої кількості операцій.

Згідно з методом «розв'язування задач за зразком» завдання для учнів повинно складатися з кількох задач, перша з яких повинна бути установчою, тобто, на основі якої розглядається орієнтовна основа дії розв'язування подібних задач. Інші задачі в завданні подібні до установчої. Наприклад, завдання на відпрацювання функції ЕСЛИ в MS Excel.

Задача 1 (установча). Прийом на роботу здійснюється на конкурентній основі. Умови прийому вимагають 20 років робочого стажу та віку не більш 42 років. Визначити, чи буде людина прийнята на роботу.

Задача 2. До школи танців приймаються юнаки та дівчата, що мають зріст не нижче 168 см. та не вище 178 см. Вага претендента повинна співвідноситись зі зростом за формулою:

$$\text{значення ваги} \leq \text{значення зросту} - 115.$$

Визначити, чи буде претендент прийнятий до школи.

Розв'язування задач різної дидактичної складності. Дидактично складна задача передбачає можливість її розчленування на простіші дидактично доцільні ситуації. Розглянемо подання однієї тієї самої задачі на різних рівнях дидактичної складності.

Задача 1. Відомі дані метеостанції про кількість опадів (в мм), що випадали кожного місяця протягом двох років.

Місяць	2010 рік	2011 рік
Січень	37,2	34,5
Лютий	11,4	34,1
...
Грудень	21,2	22,3

Визначити: загальну кількість опадів, що випали за кожний рік; кількість опадів, що випали за два роки.

Задача 2. Відомі дані метеостанції про кількість опадів (в мм), що

випадали кожного дня протягом двох років. Якщо в певний день опадів не було, то у відповідній комірці ставиться символ «←».

Число місяця	2010 рік				2011 рік			
	Січень	Лютий	...	Грудень	Січень	Лютий	...	Грудень
1	–	40		–	12	–		35
2	30	12		14	–	13		25
...
31	23	-		35	24	-		14

Визначити:

1) загальну кількість опадів, що випали кожного місяця за кожний рік;

2) загальну кількість опадів, що випали за кожний рік;

3) кількість опадів, що випали за два роки.

Друга з наведених задач є дидактично складнішою у зв'язку з тим, що передбачає виконання додаткових операцій: знаходження кількості опадів за кожний місяць, що є вже відомою величиною в першій задачі, та застосування функції ЕЧИСЛО, для перевірки, чи є числом значення комірки, яке додається до суми.

Розв'язування задач з різним ступенем повноти умови. За ступенем повноти інформації, наданої в тексті умови, виділяються наступні типи задач:

1. В умові задач першого типу міститься готова відповідь на питання задачі. Учні необхідно тільки виділити її і зрозуміти, що це і є відповідь на питання.

2. Текст умови задачі другого типу не містить готової відповіді на питання, але надає необхідний обсяг інформації для її виконання.

3. Текст умови задачі третього типу містить тільки частину необхідної інформації. Іншу частину інформації учень повинен знайти сам з інших джерел інформації, отриманих раніше знань, шляхом розмірковування тощо.

4. Текст умови задачі четвертого типу взагалі не містить інформації, необхідної для її розв'язування.

Задача 1. Об'єкти мають наступне оголошення і програмну реалізацію своїх методів (Turbo Pascal):

```
Type
Obj=object
  Constructor Create;
  Procedure Print;
```

```

    Function Stroka:string;Virtual;
end;

Ob2=object (Ob1)
    Function Stroka:string;Virtual;
end;

Var S1:Ob1;S2:Ob2;

Constructor Ob1.Create;
Begin
End;

Procedure Ob1.Print;
Begin
    Writeln('Гарна '+Stroka);
End;

Function Ob1.Stroka:string;
Begin
    Stroka:='погода';
End;

Function Ob2.Stroka:string;
Begin
    Stroka:='рослина';
End;

```

Що буде надруковано в результаті виконання операторів:

а) S1.Create;S1.Print; б) S2.Create;S2.Print;

Відповідь: а) Гарна погода; б) Гарна рослина.

Задача 2 (повно задана умова). Об'єкт *Точка* характеризується координатами x , y та кольором, має метод побудови свого зображення. Атрибути *Точки* та методи встановлення нових значень координат успадковуються її об'єктами-нащадками *Кругом* і *Квадратом*. *Круг* додатково має атрибут – радіус, а *Квадрат* – довжину сторони. В кожному з цих об'єктів перевизначається метод малювання їх зображення. Реалізація переміщення фігури по екрану в графічному режимі передбачає малювання фігури в попередній позиції, а потім кольором малюнку – в новій позиції. Написати програму переміщення цих фігур по екрану за деякою траєкторією.

Задача 3 (неповно задана умова). Реалізація переміщення фігури по екрану в графічному режимі передбачає малювання фігури в попередній позиції, а потім кольором малюнку – в новій позиції. Геометричні фігури *круг*, *квадрат* і *точка* знаходяться в ієрархічній залежності (*Круг* і

Квадрат є нащадками *Точки*) і мають власні методи малювання. Написати програму переміщення цих фігур по екрану за деякою траєкторією.

Задача 4 (частково задана умова). Описати об'єкти: *Круг, Квадрат, Точка*. Використати ці об'єкти для переміщення відповідних фігур по екрану за деякою траєкторією.

Так, задача 1 передбачає знаходження відповіді по вже наведеному розв'язку. В задачі 2 надається повне описання об'єктів, їх ієрархічної залежності, перераховуються всі необхідні атрибути об'єктів та методи, надається рекомендація щодо розв'язування. Умова задачі 3 побудована так, що учень отримує тільки частину інформації про об'єкти системи, про деякі атрибути і методи (про метод малювання, атрибути позиція і колір), про їх ієрархічну залежність та рекомендацію стосовно розв'язування задачі. В умові задачі 4 вказуються тільки задані об'єкти та їх поведінка (об'єкти переміщуються). У випадку розв'язування останньої задачі учень повинен самостійно виконати комплекс розумових дій для побудови ієрархічної структури об'єктів та програмної реалізації їх поведінки.

Серед методів непоетапного відпрацювання знань і вмій розглянемо методи: розв'язування задач на основі попереднього навчального досвіду та розв'язування задач методом «проб і помилок».

Розв'язування задач на основі попереднього навчального досвіду. Попередній навчальний досвід може бути пов'язаний як з вивченням інших шкільних предметів, так і з вивченням попередніх тем з того самого предмету. Перший випадок передбачає розв'язування задачі, фабула якої стосується цього іншого предмету. Інший випадок, який є досить цікавим з точки зору реалізації проблемного методу навчання, пов'язаний з розв'язуванням тих самих задач в різному програмному забезпеченні.

Задача 1. Протабулювати функцію $y=x^2+5x+6$ на відрізку $[a, b]$ з кроком h (розв'язати задачу в різних програмних середовищах).

На основі відмінностей між програмними засобами можна запропонувати доповнення до завдань, що дозволяють реалізувати особливості та переваги одного програмного забезпечення порівняно з іншим.

Розв'язування задач методом «проб і помилок».

Метод полягає в тому, що вчитель спочатку не подає жодних пояснень та пропонує учню розв'язати певні задачі. Учень набуває знання, проводячи аналіз власних помилок, допущених при розв'язуванні задач.

Приклад реалізації цього методу розглянемо на прикладі розв'язування задачі в MS Excel з метою пояснення випадків застосування абсолютної та відносної адресації комірок.

Завдання. Створити таблицю «Середня заробітна платня» за зразком

таблиці 1. Для кожної з професій за введеним значенням заробітної плати в у.о. за тиждень підрахувати заробітну плату за місяць в у.о. та грн. У зв'язку з тим, що курс у.о. постійно змінюється, його значення ввести в окрему комірку, наприклад, в комірку A10, адресу якої використати у відповідній формулі.

Таблиця 1

	A	B	C	D
1	Професія	з.п. за тиждень (у.о.)	з.п. за місяць (у.о.)	з.п. за місяць (грн...)
2				

Обчислення заробітної плати за місяць в у.о. передбачає введення в комірку C2 формули: =B2*4. Копіювання цієї формули в комірки, що розташовані нижче, призведе до обчислення заробітної плати за місяць в у.о. для кожної з професій. Більшість учнів намагаються обчислити заробітну плату в грн. аналогічним чином, вводячи в комірку D2 формулу: =C2*A10, та копіюючи її нижче. Цим самим вони створюють помилкову ситуацію, в чому впевнюються, побачивши нулі в усіх, окрім першої, комірках. Переглядаючи зміст комірок стовпика D, можна помітити, що після копіювання формула в комірках змінюється: =C2*A10; =C3*A11; =C4*A12 тощо. Тобто змінюється адреса комірки, де зберігається курс у.о. Щоб запобігти такої помилкової ситуації, адресу комірки, що не повинна змінюватись у формулі при копіюванні, необхідно записати в абсолютній адресації: =C2*\$A\$10. Внесення змін до формули та повторне копіювання виправить помилку в зазначених обчисленнях.

Адаптація загальних дидактичних методів навчання до вивчення інформатики акумулює досвід викладання інформатики, який може бути корисним особливо для майбутніх вчителів. Наявність системи адаптованих методів надає вчителю можливість урізноманітнити процес відпрацювання знань і вмінь, зробити його більш цікавим і мотивованим для учня.

Література

1. Гальперин П. Я. Общий взгляд на учение о так называемом поэтапном формировании умственных действий, представлений и понятий / Гальперин П. Я. // Вестник Московского университета. Серия 14. Психология. – 1998. – №2. – С.3-7.
2. Ильясов И. И. Проектирование курса обучения по учебной дисциплине / Ильясов И. И., Галатенко Н. А. – М. : Логос, 1992. – 207 с.

ВИКОРИСТАННЯ МОВИ ПРОГРАМУВАННЯ DELPHI У ПРОЦЕСІ ФОРМУВАННЯ ІНФОРМАТИЧНОЇ КОМПЕТЕНТНОСТІ МАЙБУТНІХ ІНЖЕНЕРІВ

О. А. Медведєва

Україна, м. Краматорськ, Донбаська державна машинобудівна академія
omedvedeva@mail.ru

Постановка проблеми. Розвиток людського суспільства, прискорений розвиток науки, швидке поновлення техніки, поява принципово нових технологій та впровадження їх у виробництво потребують все більш жорстоких вимог від системи освіти, задовольнити які за рахунок традиційних засобів навчання стає практично неможливо. Тому останнім часом активно дискутується питання підготовки майбутнього спеціаліста на засадах компетентнісного підходу як найефективнішого для освітнього процесу, такого, що забезпечує підготовку фахівця відповідно до сучасних вимог виробництва та суспільства взагалі [1].

Однією з основних складових в структурі компетентності майже всі науковці виділяють інформатичну компетентність [2] (компетентність у галузі інформатики), під якою розуміють системний обсяг знань, умінь, навичок та здібностей особистості в галузі методів і засобів отримання, опрацювання, перетворення, передавання, використання і розповсюдження інформації. Тому формування інформатичної компетентності стає однією із проблем підготовки майбутніх фахівців у ВНЗ.

Аналіз останніх досліджень і публікацій. Сучасні освітні технології тісно пов'язані з інформатизацією навчального процесу. Аналіз вітчизняного і зарубіжного досвіду використовування інформаційних технологій в навчальному процесі, робіт А. П. Єршова, М. І. Жалдака, Ю. І. Машбиця, Н. В. Морзе, Ю. С. Рамського, Н. Ф. Тализіної, Ю. В. Триуса та інших дозволили визначити найбільш важливі, істотні положення теорії комп'ютеризації навчання, адаптованої до особливостей національної системи вищої освіти.

Ціль статті. Метою статті є демонстрація можливостей застосування мови об'єктно-орієнтованого програмування Delphi в інженерній діяльності на заняттях з інформатики та комп'ютерної техніки.

Основний матеріал дослідження. Одним із розділів курсу «Інформатика та комп'ютерна техніка», який викладається студентам інженерного напрямку навчання, є «Мова програмування». До складу цього розділу, зокрема, входить програмування обробки двомірних масивів (вибіркове опрацювання масиву, пошук у масиві найбільшого (найменшого) елемента тощо). Бажано, щоб ці завдання носили прикладний характер.

Наприклад, для ілюстрації застосування алгоритмів обробки двовимірного масиву можна запропонувати студентам завдання з теорії статистичних рішень.

Ухвалення технічного рішення являє собою вибір одного з деякої множини розглянутих варіантів: $E_i, i = \overline{1, m}$.

Якщо ми прийняли варіант технічного рішення E_i , то, як правило, кожному такому варіанту відповідає деякий набір зовнішніх умов функціонування $F_j, j = \overline{1, n}$.

Функціонування технічного рішення E_i в умовах F_j характеризується числом a_{ij} – ефективністю функціонування рішення i в умовах j . Таким чином, ми одержимо двовимірний масив (табл. 1).

Інженерові необхідно, маючи у своєму розпорядженні двовимірний масив (a_{ij}) , прийняти якийсь, по можливості, найбільш вигідне технічне рішення E_i .

Таблиця 1

Матриця ефективності функціонування рішення

Рішення \ Умова	F_1	F_2	...	F_j	...	F_n
E_1	a_{11}	a_{12}		a_{1j}		a_{1n}
E_2	a_{21}	a_{22}		a_{2j}		a_{2n}
...						
E_i	a_{i1}	a_{i2}		a_{ij}		a_{in}
...						
E_m	a_{m1}	a_{m2}		a_{mj}		a_{mn}

Теорія статистичних рішень пропонує декілька критеріїв оптимальності вибору рішень (Вальда, Севіджа, Гурвіца, Лапласа).

Розглянемо один з цих критеріїв – критерій Вальда.

Відповідно до критерію Вальда як оптимальна вибирається стратегія, що гарантує виграш не менший, ніж «нижня ціна гри з природою».

Правило вибору рішення відповідно до критерію Вальда полягає в наступному:

а) у кожному рядку матриці (a_{ij}) необхідно знайти мінімальний елемент: $b_i = \min_j (a_{ij})$. У результаті одержимо одновимірний масив $b_i, i = \overline{1, m}$;

б) у знайденому одновимірному масиві потрібно знайти максимальний елемент: $l_r = \max_i b_i = \max_i (\min_j (a_{ij}))$.

Номер $r \in$ номером оптимального технічного рішення.

Вибране таким чином рішення повністю виключає ризик. Це озна-

чає, що людина, яка ухвалює рішення не може зіткнутися з гіршим результатом, ніж той, на який вона орієнтується. Ця властивість примушує вважати критерій Вальда одним з фундаментальних. Тому в технічних задачах він застосовується частіше за все як свідомо, так і неусвідомлено. Проте в практичних ситуаціях зайвий песимізм цього критерію може виявитися дуже невідгидним.

Результат роботи програмної реалізації цього критерію в Delphi представлений на рис. 1.

В якості індивідуального завдання студентам можна запропонувати програмування процесу вибору оптимального рішення за іншими критеріями (Севіджа, Гурвіца, Лапласа). Особливу увагу при цьому потрібно приділяти аналізу отриманих результатів. Цікавим може виявитись порівняння отриманих оптимальних рішень за різними критеріями.

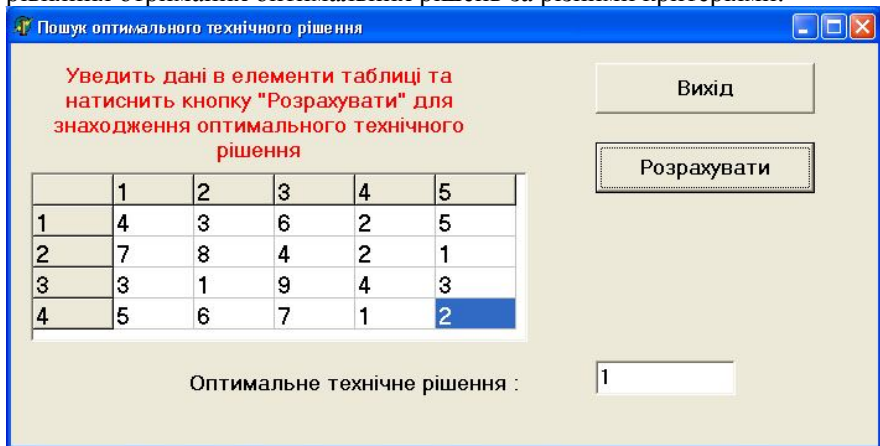


Рис. 1. Визначення оптимального технічного рішення за критерієм Вальда

Література

1. Петухова Л. Є. Інформатична компетентність майбутнього фахівця як педагогічна проблема / Л. Є. Петухова // Комп'ютер у школі та сім'ї. – 2008. – №1. – С. 3–5.
2. Бібік Н. М. Компетентнісний підхід: рефлексивний аналіз застосування / Н. М. Бібік // Компетентнісний підхід у сучасній освіті: світовий досвід та українські перспективи. Бібліотека з освітньої політики / під заг. ред. О. В. Овчарук. – К. : К.І.С, 2004. – С. 47–52.
3. Вентцель Е. С. Исследование операций: задачи, принципы, методология / Е. С. Вентцель. – М. : Наука, 1988. – 245 с.
4. Жак С. В. Математическое программирование. Нелинейные и стохастические задачи / С. В. Жак. – Ростов-на-Дону : РГУ, 1972. – 90 с.

РІВНІ СФОРМОВАНOSTI КОМПЕТЕНТНОСТІ У ПРОГРАМУВАННІ

І. С. Мінтій

Україна, м. Кривий Ріг, Криворізький національний університет
ipm_mintiy@mail.ru

Метою навчання програмування майбутніх вчителів за компетентнісного підходу є формування в них компетентності в програмуванні.

Особливістю формування компетентності в програмуванні є її взаємозв'язок з іншими, як спеціальними професійними (компетентностями з математичної інформатики, з інформаційних технологій та фундаментальних природничо-математичних дисциплін), так і загальнопрофесійними (ІКТ-компетентностями) та ключовими (навчальною, соціальною, ІКТ та підприємницькою).

Формування компетентності в програмуванні має акумулюючий характер: нові знання, що здобувають студенти, доповнюють їхню систему знань; уміння, навички, способи діяльності, яких набувають студенти, інтегруються з уже сформованими. Застосування здобутих знань, сформованих умінь, навичок, досвіду навчально-пізнавальної діяльності при розв'язуванні навчальних задач, задач майбутньої професійної діяльності та рефлексування цього процесу призводить до формування компетентностей певних рівнів. Тому, розглядаючи процес набуття майбутніми вчителями інформатики компетентності в програмуванні при вивченні деякої дисципліни слід спочатку проаналізувати попередній етап формування її основних складових та визначити шляхи подальшого їх формування [4, 62].

Формування компетентності в програмуванні у майбутніх вчителів інформатики в педагогічному університеті розпочинається в процесі вивчення дисципліни «Шкільний курс інформатики» та курсу за вибором «Вступ до програмування».

Про наявність тієї чи іншої компетентності взагалі говорити некоректно, оскільки процес її формування може бути досить тривалий і здійснюватися під впливом різних факторів: навчання у закладах освіти, професійної діяльності, міжособистісного спілкування тощо. Тому далі, говорячи про набуття студентами певних компетентностей, будемо розуміти їх сформованість на певному рівні.

Таким чином, процес формування компетентності в програмуванні є перехід від одного рівня (нижчого) до іншого (вищого), а для оцінки рівня її сформованості необхідно виокремити рівні її сформованості та критерії, які дозволяли б здійснювати перевірку рівня сформованості

компетентності.

Аналіз літератури дозволив сформулювати висновок, що не існує одностайної думки стосовно кількості рівнів сформованості компетентності.

Так, В. В. Беспалова виокремлює п'ять основних рівнів (низький, нижче середнього, середній, вище середнього та високий), О. М. Спінін – шість (початковий, мінімально-базовий, базовий, підвищений, поглиблений та дослідницький). Ряд вітчизняних дослідників (О. Є. Антонова, О. А. Дубасенюк, В. М. Єремєєва, М. В. Левківський, Н. Г. Сидорчук та ін.) співвідносять запропоновані Б. Блумом категорії цілей з рівнями сформованості компетентностей: знання – репродуктивний рівень, розуміння – адаптивний рівень, застосування – конструктивний рівень, аналіз – творчий рівень, синтез – дослідницький рівень, оцінювання – оцінно-узагальнюючий рівень.

Є. М. Смирнова-Трибульська виділяє дев'ять рівнів, що згруповані у три блоки: 0-2 – базовий (елементарний) рівень, 3-5 – середній (функціональний) рівень, 6-8 – просунутий (системний) рівень.

М. С. Головань розвиток компетентності у студентів розглядає «як незворотну, закономірну, цілеспрямовану зміну внутрішньої структури інформатичної компетентності і зовнішніх форм її прояву, у результаті чого виникають нові багаторівневі якісні її стани, основою яких є діалектична єдність можливого і дійсного, а також як саморегульований процес, тобто внутрішньо необхідний рух, «саморух» від наявного рівня компетентності до вищого відповідно до стадій цього процесу. Розвиток компетентності як системи забезпечується кількісними, якісними і структурними перетвореннями її елементів у процесі зміни стадій руху – становлення, активного розвитку, саморозвитку. У формуванні компетентності студентів М. С. Головань виділяє три рівні (низький, середній, високий) та три стадії (становлення, активного розвитку, саморозвитку). Стадії «горизонтального» просування (становлення, активного розвитку, саморозвитку) відбивають кількісне накопичення «критичної маси» суб'єктивних характеристик компетентності в кожного студента. «Вертикальне» просування – це якісний стрибок як перехід на вищий рівень розвитку.

На стадії становлення відбувається засвоєння студентами знань, вироблення умінь на репродуктивному рівні, формування мотивації до вивчення предмету та позитивного ставлення до програмування. На стадії активного розвитку студенти осмислено оперують уміньми та знаннями, мають потребу в особистій самореалізації в інформаційному середовищі, мають такі розвинені якості, як рефлексивність, креативність, критичність мислення, мають сформовані навички саморегуляції діяльності.

Основна мета стадії саморозвитку – розвиток самостійності, творчої активності, самоорганізації та самоуправління, актуалізація потреби у саморозвитку [1, 19].

Я. Б. Сікора, як і О. С. Меньяйленко, Г. В. Монастирна та М. В. Жукова виділяє чотири рівні сформованості компетентності: адаптивний (низький), алгоритмічний (середній), частково-пошуковий (достатній), творчий (високий).

Адаптивний рівень характеризується недостатньою сформованістю професійних намірів, відсутністю необхідних знань та вмінь, репродуктивним виконанням діяльності, неадекватною самооцінкою.

Для алгоритмічного рівня характерним є епізодичний інтерес до професії, недостатнє вміння використовувати наявні знання, нестійка потреба у самовдосконаленні.

Частково-пошуковий рівень відрізняється розвиненою суб'єктною позицією, яка виявляється в усвідомленості своїх дій та можливостей, прагненні до прийняття рішень, внесенні змін при використанні запозиченого досвіду, наявністю інтересу до професії, розумінням її значущості, проте недостатньою чіткістю у визначенні цілей формування компетентності.

Творчий рівень характеризується сформованістю стійкого інтересу до професії; здатністю до нестандартного розв'язання завдань, поглиблення знань та прийняттям усвідомлених рішень з урахуванням прогнозування наслідків своїх дій; прагненням до самовираження, самовдосконалення, об'єктивної самооцінки в професійній діяльності; володінням способами самодіагностики і саморозвитку [5, 11].

В. П. Беспалько пропонує впорядковану структуру, що відображає процес оволодіння певною діяльністю, і також виділяє чотири рівні засвоєння: ознайомлення, репродукції, умінь і навичок, трансформації. Кожному рівню відповідають певні дії, які повинен виконувати студент: 1) розпізнання, розрізнення; 2) відтворення, порівняння й аналіз; 3) використання на практиці до заданого класу явищ і об'єктів для завдань, що потребують буквального застосування знань і дій (дія за прикладом); 4) використання засвоєної інформації для практичних завдань поза тим класом явищ і об'єктів, на яких формувалися знання (дія з широким перенесенням) [1, 46].

С. О. Дружилов у загальному випадку в моделі навчання виділяє чотири стадії сформованості компетентності, що характеризують процес навчання, починаючи від стадії початкового знайомства з новим матеріалом (знаннями, концепціями, навичками) і закінчуючи стадією сформованої компетентності [3, 33].

Перша стадія – несвідома некомпетентність. У людини немає необ-

хідних знань, умінь, навичок, і вона не знає про їх відсутність або взагалі про можливі вимоги щодо них для успішної діяльності. Ця стадія характеризується наступною професійною самооцінкою: «Я не знаю, що я не знаю». Коли людина усвідомлює відсутність знань, умінь, навичок, необхідних для даної діяльності, вона переходить на другу стадію.

Друга стадія – свідомо некомпетентність. Людина усвідомлює, що їй не вистачає професійних знань, умінь, навичок. Тут можливі два результати а) конструктивний (як форма прояву особистісної та професійної активності) і б) деструктивний (форма соціальної пасивності). Конструктивний шлях означає, що усвідомлення суб'єктом своєї професійної некомпетентності сприяє підвищенню його мотивації на здобуття відсутніх професійних знань, умінь, навичок. Деструктивний результат може призводити до виникнення почуття невпевненості у своїх силах, психологічного дискомфорту, підвищеної тривожності та ін., що заважає подальшому професійному навчанню. Для другої стадії характерна наступна професійна рефлексія суб'єкта «Я знаю, що я не знаю».

Третя стадія – свідомо компетентність. Людина знає, що входить в структуру і становить зміст її професійних знань, умінь і навичок і може їх ефективно застосовувати. Для третьої стадії характерна професійна самооцінка суб'єкта в такій формі: «Я знаю, що я знаю».

Четверта стадія – несвідомо компетентність. Професійні навички повністю інтегровані, вбудовані в поведінку; професіоналізм є частиною особистості. Несвідомо компетентність характеризує рівень майстерності [3, 33].

Таким чином, дослідники виділяють, як правило, 4 рівні сформованості компетентності: низький, середній, достатній та високий. У дослідженні ми розглядатимемо формування компетентності в програмуванні як перехід між цими рівнями: низький → середній → достатній → високий.

Низький рівень характеризується негативним ставленням до процесу розробки, описання, налагодження, тестування та аналізу результатів роботи програм; поверхневими, несистемними знаннями з програмування, відсутністю вмінь. *Середній рівень* відзначається індивідуальним ставленням до процесу розробки, описання, налагодження, тестування та аналізу результатів роботи програм; слабкою мотивацією до опанування програмуванням; середніми за об'ємом, фрагментарними знаннями, наявністю окремих, розрізнених вмінь. *Достатній рівень* передбачає виявлення інтересу до процесу розробки, описання, налагодження, тестування та аналізу результатів роботи програм; упорядкованими, структурованими знаннями, достатніми вміннями; проявленням здатності до співпраці у процесі програмування, використанням засобів для організації

спільної роботи над проектом; здатністю до самонавчання. *Високий рівень* характеризується позитивним ставленням до процесу розробки, описання, налагодження, тестування та аналізу результатів роботи програм; стійкими, ґрунтовними знаннями, здатністю до самонавчання, творчим підходом, вміннями до нестандартного розв'язання завдань, умінням відстоювати власну думку, постійною здатністю до співпраці у процесі програмування та використанням засобів для організації спільної роботи.

Література:

1. Беспалько В. П. Элементы теории управления процессом обучения. Часть I. (Описание целей и способы их достижения в обучении) (Материалы лекций, прочитанных в Политехническом музее на факультете программированного обучения) / В. П. Беспалько. – М. : Знание, 1970. – 79 с.

2. Головань М. С. Модель процесу розвитку інформатичної компетентності студентів економічного вузу / М. С. Головань // Збірник наукових праць Кам'янець-Подільського національного університету. Серія педагогічна / Кам'янець-Подільський національний ун-т. – Кам'янець-Подільський, 2008. – Вип. 14. – С. 17–20.

3. Дружилов С. А. Обучение и стадии профессиональной компетентности / С. А. Дружилов // Непрерывное образование как условие развития творческой личности : сб. по материалам Фестиваля педагогического творчества, 28–29 авг. 2000 г. – Новокузнецк : ИПК, 2001. – С. 32–33.

4. Рафальська М. В. Формування інформатичних компетентностей майбутніх вчителів інформатики у процесі навчання методів обчислень : дис. ... канд. пед. наук : 13.00.02 – теорія і методика навчання (інформатика) / Рафальська Марина Володимирівна ; Національний педагогічний ун-т ім. М. П. Драгоманова. – К., 2010. – 225 с.

5. Сікора Я. Б. Формування професійної компетентності майбутнього вчителя інформатики засобами моделювання : автореф. дис. ... канд. пед. наук : 13.00.04 – теорія і методика професійної освіти / Сікора Ярослава Богданівна ; Житомирський державний ун-т ім. Івана Франка. – Житомир, 2010. – 21 с.

ОСОБЛИВОСТІ ОРГАНІЗАЦІЇ САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ ПРИ ВИВЧЕННІ КУРСУ «ПОДІЄ-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

М. В. Моїсеєнко, Н. В. Моїсеєнко

Україна, м. Кривий Ріг, Криворізький національний університет

Вища професійна освіта України останнім часом зазнала значних змін, що наближають її до світових тенденцій у галузі освітніх послуг. З орієнтацією освіти України на Болонський процес до підготовки спеціалістів висуваються нові вимоги, зокрема збільшення частки самостійної роботи студентів в навчальному процесі. Це обумовило посилену увагу науковців до цього питання та збільшення кількості відповідних досліджень та публікацій [1].

Дисципліна «Подіє-орієнтоване програмування» в Криворізькому педагогічному інституті ДВНЗ «Криворізький національний університет» викладається для студентів II-III курсу спеціальності «Інформатика». Даний курс спрямований на формування у студентів знань, вмій та навичок, необхідних для ефективного використання засобів сучасних інформаційних технологій у своїй майбутній професійній діяльності.

Курс продовжує дисципліну «Програмування» в напрямку поглиблення навичок в розробці програмного забезпечення методів розв'язання відносно складних задач в реальному масштабі часу, що вимагають побудови програмного продукту по принципу реакції на події, що відбуваються в багатозадачному операційному середовищі.

Крім того, курс включає методи програмування графічного інтерфейсу користувача на базі можливостей, які дає операційна система.

Після вивчення курсу студенти повинні знати:

- особливості програмування в багатозадачних операційних системах;
- формування структури програм по реактивному принципу;
- методи обробки подій та повідомлень;
- методи створення зручного для користувача інтерфейсу в графічному режимі;
- методи організації діалогів програми з користувачем.

В процесі вивчення дисципліни студенти повинні навчитися розробляти та реалізовувати проекти відносно складних програмних продуктів як взаємодію об'єктів, що реагують на події та повідомлення операційної системи, програми, інших об'єктів.

Курс складається з 54 годин лекцій, 110 годин лабораторних робіт та 160 годин самостійної роботи.

Пропонуємо розподіл годин за модулями та видами занять (табл. 1).

Таблиця 1

Структура залікового модуля

Модуль	Кількість годин, відведених на:		
	лекції	лабораторні заняття	самостійну роботу
1. Вступ до подіє-орієнтованого програмування	2	2	8
2. Створення вікон за допомогою Windows Forms	6	6	12
3. Використання елементів керування Windows Forms	10	26	34
4. Розширені функціональні можливості Windows Forms	14	24	32
5. Звичайні діалогові вікна	10	24	34
6. Програмування за допомогою класів GDI+	12	28	40
Усього годин	54	110	160
	324		

Одним з підходів до сучасної організації навчального процесу в вищій школі є створення спеціального освітнього середовища. Для підготовки майбутніх інженерів-програмістів доцільно створити інформаційно-предметне середовище, що являє собою сукупність педагогічних, інформаційно-комунікативних, матеріально-технічних умов, які необхідні для організації самостійної роботи студентів, для формування професійних знань та вмінь у виділеній предметній області в процесі розв'язання професійно-орієнтованих задач.

В складі інформаційно-предметного середовища підготовки майбутніх інженерів-програмістів виокремлюють такі педагогічні умови:

- орієнтація студентів на самостійну, пошукову, дослідницьку діяльність, з використанням різноманітних електронних та друкованих носіїв для підбору та аналізу матеріалу в виділеній предметній області;
- орієнтація студентів на професійну діяльність через включення в навчальний процес професійно-орієнтованих задач;
- наявність сукупності навчально-методичних матеріалів для організації самостійної роботи студентів при вивченні ними одного з курсів програмування.

Оскільки дві перші вищезазначені умови в певній мірі залежать від третьої, розробка навчально-методичних матеріалів та способів організації навчального процесу є основою розробки виділених педагогічних

умов.

Під час вивчення курсу подіє-орієнтованого програмування виникають проблеми, притаманні більшості курсів програмування. Більш підготовані студенти швидко втрачають інтерес до занять, оскільки можуть справитися з задачею вивчення нової мови програмування самостійно. Для цієї групи студентів більш важливою є сфера її застосування, її можливості для розв'язання конкретної задачі.

Такий підхід до організації вивчення прикладних дисциплін не стимулює студентів до самостійного вибору та вивченню програмних засобів для розв'язання прикладної задачі. Переважна більшість студентів-випускників мають навички роботи тільки з тими програмними засобами, мовами програмування, які вивчалися під час занять. Саме тут постає проблема невідповідності підготовки спеціаліста з вимогами ринку праці, вимоги якого ростуть в дуже швидкому темпі.

Виходячи з цього, головна задача викладача, що веде подібні прикладні дисципліни, викласти методологію вивчення мови програмування та способи її використання, а потім надати студенту можливість самостійного вибору та самостійного вивчення програмних засобів в процесі розв'язання прикладної задачі.

Таким чином, однією з вимог до змісту інформаційно-предметного середовища підготовки майбутніх інженерів-програмістів є необхідність містити в собі весь потрібний матеріал для організації самостійної роботи студентів по вибору засобів реалізації та розв'язання підсумкового завдання.

Ще одна проблема, що виникає під час навчання студентів прикладної дисципліни, полягає в тому, що студенти розв'язують задачі мало пов'язані з практичною діяльністю, що значно зменшує зацікавленість студентів у виконуваній ними роботі. Це приводить до того, що якість виконання робіт знижується, оскільки головний пріоритет має отримання оцінки, а не якісне розв'язання запропонованої задачі.

Враховуючи всі вищезазначені проблеми, організація навчального процесу для вивчення дисципліни «Подіє-орієнтоване програмування» може мати приблизно такий вигляд.

Лекції. Основним напрямком цього виду занять є засвоєння студентами теоретичних положень та понять, на яких базується дана прикладна дисципліна [2-4]. Даний матеріал також розміщується в електронному навчальному комплексі «Подіє-орієнтоване програмування».

Лабораторні роботи. Ціль проведення цього виду занять полягає в отриманні студентами вмінь та навичок розробки програмних додатків. Студентам надається можливість під керівництвом викладача засвоїти принципи вивчення мови програмування.

Весь необхідний навчальний та контрольний матеріал, включаючи постановки задач на лабораторні роботи розміщені в електронному навчальному комплексі.

Самостійна робота. Результатом самостійної роботи студента повинно стати розв'язання прикладної задачі – проектування та розробка програмного додатку за індивідуальним завданням. Особливо слід зупинитися на процесі постановки завдання на підсумкову роботу. В якості самостійної роботи треба поставити перед студентом таку задачу, яка вимагала б від студента обґрунтованого вибору засобів реалізації, самостійного їх вивчення та наступного розв'язання задачі з урахуванням знань, умінь та навичок отриманих на лекційних та лабораторних заняттях. Окрім задач, обраних викладачем, студентам надається можливість самостійно обрати для себе задачу та розв'язати її. Як показує досвід, група студентів з рівнем підготовки вище середнього легко з цим справляється, оскільки багато студентів старших курсів вже мають досвід роботи на підприємствах і обирають задачі, безпосередньо пов'язані з їх місцем роботи, професійною діяльністю батьків, родичів або їх особистими інтересами.

В якості основного засобу організації самостійної роботи використовується робота студентів за допомогою електронного навчального комплексу «Подіє-орієнтоване програмування», в якому також розміщена методика розв'язування підсумкової задачі.

Пропонований підхід до організації самостійної навчальної роботи студентів ґрунтується на використанні структурно-ієрархічної моделі дидактичного наповнення електронного навчального комплексу [5].

Модель передбачає дворівневу методичну базу знань. Перший рівень включає створення єдиного комплексу електронних навчальних енциклопедій за різними галузями знань. Другий рівень передбачає створення електронного навчального комплексу на ґрунті розроблених енциклопедій шляхом організації їх в ієрархічну структуру, що адекватно реалізують робочі навчальні програми у відповідності з діючими галузевими стандартами вищої освіти для напрямку.

В рамках моделі енциклопедія описує деяку предметну область, а навчальний матеріал представляється у вигляді окремих модулів. Кожен модуль належить якійсь енциклопедії та є дидактично завершеним описом деякого терміну (поняття).

Кожен модуль енциклопедії містить набір іменованих дидактичних компонент, завдяки чому модель підтримує структурування модуля за способами представлення змісту навчання. Таким чином, один і той самий матеріал представляється декількома способами, що називаються компонентними типами, кожен з яких має свої дидактичні можливості.

Кожен компонентний тип визначається шляхом визначення його інтерфейсу: набору атрибутів і набору операцій над атрибутами.

Кожен атрибут компонентного типу має структурний (об'єктний) тип. Припустима множина об'єктних типів моделі розширює стандарт SCORM [6]. В якості атрибута може виступати файл з зображенням у різних графічних форматах (JPEG, GIF та т. ін.), текстовий файл, файл презентації, XML-документ. Для будь-якого атрибуту визначені дві стандартні операції: «присвоїти значення» та «визначити значення».

Архітектурний принцип побудови електронних навчальних енциклопедій передбачає наявність схеми енциклопедії, яка визначає однако-ву внутрішню структуру для всіх модулів енциклопедії. При використанні відповідного електронного навчального комплексу в навчальному процесі такий підхід дозволяє виділяти спосіб представлення матеріалу в окремі самостійні методичні об'єкти. Наприклад, список літератури, конспект лекцій, вказівки до виконання лабораторних робіт, контрольні питання тощо.

Схеми даних енциклопедій містять однаковий набір компонентних типів:

- теоретичний опис поняття ;
- приклади;
- презентація;
- список вправ для самостійного виконання;
- тест для самоперевірки (доступний для студента список питань);
- контрольний тест (закритий для студента список питань для контролю засвоєння навчального матеріалу);
- список літератури по темі.

Висновки

В даній роботі представлений підхід до організації самостійної роботи студентів із вивчення подіє-орієнтованого програмування на основі електронних навчальних енциклопедій «Мова програмування C#» та «Подіє-орієнтоване програмування».

Дані енциклопедії являють собою набір модулів, кожен з яких присвячений одному поняттю. Модулі мають однакову структуру та включають теоретичний матеріал, приклади, презентації, тести та інші дидактичні компоненти. Електронний навчальний комплекс, що застосовується для самостійної роботи студентів, будується на основі однієї або декількох енциклопедій.

Отримані результати впроваджуються в навчальний процес підготовки бакалаврів за напрямком 6.040302 – Інформатика*.

Література

1. Белогурова А. В. Концепция создания электронных учебников по математическим дисциплинам для технических вузов Украины / Белогурова А. В., Кузнецов А. И., Самойленко Н. И. // Коммунальное хозяйство городов. – 2005. – Вып. 63. – С. 344–352.
2. Культин Н. Б. Microsoft Visual C# в задачах и примерах / Культин Н. Б. – СПб. : БХВ-Петербург, 2009. – 320 с.
3. Троелсен Э. Язык программирования C# 2005 и платформа .NET 2.0 / Э. Троелсен . – М. : Вильямс, 2007. – 1154 с.
4. Visual C# 2008: базовый курс / Уотсон К., Нейгел К., Петерсен Я., Хаммер Р., Джон Д., Скиннер М., Уайт Э. – М. : Вильямс, 2009. – 1216 с.
5. Жигальская Н. С. Стандартизация содержания электронных учебных курсов и энциклопедий на основе структурно-иерархического подхода / Жигальская Н. С., Соколинский Л. Б. // Новые информационные технологии в образовании : материалы междунар. науч.-практ. конф., Екатеринбург, 26-28 февраля 2008 г. : в 2 ч. – Екатеринбург : Изд-во Рос. гос. проф.-пед. ун-та, 2008. – Ч. 1. – С. 84–89.
6. Advanced Distributed Learning. Sharable Content Object Reference Model (SCORM) 2004 / Перевод с англ. Е. В. Кузьминой. – М. : Информика, 2005. – 29 с.

ПРО МЕТОДИ НАБЛИЖЕННЯ ТАБЛИЧНИХ (ДОСЛІДНИХ) ДАНИХ ЗАСОБАМИ VBA MS EXCEL

Б. Р. Монцібович¹, О. Б. Хапко²

¹ Україна, м. Львів, Львівський навчально-науковий інститут
Закарпатського державного університету

² Україна, м. Львів, Львівська філія Дніпропетровського національного
університету залізничного транспорту ім. акад. В. Лазаряна
mon_ua@yahoo.com

У курсах дисциплін «Числові методи», «Додаткові розділи чисельного аналізу», «Математичне прогнозування», «Математична статистика», «Графічне та геометричне моделювання», «Моделювання систем», які вивчаються студентами напряму «Комп'ютерні науки», важливе місце займає задача апроксимації таблично-заданих функцій простими аналітичними виразами з невеликим числом невідомих параметрів.

Такі задачі часто виникають у практиці науково-інженерних розрахунків, у таких, наприклад, дослідженнях:

– проектування контрольно-вимірювальних приладів, зокрема, термометрів, пірометрів, вологомірів, манометрів і ін. для градування їх статичних характеристик;

– проектування різноманітних функціональних перетворювачів для опису передавальних функцій;

– реалізація алгоритмів обчислення спеціальних функцій на високопродуктивних процесорах для апроксимації складних функціональних залежностей аналітичними виразами, зручними для обчислень.

Загальне формулювання задачі наближення дослідних даних.

Нехай x_i, y_i – відомі дослідні дані ($i=1, n$). Необхідно наблизити ці дані виразом $F(A, x) \equiv F(a_0, a_1, \dots, a_m, x)$, $m < n$.

Вважається, що вид $F(A, x)$ вибраний дослідником з певних міркувань. Необхідно визначити коефіцієнти a_0, a_1, a_m так, щоб значення y_i та $\tilde{y}_i = F(A, x_i)$ були достатньо близькі.

Розрізняють два критерії наближення даних:

1) **середньоквадратичний** (за методом найменших квадратів – МНК)

$$\sum_{i=1}^n (y_i - \tilde{y}_i)^2 \equiv \sum_{i=1}^n (y_i - F(A; x_i))^2 \Rightarrow \min$$

Середньоквадратичний критерій застосовується тоді, коли є підозра, що серед дослідних даних є дані, що містять випадкові помилкові дані, тому що метод найменших квадратів ігнорує випадкові викиди.

Критерієм якості наближення за методом найменших квадратів є середньоквадратична похибка

$$\Delta_{\text{скв}} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_i - \tilde{y}_i)^2};$$

2) **мінімаксий критерій** (за методом рівномірного наближення) – за цим критерієм мінімізується максимальна похибка наближення

$$\max_{1 \leq i \leq n} |y_i - \tilde{y}_i| \Rightarrow \min.$$

Мінімаксий критерій доцільно застосовувати тоді, коли усі дослідні дані задані з однаковою похибкою і в однаковій мірі заслуговують довіри, серед них немає помилкових – це зокрема табличні дані, дані різноманітних вимірювань за допомогою вимірювальних приладів, такі дані містять обмежену похибку, зумовлену точністю вимірювального приладу [1; 2].

Порівняння критеріїв та їх застосування на практиці.

В наш час зазвичай використовується метод найменших квадратів, оскільки він більш відомий і доступний, зокрема реалізований для часткових випадків в MS Excel. Проте рівномірне наближення $P_m(x)$ має ту перевагу, що графік $P_m(x)$ лежить в криволінійній смугі шириною $2\Delta_0$, обмеженої графіками функцій $y(x)+\Delta_0$ і $y(x)-\Delta_0$.

Теперішній стан прикладного програмного забезпечення для знаходження найкращого рівномірного наближення функцій не відповідає практичним потребам. У свій час для машин серії МИР [3; 4] і СМ-4 [5; 6] були розроблені пакети програм для розв'язування деяких задач з рівномірної апроксимації функцій. Проте, сучасні інтегровані пакети, такі як Mathematica, Mathcad чи Matlab включають, на жаль, процедури лише для розв'язування окремих задач із застосуванням методу найменших квадратів. Математична система Maple передбачає визначення рівномірного наближення лише аналітично заданих функцій поліномом і раціональним виразом, хоча актуальність апроксимації таблично заданих функцій загальновідома. У Центрі математичного моделювання НАН України (м. Львів) протягом останніх років розробляється і постійно вдосконалюється пакет РАДАН – пакет програм для рівномірного наближення таблично-заданих функцій (дослідних даних) [7; 8]. Але як і всі пакети, що є корпоративною власністю, він важко доступний для окремих дослідників та студентів.

У Львівському навчально-науковому інституті Закарпатського державного університету розробляються програми для рівномірного наближення дослідних даних простими аналітичними виразами засобами VBA MS Excel та їх порівняння з наближеннями за методом найменших квадратів. Ці засоби є доступними і добре знайомими студентам, їх за-

стосування передбачає більше самостійної і творчої роботи, ніж використання готового пакету [9; 10].

Відмінність середньоквадратичного наближення від рівномірного показано на рис. 1. Із цього рисунку видно, що Y_{skv} не забезпечує досягнення найменшої абсолютної похибки на відрізку апроксимації. Наступний рис. 2 демонструє різницю між методом найменших квадратів та рівномірним наближенням для наближення прямою $\hat{y}(x)=Ax+B$ у трьох точках (x_i, y_i) .

Різниця між Y_{rivn} і Y_{skv} - за межі $[y(x)-d, y(x)+d]$ виходить Y_{skv}

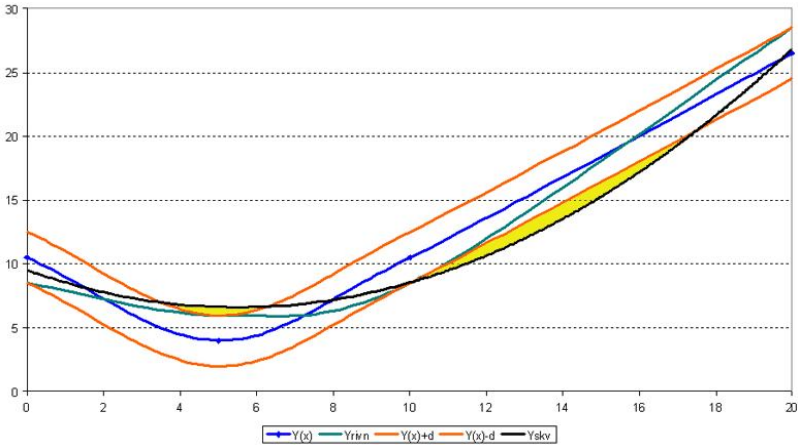


Рис. 1. Графік мінімаксної та середньоквадратичної апроксимацій

З цього прикладу видно, що для трьох заданих точок з координатами $(0;0)$, $(4;8)$, $(8;1)$ отримано два наближення прямою $y=Ax+B$: МНК – $y=0.125x+2.5$, рівномірне – $y=0.125x+3.75$.

Для методу найменших квадратів максимальна похибка наближення дорівнює 5.00, а для рівномірного – 3.75, тобто в 1.3 рази менша; рівномірне наближення проводить пряму на однаковій відстані від заданих точок, а МНК – ближче до двох крайніх точок, а середню ніби ігнорує (при цьому сума квадратів відхилень є найменшою).

У додатку до доповіді подано файл RizMNRivn3.xls, що містить програму, за допомогою якої були отримані наведені вище таблиця і графік. Цією програмою можна міняти розташування заданих трьох точок і отримувати відповідно різні результати наближення. Фактично для отримання різних варіантів наближення достатньо міняти тільки значення x_2 , тобто значення абсциси другої точки – вона може набувати значень від 1 до 7.

На рис. 3 подано результати порівняння середньоквадратичного та

рівномірного наближень з відносною похибкою виразом $Y=Ax^B$ реальних табличних даних, а саме – залежність густини газу (Y , кг/м³) від тиску (X , МПа) для газу неон при температурі $T=500$ °К. Джерело цих даних зазначене на рисунку. Принагідно зауважимо, що задача аналітичного подання важливих залежностей між фізичними величинами є актуальною – адже наявність достатньо точної апроксимаційної формули дає можливість обчислювати і використовувати ці залежності у комп’ютерному опрацюванні даних, не звертаючись до друкованих довідників – сьогодні значення таких математичних функцій, як $\sin(x)$, $\cos(x)$, $\ln(x)$ тощо ніхто не читає з таблиць Брадїса, а отримують, натискаючи кнопку на калькуляторі чи викликаючи потрібну функцію у програмі на комп’ютері (де звичайно «зашиита» апроксимаційна формула).

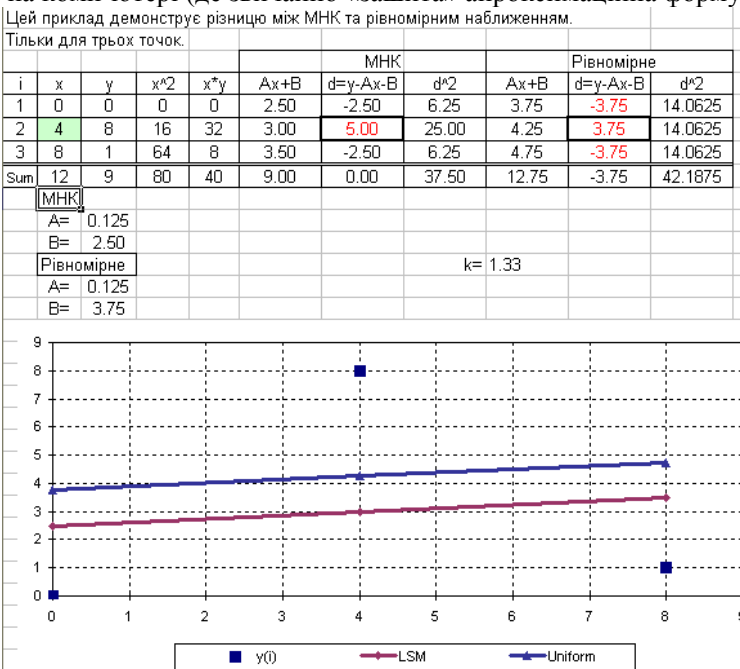


Рис. 2. Різниця між МНК та рівномірним наближенням 3 точок

Знаходження рівномірного наближення, яке, як відомо, забезпечує найменшу абсолютну чи відносну похибку наближення, є задачею значно складнішою, ніж знаходження наближення МНК – тут застосовується ітераційний метод Ремеза відшукання так званих точок альтернансу (точок у яких досягається максимальне значення похибки) – його в MS Excel можна реалізувати тільки у вигляді модуля VBA. Нижче подано текст модуля.

Наближення з найменшого відносно похибкою степеневою функцією $A \cdot x^B$ на проміжку $[X(n0), X(n)]$											
R= 10		n0= 1				n= 10				Виконати	
x	Y	t=ln(x)	u=ln(Y)	x^2	xY	MНК				Rivnomirne	
						Ax^B	d=1-Ax^B/y	d^2	Ax^B	d=1-Ax^B/y	d^2
1	4.84	0	1.5769	0.000	0.000	4.882	-0.06	0.7457	4.876	-0.74	0.5473
2	9.65	0.6931	2.267	0.480	1.571	9.651	-0.01	0.0001	9.635	0.16	0.0243
3	14.42	1.0986	2.6686	1.207	2.932	14.378	0.29	0.0830	14.351	0.48	0.2289
4	19.17	1.3863	2.9533	1.922	4.094	19.079	0.47	0.2246	19.039	0.68	0.4646
5	23.88	1.6094	3.173	2.590	5.107	23.760	0.50	0.2525	23.707	0.72	0.5240
6	28.57	1.7918	3.3524	3.210	6.007	28.425	0.51	0.2572	28.359	0.74	0.5473
8	37.84	2.0794	3.6334	4.324	7.555	37.718	0.32	0.1039	37.623	0.57	0.3285
10	47.00	2.3026	3.8501	5.302	8.865	46.972	0.06	0.0036	46.847	0.33	0.1060
12	56.05	2.4849	4.0262	6.175	10.005	56.194	-0.26	0.0662	56.039	0.02	0.0004
16	73.80	2.7726	4.3014	7.687	11.926	74.566	-1.04	1.0763	74.346	-0.74	0.5473
67	315.22	16.2188	31.8024	32.898	58.062	315.625	-0.01	2.8131	314.822	2.22	3.3185
MНК:		b1= 0.9833	B= 0.9833								
		b0= 1.59	A= 4.6818								
Rivnomirne:								-1.04	=	1.40	
								-0.74			
NEON, T=500 K		Джерело даних:		Термофізическі свойства технічески важных газов при высоких температурах							
x - тиск газу, Мпа				и давлениях. Справочник. - Москва, Энергоиздат, 1989. - 232 с.							
Y - густина газу, кг/м³				(стр. 74)							

Рис. 3. Порівняння МНК і рівномірного наближень певних реальних даних

```

Sub ST1VP() 'Ax^B з відносною похибкою у %
Dim n0, n, r, k, z, l, i, j As Integer
Dim A, B, p, g, ya, a0, c, yc, d, e, da, gz As Single
Dim X(100), Y(100), T(100), U(100), Yn(100), En(100) As Single

| n0 = Range("F3"): n = Range("H3")
For i = n0 To n
X(i) = Cells(6 + i, 1): Y(i) = Cells(6 + i, 2)
T(i) = Cells(6 + i, 3): U(i) = Cells(6 + i, 4)
Next i
k = n0: l = n: z = Int((l - k) / 2) + k
2: ya = Y(k): a0 = X(k): B = (U(l) - U(k)) / (T(l) - T(k))
3: c = X(z): yc = Y(z): A = 2 * yc * ya / (ya * c ^ B + yc * a0 ^ B)
d = (yc - A * c ^ B) / yc: da = Abs(d): g = 0

For i = n0 To n
Yn(i) = A * X(i) ^ B: En(i) = (Y(i) - Yn(i)) / Y(i): e = Abs(En(i))
If e > g Then g = e: gz = En(i): j = i 'end if
Next i
If g - da < 0.003 * g Then GoTo 6
If (j - k) * (j - z) * (j - l) = 0 Then GoTo 6
If d * gz > 0 Then
If j > l Then k = z: z = l: l = j: GoTo 2
If j < k Then l = z: z = k: k = j: GoTo 3
z = j: GoTo 3
Else
If j < z Then
k = j: GoTo 2
Else
l = j: GoTo 2
End If

```

```

End If
6: Range("E22") = B: Range("E23") = A
For i = n0 To n
    Cells(6 + i, 10) = Yn(i): Cells(6 + i, 11) = En(i) * 100
Next i
End Sub

```

Програма для знаходження наближення вигляду $Y=Ax^B$ з найменшою відносною похибкою та для порівняння з МНК, оформлена у вигляді файлу ST1VP_MNK_RIVN.xls, додається.

Література

1. Коллатс Л. Теория приближений. Чебышевские приближения и их приложения / Коллатс Л., Крабс В. – М. : Наука, 1978. – 272 с.
2. Ремез Е. Я. Основы численных методов чебышевского приближения / Е. Я. Ремез. – К. : Наукова думка, 1969. – 623 с.
3. Монцибович Б. Р. Наилучшие приближения табличных функций (алгоритмы и программы). В 2-х частях / Монцибович Б. Р., Попов Б. А. – К. : Институт кибернетики АН УССР, 1973. – 214+238 с.
4. Попов Б. О. Розв'язування задач на машинах для інженерних розрахунків / Попов Б. О., Монцібович Б. Р. – К. : Наукова думка, 1978. – 346 с.
5. Диалоговый пакет программ для равномерной аппроксимации табличных данных на языке Бейсик для СМ-4 (ППП РАДАН) / Монцибович Б. Р., Малачивский П. С., Тькайло С. В. и др. – Львов : Ин-т прикл. пробл. мех. и мат. АН УССР, 1985. – 383 с. – Деп. в Укр.РФАП 25.02.1986, № АП0091.
6. Диалоговый пакет программ для аналитической обработки табличных данных (ППП РАДАН-2) / Монцибович Б. Р., Криворучко Г. Ф., Малачивский П. С. и др. – Львов, 1988. – 135 с. – Деп. в Укр.РФАП 27.03.1990, № АП0278.
7. Малачивский П. С. Алгоритмы и программное обеспечение для равномерной аппроксимации экспериментальных данных / Малачивский П. С., Монцибович Б. Р. // Электронное моделирование. – 2011. – Том 33, №5. – С. 97-106.
8. Яцук В. О. Методи підвищення точності вимірювань : підручник / Яцук В. О., Малачівський П. С. – Львів : Бескид-Біт, 2008. – 368 с.
9. Основы алгоритмізації та програмування: середовище VBA : навч. посібник / Делявський М. В., Жмуркевич А. Є., Одрехівський М. В., Чаповська Р. Б. – Чернівці : Книги-XXI, 2006. – 430 с.
10. Створення програмних додатків в середовищі MS Office Visual Basic for Application (для самостійного вивчення) : навчальний посібник / Укладачі І. А. Григоришин, С. В. Косяченко, Л. В. Кулібаба. – К. : Дакор, КНТ, 2007. – 208 с.

СОПОСТАВЛЕНИЕ ПРОИЗВОДИТЕЛЬНОСТЕЙ GPU И CPU ДЛЯ МАТРИЧНОГО УМНОЖЕНИЯ С ДВОЙНОЙ ТОЧНОСТЬЮ

А. А. Мясищев

Украина, г. Хмельницкий, Хмельницкий национальный университет
alex@tup.km.ua

Матричные вычисления составляют основу многих научных и инженерных расчетов. Например, для решения задач расчета на прочность, исследования процессов пластического течения материалов часто используется метод конечных элементов, который сводится к решению систем линейных уравнений и матричным вычислениям. В связи с этим многие стандартные библиотеки программ содержат процедуры для различных матричных операций. Вследствие своей вычислительной трудоемкости и возможности эффективного распараллеливания, матричные вычисления представляют собой классическую область для параллельных вычислений.

Сопоставим эффективность использования для расчета произведения квадратных матриц для **двойной точности** вычислительную систему, в которой установлены 6-тиядерный процессор AMD Phenom II X6 1090T (CPU) с видеокартой NVIDIA GeForce GTX 480 и вычислительную систему с таким же процессором, но с графическим процессором (GPU) NVIDIA Tesla C2075. Расчет будет выполняться для 3-х случаев. В первом случае используется только процессор (CPU), для которого будет выполняться распараллеливание по 6-ядрам с использованием библиотек ScaLAPACK и библиотек ATLAS (автоматически настраиваемое программное обеспечение для решения задач линейной алгебры). Библиотеки ATLAS при компиляции настраиваются под конкретную архитектуру процессора вычислительной системы [1,2]. Во втором случае расчет будет выполняться на видеокарте NVIDIA GeForce GTX 480 с использованием технологии CUDA [1]. В третьем – на графическом процессоре NVIDIA Tesla C2075. Вычислительная система настроена на работу с 32-х разрядной операционной системой Linux Ubuntu ver. 10.10 desktop. В системе установлены компилятор FORTRAN F77, библиотеки MPI [3], ScaLAPACK [1] и ATLAS [2] для 6-тиядерного процессора. Для программирования на NVIDIA инсталлируются видеодрайвер nvidia и программное обеспечение с сайта <http://developer.nvidia.com/cuda-toolkit-archive>. Последовательность установки библиотек MPI, ScaLAPACK, ATLAS под ОС Linux Ubuntu ver. 9.04 desktop подробно представлена в работе [4] для системы с 4-х ядерным процессором CORE 2 QUAD PENTIUM Q6600 2.4GHZ, поэтому здесь не рассматривается. Рассмотр-

рим более подробно установку программного обеспечения для NVIDIA CUDA.

1. Копируем видеодрайвер NVIDIA с сайта <http://www.nvidia.ru/Download/index.aspx?lang=ru> для 32-х разрядной ОС Linux.

2. Открываем файл `/etc/modprobe.d/blacklist.conf` и добавляем в него модули свободных драйверов `vga16fb`, `nouveau`, `riva16b`, `nvidiafb`, `rivatv`.

3. Устанавливаем необходимые пакеты

```
sudo apt-get install linux-headers-`uname -r` binutils pkg-config build-essential xserver-xorg-dev
```

4. Далее переходим в параллельно работающую консоль и вводим свои данные (логин и пароль). Остановим X-сервер командой:

```
sudo /etc/init.d/gdm stop
```

и запускаем установку драйвера

```
sudo sh ./NVIDIA-Linux-x86-290.10.run
```

После ответа на ряд вопросов драйвер должен установиться. Далее запускаем графический сервер

```
sudo service gdm start && exit
```

Выполняем команду

```
sudo apt-get --purge remove nvidia-*
```

и перезагружаем компьютер

```
reboot
```

5. С сайта <http://developer.nvidia.com/cuda-toolkit-40> копируем для Linux Ubuntu 10.10, 32-и разрядной версии CUDA Toolkit и GPU Computing SDK. Устанавливаем их командами:

```
sudo sh ./cudatoolkit_4.0.17_linux_32_ubuntu10.10.ru
```

```
sudo sh ./gpubcomputingsdk_4.0.17_linux.run
```

6. Выполняем настройку системы

```
export PATH=/usr/local/cuda/bin:$PATH
```

```
export LD_LIBRARY_PATH=/usr/local/cuda/lib:$LD_LIBRARY_PATH
```

```
echo 'export PATH=/usr/local/cuda/bin:$PATH' >> ~/.bashrc &&
```

```
echo 'export LD_LIBRARY_PATH=
```

```
/usr/local/cuda/lib:$LD_LIBRARY_PATH' >> ~/.bashrc
```

и устанавливаем дополнительные пакеты

```
sudo apt-get install g++ freeglut3-dev libxi-dev libxmu-dev
```

7. Для правильной компиляции всех примеров может возникнуть необходимость выполнения команды (или подобных)

```
sudo ln -s /usr/lib/libGL.so.1 /lib/libGL.so
```

8. Выполняем компиляцию примеров

```
cd ~/NVIDIA_GPU_Computing_SDK/C && make
```

9. Для тестирования правильности установки запускаем один из примеров

```
cd ~/NVIDIA_GPU_Computing_SDK/C/bin/linux/release&&./deviceQuery
```

На консоли должны появиться основные возможности GPU.

Рассмотрим программы расчета произведения квадратных запол-

ненных матриц для NVIDIA GPU. В первом случае для простоты прием, что элементы матриц хранятся в глобальной памяти GPU. Для составления даже этой простой программы необходимо подробнее рассмотреть программную модель GPU для компилятора Си [5]. Верхний уровень ядра, называемый сеткой (grid), состоит из блоков. В свою очередь блоки представляют собой либо одномерную, либо двухмерную сеть нитей. Вышесказанное может быть проиллюстрировано рис. 1. Номер нити в блоке или номер блока в grid синтаксически специфицируется оператором <<<...>>> и может быть переменной типа int или dim3.

Число нитей, входящих в блок, определяется встроенной переменной blockDim. Индекс нити внутри блока определяется переменной threadIdx, а индекс блока внутри grid – переменной blockIdx. Нити в блоке являются непосредственными исполнителями вычислений. Нить является 3-х компонентным вектором, т.е. может идентифицироваться, используя одномерный, двухмерный и трехмерный индекс нити, образуя в свою очередь одномерный, двух размерный и трех размерный блок нитей. Существует ограничение количества нитей на один блок, которое не может превышать 1024 нити.

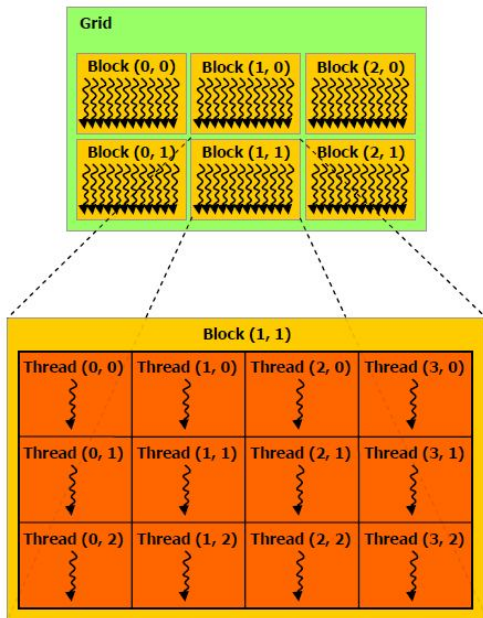


Рис. 1

Расширение Си CUDA позволяет вызвать функцию, называемую ядром так, что она будет параллельно выполнять N разных нитей CUDA. Такие функция декларируется спецификатором __global__. В качестве иллюстрации ниже рассмотрен пример распечатки функцией ядра трех раз слова hello=n (n-номер нити), т.к. в функции hello<<<1, 3>>>(); указан запрос на работу одного блока и трех нитей.

```
#include <stdio.h>
__global__ void hello() {
    int i=threadIdx.x; printf("Hello=%d\n",i);
}
__host__ int main() { hello<<<1, 3>>>(); cudaThreadExit(); }
```

Компилируем командой [5]

```
nvcc -arch=sm_20 aal.cu -o aal
```

Запускаем

```
./aal  
Hello=0  
Hello=1  
Hello=2
```

Если задействовать работу еще и 2-х блоков, то программа будет иметь вид:

```
#include <stdio.h>  
__global__ void hello() {  
    int i=threadIdx.x, b=blockIdx.x;  
    printf("Hello block=%d, thread=%d\n",b,i);  
}  
__host__ int main() {  
    hello<<<2, 3>>>();  
    cudaThreadExit();  
}
```

Результатом ее работы будет следующий листинг:

```
Hello block=0, thread=0  
Hello block=0, thread=1  
Hello block=0, thread=2  
Hello block=1, thread=0  
Hello block=1, thread=1  
Hello block=1, thread=2
```

Рассмотренные программы являются простейшими, т.к. передачи данных между памятью CPU и GPU не происходит.

Рассмотрим более сложные программы произведения квадратных заполненных матриц, которые и будут использованы для сопоставления производительности CPU и GPU. Программы разбиты на три класса в соответствии с типом используемой памяти на GPU и библиотеки расчета произведения [5]:

1. При расчете произведения матрицы размещаются в глобальной памяти GPU.
2. Матрицы размещаются в разделяемой памяти GPU.
3. Расчет произведения выполняется подпрограммой cublasDgemm библиотеки cuBLAS.

Первая программа. Предполагается, что матрицы a, b и c размещаются в одномерных массивах. Ниже представлен текст программы с пояснениями:

```
#include <stdio.h>  
#define BLOCK 16  
// Функция умножения двух матриц  
__global__ void mulMatr(double* a,double* b,double* c,int n) {  
    //Получаем id текущей нити.
```

```

int i = threadIdx.y+blockIdx.y*blockDim.y;
int j = threadIdx.x+blockIdx.x*blockDim.x;
//Расчитываем результат.
double sum=0.0;
for(int p = 0; p < n; p++) sum+=a[i*n + p] * b[p*n + j];
c[i*n+j] = sum;
}
__host__ int main() {
int N, M;
double mf=0.0;
printf ( "Input N->");
scanf ( "%d",&N);
printf ( "Matrix = %dx%d elements\n", N,N );
M=N*N;
#define shorter() (double*)malloc(sizeof(double)*M)
//Выделяем память под вектора
double* a = shorter(), * b = shorter(), * c = shorter();
//Инициализируем значения векторов
for (int i = 0; i < N; i++)
for (int j = 0; j < N; j++)
a[i*N+j] = 1.0*((i+1)+2*(j+1)), b[i*N+j] = 1.0/a[i*N+j];
//Указатели на память в видеокарте
double* deva, * devb, * devc;
//Выделяем память для векторов на видеокарте
cudaMalloc((void**)&deva, sizeof(double) * M);
cudaMalloc((void**)&devb, sizeof(double) * M);
cudaMalloc((void**)&devc, sizeof(double) * M);
cudaEvent_t start, stop;
float gpuTime=0.0;
cudaEventCreate ( &start ); cudaEventCreate ( &stop );
//Копируем данные в память видеокарты
cudaMemcpy(deva,a,sizeof(double)*M, cudaMemcpyHostToDevice);
cudaMemcpy(devb, b, sizeof(double)*M, cudaMemcpyHostToDevice);
//Выполняем вызов функции ядра
dim3 threads = dim3(BLOCK,BLOCK);
dim3 blocks = dim3(N/BLOCK,N/BLOCK);
cudaEventRecord(start, 0); // привязываем событие к началу
// выполнения ядра
mulMatr<<<blocks, threads>>>(deva, devb, devc,N);
cudaEventRecord(stop, 0); // привязываем событие к концу
//выполнения ядра
//Получаем результат расчета
cudaMemcpy(c, devc, sizeof(double)*M, cudaMemcpyDeviceToHost);
cudaEventSynchronize(stop); //Дожидаемся выполнение
//ядра, синхронизируя по событию stop
cudaEventElapsedTime (&gpuTime,start,stop);//Запрашиваем время
//между start, stop
//Результаты расчета
gpuTime=gpuTime/1000.0;

```

```

mf=((2.0*N-1)*N*N)/(gpuTime*1000000.0);
printf("time=%.14fsec\nspeed=%0.2fMFlops\n",gpuTime,mf);
printf("i=%d\tj=%d\tC=%.15f\n",
        N/256,N/128,c[(N/256)*N+(N/128)]);
printf("i=%d\tj=%d\tC=%.15f\n",
        3*N/4,5*N/16,c[(3*N/4)*N+(5*N/16)]);
printf("i=%d\tj=%d\tC=%.15f\n",N-4,N-2,c[(N-4)*N+(N-2)]);
// Высвобождаем ресурсы
cudaEventDestroy(start);   cudaEventDestroy(stop);
cudaFree(deva);   cudaFree(devb);   cudaFree(devc);
cudaThreadExit();
delete[] a; a = 0;   delete[] b; b = 0;   delete[] c; c = 0;
}

```

Вторая программа. В функции ядра используется разделяемая память. Здесь принимается, что из-за малого размера разделяемой памяти, исходные квадратные матрицы разбиваются на блочные матрицы размером 16x16 элементов (BLOCK_SIZE 16), как в работе [5].

```

#include <stdio.h>
#define BLOCK_SIZE 16
__global__ void matrixMul( double* C, double* A, double* B, int
wA, int wB) {
Текст представленной программы, которая выполняется на GPU
(__global__ void matrixMul), соответствует тексту программы-
примера, которая поставляется с сайта
http://developer.nvidia.com/cuda-toolkit-32-downloads под именем
matrixMul_kernel.cu.
}
__host__ int main() {
int N, M;
double mf=0.0f;
printf ( "Input N->");
scanf ( "%d",&N);
printf ( "Matrix = %dx%d elements\n", N,N );
M=N*N;
//Выделяем память под вектора
double* a = new double[M], * b = new double[M];
double* c = new double[M];
//Инициализируем значения векторов
for (int i = 0; i < N; i++)
for (int j = 0; j < N; j++)
a[i*N+j] = 1.0*((i+1)+2*(j+1)), b[i*N+j] = 1.0/a[i*N+j];
//Указатели на память в видеокарте
double* deva, * devb, * devc;
//Выделяем память для векторов на видеокарте
cudaMalloc((void**)&deva, sizeof(double) * M);
cudaMalloc((void**)&devb, sizeof(double) * M);
cudaMalloc((void**)&devc, sizeof(double) * M);
cudaEvent_t start, stop;

```



```

float gpuTime=0.0f;
cudaEventCreate ( &start ); cudaEventCreate ( &stop );
//Копируем данные в память видеокарты
cudaMemcpy(deva, a, sizeof(double)*M, cudaMemcpyHostToDevice);
cudaMemcpy(devb, b, sizeof(double)*M, cudaMemcpyHostToDevice);
//Выполняем вызов функции ядра
dim3 threads = dim3(BLOCK_SIZE,BLOCK_SIZE);
dim3 blocks = dim3(N/BLOCK_SIZE,N/BLOCK_SIZE);
cudaEventRecord(start, 0); // привязываем событие
//к началу выполнения ядра
matrixMul<<<blocks, threads>>>(devc, deva, devb,N,N);
cudaEventRecord(stop, 0); // привязываем событие
//к концу выполнения ядра
//Получаем результат расчета
cudaMemcpy(c, devc, sizeof(double)*M, cudaMemcpyDeviceToHost);
cudaEventSynchronize(stop); //Дожидаемся выполнение
// ядра, синхронизируя по событию stop
cudaEventElapsedTime (&gpuTime,start,stop);//Запрашиваем время
//между start, stop
//Результаты расчета
gpuTime=gpuTime/1000;
mf=((2.0*N-1)*N*N)/(gpuTime*1000000.0);
printf("time=%.4fsec\nspeed=%0.2fMFlops\n",gpuTime,mf);
printf("i=%d\tj=%d\tC=%.5f\ni=%d\tj=%d\tC=%.5f\n",
        N/256,N/128,c[(N/256)*N+(N/128)],
        3*N/4,5*N/16,c[(3*N/4)*N+(5*N/16)]);
printf("i=%d\tj=%d\tC=%.5f\n",N-4,N-2,c[(N-4)*N+(N-2)]);
// Высвобождаем ресурсы
cudaEventDestroy(start); cudaEventDestroy(stop);
cudaFree(deva); cudaFree(devb); cudaFree(devc);
delete[] a; a = 0; delete[] b; b = 0; delete[] c; c = 0;
}

```

Третья программа. Для перемножения используется б-ка CUBLAS.

```

#include <stdio.h>
#include <cublas.h>
int main ( int argc, char** argv ) {
float time_seconds=0.0f, mf=0.0f;
int N;
cudaEvent_t start, stop;
cudaEventCreate ( &start ); cudaEventCreate ( &stop );
printf ( "Input N->");
scanf ( "%d",&N);
printf ( "Matrix = %dx%d elements\n", N,N );
int M=N*N;
double *d_A=new double[M],*d_B=new double[M],*d_C=new double[M];
for (int j = 0; j < N; j++)
for (int i = 0; i < N; i++)
A[i+j*N] = 1.0*((i+1)+2*(j+1)), B[i+j*N] = 1.0/A[i+j*N];
cublasInit();

```

```

cublasAlloc ( N * N, sizeof(double), (void**) &d_A);
cublasAlloc ( N * N, sizeof(double), (void**) &d_B);
cublasAlloc ( N * N, sizeof(double), (void**) &d_C);
cublasSetMatrix(N,N,sizeof(double), (void**)A,N, (void**)d_A, N);
cublasSetMatrix(N,N,sizeof(double), (void**)B,N, (void**)d_B, N);
cudaEventRecord(start, 0);
cublasDgemm('n','n',N,N,N,1.0,d_A, N, d_B, N, 0.0f, d_C, N );
cudaEventRecord(stop, 0); cudaEventSynchronize(stop);
cudaEventElapsedTime (&time_seconds,start,stop);
cublasGetMatrix(N,N,sizeof(double), (void**)d_C,N, (void**)C,N);
cublasFree (d_A); cublasFree (d_B); cublasFree (d_C);
cublasShutdown();
time_seconds=time_seconds/1000;
mf=(2.0*N-1)*N*N/(time_seconds*1000000.0);
printf("time=%.4fsec\nspeed=%0.2fMFlops\n",time_seconds,mf);
printf("i=%d\tj=%d\tC=%0.5f\n",
        N/256,N/128,C[(N/256)+(N/128)*N]);
printf("i=%d\tj=%d\tC=%0.5f\n",
        3*N/4,5*N/16,C[(3*N/4)+(5*N/16)*N]);
printf("i=%d\tj=%d\tC=%0.5f\n",N-4,N-2,C[(N-4)+(N-2)*N]);
}

```

В таблице 1 сведены результаты расчетов по представленным выше программам для GPU при двойной точности вычислений. Также представлены результаты и для CPU с использованием библиотек ScaLAPACK и ATLAS по методу, изложенному в работе [4]. Результаты представлены в виде дроби: числитель – время счета в секундах, знаменатель – производительность в Гигафлопсах в секунду. Производительность определялась как отношение числа операций с плавающей точкой при матричном произведении к затраченному времени. Число операций в программах определяется по выражению: $op=N*N(2*N-1)$, где N – число строк или столбцов квадратной матрицы.

Таблица 1

MatNxN	CPU 6 ядер	GPU GeForce GTX 480			GPU Tesla C2075		
		Глоб.	Разд.	cublas	Глоб.	Разд.	cublas
1024	0.062	0.034	0.018	0.014	0.045	0.022	0.008
	34.6	61.9	122.1	155.7	47.3	96.0	284.4
2048	0.407	0.282	0.140	0.105	0.375	0.178	0.058
	42.2	60.8	123.0	163.0	45.9	96.4	295.8
3072	1.234	0.919	0.471	0.352	1.233	0.600	0.194
	47.0	63.1	123.1	164.6	47.0	96.7	299.2
4096	2.870	2.213	1.141	0.835	2.963	1.455	0.458
	47.9	62.1	120.5	164.5	46.4	94.5	300.1
5120	5.687	4.253	2.181	1.626	5.715	2.781	0.893
	47.2	63.1	123.0	165.1	47.0	96.5	300.5
6144	-	7.482	3.771	2.810	10.025	4.805	1.543
		62.0	123.0	165.1	46.3	96.5	300.6

Выводы: 1. Производительность GPU GeForce GTX 480 в 3.5 раза выше производительности CPU AMD Phenom II X6 1090T, а производительность GPU Tesla C2075 в 6.3 раза выше CPU при использовании библиотеки CUBLAS.

2. Использование библиотеки CUBLAS значительно повышает производительность GPU по сравнению с традиционными методами программирования, которые используют большинство пользователей. Использование CUBLAS существенно уменьшает сложность написания программ для задач линейной алгебры.

3. Производительность GPU при матричном умножении с использованием CUBLAS для GeForce GTX 480 практически равна пиковой производительности при двойной точности вычислений (пиковая – 168.1, полученная 165.1 Гигафлопс). Для GPU Tesla C2075 пиковая производительность почти в два раза выше полученной (пиковая – 515.2, полученная 300.6 Гигафлопс). Это указывает на недостаточную эффективность библиотеки CUBLAS для GPU Tesla.

4. Стоимость GPU GeForce GTX 480 в момент написания статьи была равна 2500 грн., а GPU Tesla C2075 – 20000 грн. Поэтому, исходя из соотношения стоимость – производительность выгоднее использование для небольших расчетов, т.е. требующих памяти до 1.5 Гбайт, GPU GeForce GTX 480. Особенно выгодно его использование для расчетов с одинарной точностью, т.к. GPU GeForce GTX 480 работает в 8.4 раза быстрее CPU AMD Phenom II X6 1090T, а GPU Tesla C2075 лишь в 6.5 раза (это показали расчеты по этим же программам, которые были переписаны для чисел с одинарной точностью).

Литература

1. The ScaLAPACK Project [Electronic resource]. – Mode of access : <http://www.netlib.org>
2. Automatically Tuned Linear Algebra Software (ATLAS) [Electronic resource]. – Mode of access : <http://math-atlas.sourceforge.net/>
3. Антонов А. С. Параллельное программирование с использованием технологии MPI : учебное пособие / А. С. Антонов. – М. : Изд-во МГУ, 2004. – 71 с.
4. Мясичев А. А. Достижение наибольшей производительности перемножения матриц на системах с многоядерными процессорами / А. А. Мясичев // Теорія та методика навчання математики, фізики, інформатики : збірник наукових праць. Випуск VIII : в 3-х томах. – Кривий Ріг : Видавничий відділ НМетАУ, 2010. – Т. 3. – С. 174-186.
5. Боресков А. В. Основы работы с технологией CUDA / А. В. Боресков, А. А. Харламов. – М. : ДМК Пресс, 2010. – 232 с.

ВИКОРИСТАННЯ ПАКЕТІВ EXCEL ТА MATHCAD ПРИ ВИВЧЕННІ ДИСЦИПЛІНИ «ЕКОНОМІЧНА ІНФОРМАТИКА»

І. В. Овчарук¹, В. О. Овчарук^{2а}

¹ Україна, м. Київ, Київська державна академія водного транспорту ім. Гетьмана Петра Конашевича-Сагайдачного

² Україна, м. Київ, Національний університет харчових технологій
^а Ovcharuk2004@ukr.net

Основною метою вивчення інформаційних технологій студентами за напрямками «Економіка підприємств», «Менеджмент організацій», «Облік і аудит» та іншими спорідненими напрямками підготовки є необхідність підвищення рівня і якості підготовки фахівців. Для цього потрібно вирішити цілий комплекс задач: формування і розвиток системного мислення, забезпечення усіх видів пізнавальної діяльності, розвиток і закріплення навичок і умінь у сполученні з активними методами навчання. Розвиток засобів інженерних та наукових розрахунків дає можливість фахівцю розв'язувати поставлені задачі без досконалого знання мов програмування, із застосуванням природної математичної нотації. Проте виникає необхідність досконалого володіння таким програмним продуктом, як системи автоматизованих інженерних та економічних розрахунків.

Цілі статті полягають у запропонуванні методики розв'язання задач оптимізації та лінійного програмування, як найбільш популярних в економічних обчисленнях, використовуючи системи автоматизованих інженерних та економічних розрахунків Excel та Mathcad.

Загальна задача лінійного програмування полягає в знаходженні екстремуму (максимуму або мінімуму) функції

$$F = \sum_{j=1}^n c_j x_j \rightarrow \max(\min) \quad (1)$$

за умов

$$\sum_{j=1}^n a_{ij} x_j \begin{cases} \leq \\ = \\ \geq \end{cases} b, \quad i = \overline{1, m}, \quad (2)$$

$$x_j \geq 0, \quad j = \overline{1, k}, \quad (k \leq n). \quad (3)$$

Такі задачі часто зустрічаються на практиці, наприклад, при вирішенні проблем, пов'язаних з розподілом ресурсів, плануванням виробництва, організацією роботи транспорту тощо.

Побудова математичної моделі конкретної задачі передбачає вико-

нання такої послідовності дій:

- уведення змінних, значення яких потрібно знайти;
- формулювання критерію оптимальності, запис цільової функції;
- визначення обмежень на ресурси і вираження цих умов через змінні.

Розглянемо конкретну задачу.

Консервний завод випускає два види продукції, використовуючи при цьому чотири види сировини. Норми витрат сировини, його запаси, а також прибуток від продажу продукції наведені у табл. 1:

Таблиця 1

Вихідні дані

Сировина	Продукція 1-го виду	Продукція 2-го виду	Запаси сировини, кг
S_1	0.3	0.4	165
S_2	0.6	0.4	240
S_3	0.8	0	280
S_3	0	0.1	35
Прибуток від продажу продукції	6	5	

Знайти план випуску продукції, при якому буде досягнуто максимального прибутку.

Покажемо, як розв'язати поставлену задачу засобами Excel і Mathcad. Передбачається, що студенти ознайомлені з цими програмами.

В Excel створюється таблиця з початковими даними (рис. 1).

Через x_1 і x_2 позначимо кількість одиниць продукції першого виду та другого виду відповідно. В клітини D7, D8, D9, D10 вводяться формули, які визначають обмеження по затратах сировини на продукцію кожного виду. В клітині F12 вводиться цільова функція, яка визначає прибуток. У клітинах D13, D14 має бути знайдена кількість відповідної продукції.

Для розв'язування цієї задачі застосуємо засіб «Пошук рішення». Для цього необхідно:

- 1) курсор розмістити у клітині, де введена цільова функція F12;
- 2) Вибрати команду «Пошук рішення» *Сервіс*→ *Поиск решения*.

Відкриється діалогове вікно, в якому вказується розміщення цільової функції, вказується, що цільову функцію потрібно **максимізувати**, та вводяться обмеження (рис. 2).

Результати обчислень відображені на рис. 3. За результатами обчислень кількість продукції першого виду $x_1=250$ одиниць, кількість продукції другого виду $x_2=225$ одиниць. При такому плані випуску продукції прибуток має бути максимальним.

	A	B	C	D	E	F
1						
2				П1	П2	Запаси сировини, кг
3			s1	0,3	0,4	165
4			s2	0,6	0,4	240
5			s3	0,8	0	280
6			s4	0	0,1	35
7			Норма1	=D3*\$D\$13+E3*\$D\$14		
8			Норма2	=D4*\$D\$13+E4*\$D\$14		
9			Норма3	=D5*\$D\$13+E5*\$D\$14		
10			Норма4	=D6*\$D\$13+E6*\$D\$14		
11						Цільова функція
12			Прибуток від продажу одиниці	6	5	=6*D13+E12*D14
13			x1			
14			x2			
15						

Рис. 1. Розміщення вихідних даних в Excel

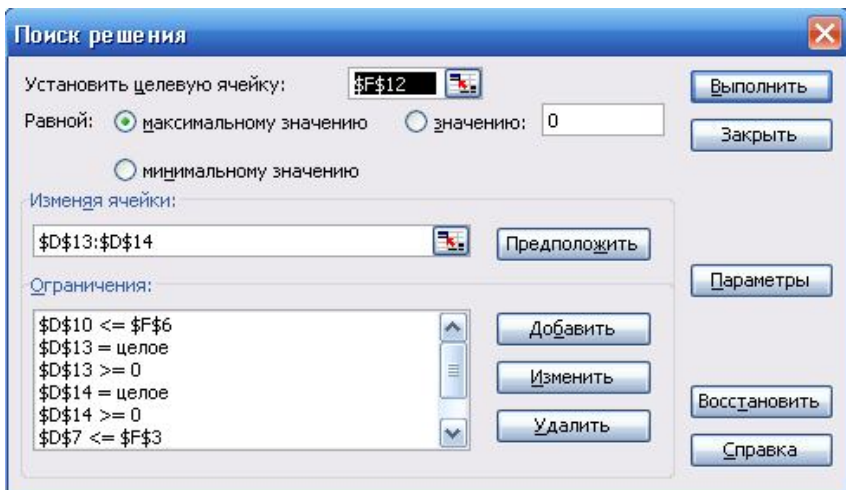


Рис. 2. Вигляд діалогового вікна «Поиск решения»

Тепер розв'яжемо цю задачу засобами пакету Mathcad. Для змінних x_1 і x_2 задаються початкові значення, потім вводиться цільова функція і задається система обмежень (блок Given). Далі розв'язується оптимізаційна задача, тобто знаходяться змінні $x_1=P_0$, $x_2=P_1$ (компоненти вектору

P), враховуючи, щоб цільова функція була максимальною.

	A	B	C	D	E	F
1						
2				П1	П2	Запаси сировини, кг
3			≤1	0,3	0,4	165
4			≤2	0,6	0,4	240
5			≤3	0,8	0	280
6			≤4	0	0,1	35
7			Норма1	165		
8			Норма2	240		
9			Норма3	200		
10			Норма4	22,5		
11						Цільова функція
12			Прибуток від продажу одиниці продукції	6	5	2625,000001
13			x1	250		
14			x2	225		

Рис. 3. Отриманий результат.

$$x1 := 0, \quad x2 := 0$$

$$f(x1, x2) := (6 \cdot x1 + 5 \cdot x2)$$

Given

$$0.3 \cdot x1 + 0.4 \cdot x2 \leq 165$$

$$0.6 \cdot x1 + 0.4 \cdot x2 \leq 240$$

$$0.8 \cdot x1 + 0 \cdot x2 \leq 280$$

$$0.1 \cdot x2 \leq 35$$

$$x1 \geq 0 \quad x2 \geq 0$$

$$P := \text{Maximize}(f, x1, x2)$$

$$P = \begin{pmatrix} 250 \\ 225 \end{pmatrix}$$

$$f(P_0, P_1) = 2.625 \times 10^3$$

Висновки. В роботі приведено детальний розв'язок оптимізаційної задачі, що використовує обидві системи автоматизованих інженерних та економічних розрахунків Excel та Mathcad. Автори сподіваються, що в умовах обмеженості аудиторних годин на вивчення інформатики дані розробки сприятимуть підготовці висококваліфікованих спеціалістів в економіці, маркетингу, менеджменті, обліку і аудиті.

КУРС ІНФОРМАТИКИ ЯК КОМПОНЕНТ СУЧАСНОЇ ПОЧАТКОВОЇ ОСВІТИ

Н. В. Олефіренко

Україна, м. Харків, Харківський національний педагогічний університет
імені Г. С. Сковороди
olefirenkon@mail.ru

В умовах реформування освіти та змін, що відбуваються у суспільстві, особливої уваги потребує початкова ланка освіти. Початкова школа є фундаментом для формування інтелектуальних та загальнонавчальних навичок дитини, формування пізнавальної активності, розвитку самостійності. Саме початкова школа впливає на весь подальший характер взаємовідносин школяра з навчальним середовищем та суспільством.

На сучасному етапі розвитку системи освіти початкова школа зазнає суттєвих змін. Ці зміни пов'язані з широким проникненням інформаційних і комунікаційних технологій у всі галузі діяльності людини, з швидким оновленням інформаційних технологій та високим рівнем інформаційної активності дітей. Про це свідчить:

- зниження віку першого знайомства дитини з комп'ютером. Дитина, яка приходить до школи, як правило, вже має перший досвід використання комп'ютера. Цьому сприяє й наявність в сім'ї побутових цифрових пристроїв – фото- та відеокамер, мобільних телефонів, смартфонів, які сумісні з комп'ютером і передбачають опрацювання даних за допомогою комп'ютера. Крім того, сучасний ринок програмних засобів наповнений мультимедійними програмами розважального та навчального характеру для дітей від 3-4 років. Проте, відсутність цілеспрямованого вивчення інформатики, яка відповідає потребам дитини, компенсуються переважно ігровою діяльністю на комп'ютері.

- поява мобільних пристроїв, підключених до Інтернету. У навчальному процесі потужності подібних мобільних пристроїв зовсім не використовуються. Проте школярі, як правило, добре ознайомлені з подібними пристроями і використовують їх виключно для дозвілля. Крім того, у навчальному процесі можна було б використовувати й інші технічні пристрої автоматизації алгоритмів – побутову й оргтехніку, роботизовані станки, автоматизовані конструктори тощо.

- наявність у школярів молодшого шкільного віку певних вмій використання інформаційних технологій для особистих потреб. На даний час школярі знайомляться з складовими комп'ютера та програмними засобами і ресурсами задовго до початку вивчення курсу інформатики самостійно за мірою їх необхідності – для написання й оформлення

рефератів або доповідей (текстові редактори, програми-браузери, графічні редактори), для спілкування (соціальні мережі, програми для спілкування), для пошуку нових додатків для телефонів, тощо. Разом з тим, оволодіння термінологією та основним понятійним апаратом відбувається стихійно, в рамках особистісного спілкування;

- потреба дітей у змінні підходів до подання нової інформації на уроці. Школярі очікують використання гарних, яскравих ілюстрацій, презентацій, відеозаписів, створення ігрових ситуацій тощо. Це зумовлено змінами, які відбулися у засобах масової інформації – сучасні телепередачі є яскравими, емоційно забарвленими, динамічними з тим, щоб постійно підтримувати увагу глядачів; статті у друкованих виданнях для дітей супроводжуються якісними ілюстраціями, відео демонстраціями тощо; сучасні книжки передбачають виконання активних дій читача – розмалювати, знайти шлях героя, знайти відмінності тощо.

- наявність значної кількості додатків та пристроїв, які завжди можуть допомогти школяреві – калькуляторів та перекладачів у мобільних пристроях та комп'ютерах; електронних довідників; орфографічних словників, вбудованих у текстові редактори. Заборона використання таких додатків може бути змінена на підбір таких завдань, які передбачають здійснення навчальних досліджень і надають школярам можливість скористатися різними засобами для перевірки власної гіпотези.

- готовність вчителів, особливо початкової школи, до використання нових інформаційних технологій у навчанні. У початковій школі вчитель та підручник залишається основним джерелом інформації, і саме від вчителя залежить рівень набутих знань, рівень сформованих загальнонавчальних вмінь.

Зазначені причини обумовлюють необхідність нових підходів до упровадження засобів інформаційно-комунікаційних технологій у навчанні молодших школярів та впровадження такого курсу інформатики, який би задовольняв потребам сьогодення.

У цьому зв'язку цікаво звернутися до досвіду зарубіжжя щодо використання інформаційно-комунікаційних технологій (ІКТ) та відповідної дисципліни у початковій школі.

У зарубіжних країнах існує власний досвід щодо впровадження інформаційних технологій у початкову школу та вивчення школярами курсу ІКТ. Проте у шкільних навчальних планах багатьох країн підкреслюється, що знання з ІКТ є фундаментальними для життя та взаємодії у сучасному суспільстві, і навички використання ІКТ розглядаються як одна із основ освіти, поряд із читанням і писемністю.

У багатьох країнах ІКТ протягом перших 4 років навчання є частиною предметів, пов'язаних із вивченням доквілля та математики (Шот-

ландія, Словенія, Сінгапур).

У шкільній освіті Великої Британії [2] ІКТ є частиною двох міжпредметних галузей – «Інформаційні та комунікаційні технології» та «Дизайн і технології» і за рекомендаціями Міністерства освіти, культури та людських ресурсів вивчаються з першого класу (5 років) в рамках окремого предмету (Information and communication technology).

Вивчення ІКТ відбувається за такими напрямками:

- пошук та аналіз інформації;
- реалізація ідей і прийняття рішень;
- обмін інформацією та взаємодія;
- аналіз, удосконалення та оцінювання власних результатів роботи.

ти.

Протягом першого етапу навчання (5-7 років) школярі навчаються використовувати ІКТ для досягнення власних цілей та для оформлення і реалізації творчих ідей. Школярі навчаються зберігати інформацію у різних формах (наприклад, шляхом введення інформації до підготовлених баз даних), одержувати інформацію із різних джерел (книжок, електронних носіїв, телебачення тощо), представляють інформацію у вигляді тексту, таблиць, графічних зображень; планують і розробляють алгоритми для виконання дій виконавцями, розміщують надані інструкції у правильному порядку; аналізують наслідки власних дій в комп'ютерних середовищах і в реальному житті.

На другому етапі навчання (7-11 років) школярі використовують широкий діапазон комп'ютерного інструментарію, обговорюють способи одержання інформації, вчать здійснювати пошук інформації в Інтернеті, на електронних носіях. На цьому етапі окрема увага приділяється методам перевірки інформації, аналізу достовірності та якості інформації. За допомогою ІКТ школярі розвивають дослідницькі вміння та самостійно відбирають інформацію, яка їм потрібна для роботи.

У Австралії формуванню навичок використання ІКТ у різних формах діяльності приділяється велика увага. Хоч ІКТ є складовою частиною всіх освітніх галузей, школярі початкової школи навчаються ІКТ в галузі «Наука і технології», в 4-5 класах – в рамках освітньої галузі «Технологічні та прикладні дослідження»,

У шкільній освіті Австралії знання, вміння й навички роботи з ІКТ включені до переліку обов'язкових вимог, які пред'являються школярам від дошкільного віку до завершення навчання в школі.

Наприклад, перший рівень навчальних досягнень з ІКТ (у дошкільному віці) передбачає вміння безпечного для здоров'я використання засобів ІКТ. Дошкільнята вчать правильно називати пристрої інформаційно-комунікаційного обладнання, ознайомлюються з основними знач-

ками робочого столу операційної системи, вчать співвідносити рухи мишкою з курсором на екрані комп'ютера. За допомогою вчителя вони працюють з різними типами даних, створюють прості інформаційні продукти та виражають свої ідеї, використовують мультимедійні ресурси для розвитку навичок лічби та письма.

Школярі початкової школи використовують інформаційні ресурси для одержання нових знань і навичок з усіх галузей шкільного навчального плану, створюють і представляють інформацію у зрозумілій формі. Вони вчать систематизувати, класифікувати інформацію, намагаються перевіряти достовірність фактів, які обробляються. Школярі самостійно збирають дані (наприклад, зріст школярів), вводять дані до комп'ютера (електронних таблиць), опрацьовують їх (обчислюють середній зріст школярів) та представляють дані (будують діаграми).

У всіх провінціях Канади визнано інтегруючу роль ІКТ в освіті та ключову роль для підготовки школярів до діяльності у інформаційному суспільстві [1].

Дошкільнята та школярі 6-9 років навчаються одержувати інформацію із електронних джерел, опрацьовувати та порівнювати її, слідкують за планом виконання дослідження, використовують інформаційні технології для організації та відображення даних і результатів розв'язання проблемних завдань. Школярі повинні вміти: визначати методи й інструменти для передання, зберігання, пошуку і відбору інформації; використовувати термінологію, відповідну тим технологіям, які застосовувалися протягом навчання; демонструвати розуміння того, що саме користувач контролює і управляє інформаційними технологіями; визначати технології, що використовуються у повсякденному житті; описувати особливості технологій, які використовують для конкретної мети; спільно працювати з метою ефективного використання обмежених ресурсів, доглядати за обладнанням, усвідомлювати та визнавати право власності електронних матеріалів, дотримуватися техніки безпеки при роботі за комп'ютером та правил етикету при взаємодії.

У стандартах освіти Канади визначено, що молодші школярі повинні володіти початковими навичками роботи з комп'ютером - створювати та редагувати текст за допомогою текстового процесора, створювати електронні повідомлення для пересилання однокласникам; спільно працювати в межах класу за допомогою локальної мережі, одержувати інформацію із підготовлених баз даних тощо.

В школах Росії вивчення інформатики в початковій школі відбувається в межах окремої дисципліни «Інформатика» за різними методичними концепціями.

Так, за програмою авторів О. В. Горячева, Т. О. Волкової, К. І. Горі-

ної «Інформатика в іграх и задачах», вивчення інформатики спрямовано на досягнення цілей: засвоєння системи базових знань щодо внеску інформатики у формування сучасної картини світу; оволодіння вміннями використовувати, аналізувати та перетворювати інформаційні моделі реальних об'єктів; розвиток пізнавальних інтересів, інтелектуальних і творчих здібностей; виховання відповідального ставлення до етичних і правових норм інформаційної діяльності; набуття досвіду використання ІКТ у власній навчально-пізнавальній діяльності [3].

Курс «Інформаційна культура» для школярів 1-4 класів Ю. А. Первіна орієнтований на формування логічного і алгоритмічного мислення та формування вмінь зберігання, опрацювання й передання інформації на базі нових інформаційних технологій. Запропонований курс містить значну кількість задач, передбачає оволодіння інструментарієм комп'ютерної графіки і комп'ютерної музики, навчання школярів технологіям підготовки текстів, роботи з електронними таблицями та базами даних [5].

Метою курсу інформатики для початкової школи авторів С. М. Тур, Т. П. Бокучава є набуття уявлень про інформаційну картину світу та інформаційні процеси, оволодіння вміннями використання комп'ютера як інструменту для роботи з інформацією у навчальній діяльності, і повсякденному житті, виховання інтересу до інформаційної та комунікаційної діяльності, виховання дбайливого ставлення до технічних пристроїв.

Авторський колектив О. Л. Семенова, Т. О. Рудченко, О. В. Щеглової розглядають інформатику як системотвірний елемент змісту освіти початкової школи, як предмет, що підтримує інші дисципліни та створює зручний апарат (лексичний, структурний, логічний) для викладу матеріалу, розв'язання задач і формування технічних навичок школярів. Метою курсу є формування системи знань, уявлень, стереотипів поведінки, які дозволяють школяреві будувати свою поведінку в інформаційній галузі: шукати інформацію, сприймати, збирати, представляти і передавати [6].

Курс інформатики авторського колективу під керівництвом Н. В. Матвеевої спрямований на формування формування системного, об'єктно-орієнтованого та теоретичного мислення, оволодіння прийомami і способами інформаційної діяльності та формування початкових навичок використання комп'ютера для виконання практичних задач.

В Україні рекомендовано вивчення інформатики за програмою «Сходинки до інформатики» авторського колективу під керівництвом Н. В. Морзе. Автори програми наголошують, що курс є підготовчим, таким, що передусє більш широкому і глибокому вивченню базового ку-

рсу інформатики в середній школі, являє собою скорочений систематичний виклад основних питань науки інформатики та інформаційних технологій в елементарній формі та носить світоглядний характер.

Кожна країна вибирає свій шлях для навчання молодших школярів інформаційних технологій. На нашу думку, курс інформатики для молодших класів не повинен дублювати існуючий курс для старшокласників на спрощеному рівні і в той же час повинен відповідати потребам сучасного школяра в опануванні основними вміннями роботи з інформацією. Інформатика у початковій школі повинна мати інтегруючий, міжпредметний характер, об'єднувати всі шкільні дисципліни на основі інформаційної діяльності, пронизувати навчально-пізнавальну діяльність дитини в школі і вдома.

Курс інформатики повинен бути націлений на:

- виховання інформаційної культури школярів. Молодших школярів потрібно навчити грамотно працювати з інформацією – здійснювати пошук інформації з різних джерел, в тому числі в Інтернеті, аналізувати та оцінювати її достовірність, створювати інформаційні моделі реальних об'єктів і процесів, використовувати сучасні технології для пізнання сучасного світу. Крім того, школярів потрібно навчати відповідальному відношенню до інформаційних джерел, дотриманню етичних і правових норм інформаційної діяльності, безпечній поведінці при спілкуванні засобами Інтернету;

- реалізацію творчих задумів за допомогою ІКТ. Школярі можуть малювати, створювати мультфільми, робити презентації, оформляти твори тощо. При цьому сучасних школярів не потрібно обмежувати спеціально створеними середовищами-редакторами, вони швидко опановують інструментарій універсальних та спеціальних програмних засобів та сучасні цифрові пристрої – фотокамери, сканери, мобільні телефони тощо.

- розвиток алгоритмічного та логічного мислення школярів. Саме молодший шкільний вік є найбільш сприятливим для формування і розвитку алгоритмічного і логічного мислення. Разом з тим, для розвитку такого мислення необхідною є спеціальна система завдань і прийомів, які ґрунтуються на використанні життєвого досвіду дитини. Такі завдання можуть бути спрямовані на виконання інструкцій, встановлення послідовності дій, планування власної діяльності, аналіз запропонованих алгоритмів, створення власних алгоритмів, виявлення істотних ознак, логічних взаємозв'язків тощо.

Таким чином, інформатика як окрема навчальна дисципліна у початковій школі, повинна мати інтегруючий, міжпредметний характер і представляти собою середовище для розвитку мислення школярів, їх

інформаційної культури, бути необхідним компонентом у змісті сучасної освіти.

Література

1. Alberta. Alberta Learning. Learning and Teaching Resources Branch. Classroom assessment tool kit for the Information and Communication Technology (ICT) program of studies : grades 1–6 [Electronic resource]. – 2003. – 130 p. – Mode of access : <http://www.learning.gov.ab.ca/ict>
2. Department for Children, Schools and Families, England, UK. National Curriculum for England. Information and Communication Technology. The National Curriculum for England [Electronic resource]. – London : Department for Children, Schools and Families, England, UK, 2000. – Mode of access : <http://www.nc.uk.net>
3. Волкова Т. О. Информатика в играх и задачах : методические рекомендации для учителя 1 класса четырехлетней начальной школы Школа 2100 / Волкова Т. О., Горячев А. В., Горина К. И. – М. : Экспресс, Баласс, 2010. – 128 с.
4. Обучение информатике во втором классе : методическое пособие / Н. В. Матвеева, Н. К. Конопатова, Л. П. Панкратова, Е. Н. Челак. – М. : БИНОМ. Лаборатория знаний, 2004. – 287 с.
5. Первин Ю. А. О проекте сквозного курса «Информационной культуры» / Первин Ю. А. // Педагогическая информатика. – 1994. – №2.
6. Семенов А. Л. Учебная программа «Информатика 1-3» / Семенов А. Л., Рудченко Т. А., Щеглова О. В. // Информатика и образование / – 1998. – №6. – С. 19-23.

ПРОФЕСІЙНИЙ РОЗВИТОК СТУДЕНТА У СИСТЕМІ ФАХОВОЇ ПІДГОТОВКИ ВЧИТЕЛЯ ІНФОРМАТИКИ

Н. С. Павлова

Україна, м. Рівне, Рівненський державний гуманітарний університет
pavlovaNata@ukr.net

Період навчання у вищому навчальному закладі (ВНЗ) відіграє важливу роль у процесі формування фахівців, оскільки саме у цей час студенти набувають компетентностей певного рівня, оволодіння якими сприяє їхньому професійному й особистому розвитку. Тому питання професійного становлення студентів через призму їхньої навчально-пізнавальної діяльності знаходиться в центрі уваги викладачів ВНЗ.

До того ж швидкі темпи розвитку інформаційних та комунікаційних технологій, педагогічних програмних засобів вказують на те, що сучасний вчитель інформатики повинен: бути готовий до постійного самодосконалення, оновлення своїх знань; вміти грамотно працювати з різноманітними засобами доступу до інформаційних ресурсів; володіти високим рівнем сформованих соціально- та професійно-значущих компетентностей. Професійний розвиток студентів при вивченні фахових дисциплін сприяє не лише набуттю компетентних знань, виробленню механізмів їх запровадження в педагогічну діяльність, але й формуванню конкурентно спроможних фахівців та їх подальшому фаховому зростанню.

Саме тому система підготовки майбутніх вчителів інформатики у педагогічному ВНЗ повинна бути спрямована на формування творчих, професійно і соціально відповідальних фахівців, здатних не лише виділяти педагогічні проблеми, але й компетентно шукати шляхи та засоби їх вирішення. Але в умовах існуючої організації навчального процесу у ВНЗ важко сформувати активного, творчого фахівця з яскраво вираженими індивідуальністю, самостійністю, цілеспрямованістю, готового до подальшого саморозвитку.

Теоретично-практичні аспекти підготовки студентів до майбутньої професійної діяльності висвітлено у працях С. У. Гончаренка, Н. В. Кузьміної, В. І. Лозової, А. К. Маркової [4], Г. О. Михаліна, І. П. Підласого, Н. Ф. Талізної, М. І. Шкіля та ін. Основи компетентнісного підходу при фаховій підготовці студентів описували Б. Блум, І. А. Зязюн, В. В. Краєвський, О. В. Овчарук, О. І. Пометун, А. В. Хуторський [8] та ін.

Педагогічні стратегії інформатизації навчального процесу на всеможливих освітніх рівнях досліджено у працях А. Ф. Верляня, А. П. Єршова, М. І. Жалдака [2], Ю. І. Машбиця, Ю. С. Рамського [3],

С. О. Семерікова [7], Ю. В. Триуса та інших науковців.

Питання удосконалення системи підготовки майбутніх учителів інформатики до фахової діяльності, формування професійних компетентностей у різний час вивчали Н. В. Апатова, В. Ю. Биков, В. І. Ключко, Е. І. Кузнецов, О. А. Кузнецов, М. П. Лапчик, Н. В. Морзе [5], С. А. Раков, Є. М. Смірнова-Трибульська, О. І. Теплицький та ін.

Компетентнісний підхід при підготовці майбутніх фахівців – це орієнтація на цілі навчальної діяльності студентів, що ставляться з урахуванням результату, який розглядається не як сума знань, умінь й навичок, а як здатність особистості компетентно діяти в різних професійних ситуаціях. При цьому навчально-пізнавальна діяльність студентів повинна підсилюватися мотиваційними чинниками, оскільки мотиви є внутрішньою умовою, що визначають ефективність будь-якої діяльності індивіда, основою активного розвитку особистості як суб'єкта цієї діяльності.

Під розвитком особистості розуміють процес руху від нижчого (простого) рівня розвитку до вищого (складного) внаслідок накопичення кількісних змін, що ведуть до якісних перетворень [6, 158]. Зокрема, низькі показники навчальних досягнень студентів молодших курсів пояснюємо не лише невмінням оперувати прийомами розумової діяльності (наприклад, аналізом через синтез, порівнянням, виділенням головного), але й низькою пізнавальною мотивацією, а старших курсів – не усвідомленням своїх спонукань до педагогічної діяльності, мотивів, уявлень про власні соціально-професійні ролі. Відзначимо, що перші роки навчання у педагогічному ВНЗ характеризуються: формуванням студентського колективу та визначенням власної позиції у ньому; усвідомленням власних здібностей, запитів та інтересу до обраної професії; самоствердженням. Труднощі, які виникають у студентів при вивченні фахових дисциплін на першому та другому курсах можна пояснити не лише низьким рівнем інтелектуального розвитку, але й не вмінням визначити власні цінності, самостійно працювати, контролювати власні результати, організовувати роботу у групах, виділяти проміжні етапи, прогнозувати здобуті результати та ін.

Серед чинників, які впливають на підвищення якості навчання студентів старших курсів виділимо професійну спрямованість дисциплін, фахову значущість практичних завдань, підвищення особистісної самооцінки, здатність студентів до перенесення знань з однієї предметної галузі в іншу. Серед інших факторів, які впливають на якість навчального процесу варто виділити: здатність студентів аналізувати й оцінювати власні професійні можливості та співвідносити їх з сучасними критеріями, здатність до саморозвитку, інтеграції знань. Як бачимо, якість підго-

товки майбутніх учителів у педагогічному ВНЗ, набуття ними професійних компетентностей певного рівня залежить від багатьох факторів, серед яких найважливішими є мотивація, що базується на потребах, інтересах, спрямованості особистості та індивідуальна пізнавальна діяльність студентів з опорою на творчість.

Компетентність фахівця базується на формуванні у нього професійно значущих якостей і здатностей та включає наступні аспекти: розуміння сутності виконуваних завдань; знання досвіду, який є в цій галузі; вміння обирати засоби та способи дії, адекватні конкретним ситуаціям; почуття відповідальності за досягнуті результати; здатність коригувати процес досягнення цілей; готовність до подальшого саморозвитку.

Під професійною компетентністю вчителя розуміємо єдність його теоретичної та практичної готовності, значущих особистісних якостей та досвіду, що зумовлюють готовність до здійснення педагогічної діяльності та особистісної інтерпретації здобутих у процесі навчання результатів. Інакше кажучи, професійна компетентність містить наступні компоненти: вміння системно аналізувати педагогічну реальність і діяти в ній, здатність орієнтуватися в предметній галузі, навички володіння сучасними фаховими технологіями.

Науковці, беручи до уваги розвиток суспільства та освіти, аспекти педагогічної діяльності та основні функції вчителя у загальноосвітньому закладі, професійні компетентності розглядають через сукупність взаємопов'язаних загально-професійних й предметних компетентностей [1–3]. Уточнимо зміст загально-професійних та предметних компетентностей, якими, з одного боку, має володіти майбутній вчитель інформатики з позицій професійної діяльності, а з іншого – які обов'язково мають охоплювати змістові лінії шкільного курсу «Інформатика».

Загально-професійні компетентності вчителя інформатики окреслюють коло його здатностей до теоретичного та методологічного використання професійних знань у педагогічній діяльності. У їх структурі виділяють наступні компетентності: дидактико-методичні, організаційно-управлінські, психолого-педагогічні, дослідницькі, комунікативні, природничо-математичні.

Оскільки предметні компетентності характеризують професійний профіль фахівця, ідентифікують його професійну діяльність в конкретній предметній галузі на певному кваліфікаційному рівні, то майбутній вчитель інформатики повинен володіти: інформаційно-комунікаційними й інформаційно-методологічними компетентностями, компетентностями в галузях моделювання, алгоритмізації й програмування.

М. І. Жалдак уточнює вимоги до якості фахової підготовки вчителя і відзначає, що вчитель повинен мати «універсальні, фундаментальні

знання, щоб мати можливість ефективно в педагогічному плані використовувати засоби сучасних інформаційно-комунікаційних технологій, створювати для дітей умови для повного розкриття їхнього творчого потенціалу, нахилів і здібностей, задоволення запитів і навчально-пізнавальних потреб» [2, 8].

Структурна модель професійного розвитку студентів відображена на рис. 1.

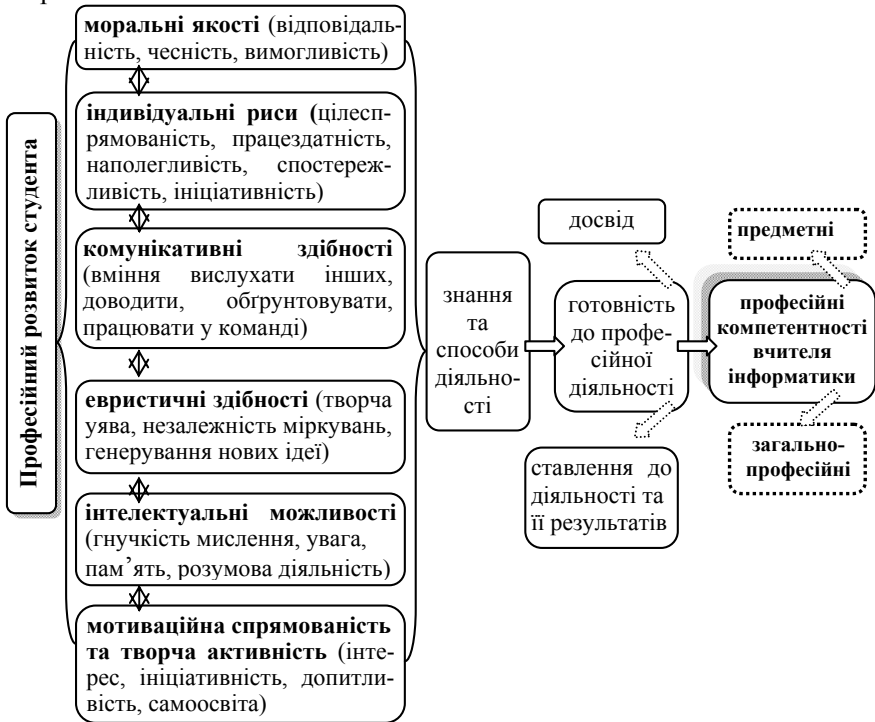


Рис. 1. Структурна модель професійного розвитку студента

Для того, щоб навчальний процес у ВНЗ спонукав студентів до професійного розвитку, необхідно спрямувати навчання на трансформацію одного виду діяльності в інший з відповідною зміною мотивів, цілей, дій, засобів і результатів навчання, зокрема навчально-пізнавальну діяльність послідовно переводити у навчально-дослідницьку і навчально-професійну та залучати майбутніх вчителів до спроб реалізувати себе в рамках фахової діяльності. Кожний вид діяльності характеризується професійними завданнями, специфічними новоутвореннями особистості студента як суб'єкта цієї діяльності. Для встановлення на кожному етапі тісного зв'язку теоретичної підготовки з практикою, надання результа-

там навчання практично значимого характеру, що має важливе значення для фундаменталізації знань студентам варто пропонувати:

- розв'язувати професійно орієнтовані завдання, які переплітаються між собою та не є відірваними від реальності;
- аналізувати всеможливі педагогічні ситуації, інтегруючи міжпредметні зв'язки з одного боку, і враховуючи диференціацію навчання у відповідності до вікового розвитку, нахилів, запитів і здібностей учнів, з іншого боку;
- обговорювати творчі ідеї, нестандартні рішення не лише з викладачем, але й один з одним у парах, групах;
- працювати з різноманітними засобами доступу до інформаційних ресурсів, опрацьовувати різні види повідомлень та даних;
- самостійно проводити дослідження, реалізовувати проектну діяльність;
- публічні виступи та дискутування з опорою на висловлювання та захист власної позиції, доведення істинності тверджень та ін.;
- виважено, обґрунтовано та доцільно поєднувати сучасні інформаційні технології з традиційними засобами навчання.

На основі аналізу навчальних планів педагогічних ВНЗ та робочих програм, відзначимо, що орієнтація навчального процесу на професійний розвиток студента реалізується при вивченні всіх без винятку дисциплін, під час аудиторних занять, самостійної роботи, обчислювальної та педагогічної практик, на основі побудови відповідних стратегій діяльності викладачів з використанням сучасних комп'ютерно-орієнтованих методичних систем навчання конкретних предметів. Але найвагоміше узагальнення знань й умінь, розвиток у студентів готовності до професійної діяльності у РДГУ відбувається при вивченні дисциплін «Методика навчання інформатики», «Вибрані питання шкільного курсу інформатики з методикою викладання у профільній школі».

Наведемо приклади завдань, які пропонуємо студентам на практичних заняттях з названих вище дисциплін.

1. При вивченні теми «Психолого-дидактичні основи навчання інформатики» студенти описують навчальний процес (табл. 1), беручи до уваги «модель збагачення навчання» Дж. Рензуллі, таксономію Б. Блума та вимоги до навчальних досягнень учнів з інформатики. При цьому студенти повинні знаходити відповіді на найбільш важливі для майбутньої професійної діяльності запитання: що роблять учні на певному етапі навчально-пізнавальної діяльності та навіщо учні це роблять.

2. При вивченні теми «Комп'ютерні мережі» студентам пропонується розробити веб-квести, використання яких дозволяє змоделювати актуальні практичні ситуації, організувати роботу з інформаційним прос-

тором гіпермедіа як з одним цілим, максимально реалізувати міжпредметні зв'язки. Крім того, робота з веб-квестами формує у студентів стійкі інтереси, допитливість, спонукає до творчих пошуків, виробляє уміння орієнтуватися в інтенсивному потоці повідомлень, що стосуються відповідної предметної галузі й суміжних з нею, поєднувати індивідуальну та колективні роботи. Як показує досвід, вибір тематики (наприклад, «Інтернет у школі», «Інформаційна безпека школяра», «Інтернет на допомогу батькам», «Готуємо стенди в кабінет інформатики») та формулювання проблемних завдань, які будуть відображені у веб-квесті може відбуватися по різному. В одних випадках викладач визначає напрямок роботи з урахуванням навчальної ситуації зі свого предмету, здібностей, професійної готовності студентів, в інших – тематику і проблемне завдання добирають студенти самостійно, орієнтуючись на власні інтереси, пізнавальні та творчі можливості. Оскільки на кожному етапі роботи студенти самостійно добирають навчальний матеріал, спосіб подання, визначають рівень його деталізації, структурують завдання, визначають спосіб взаємодії учителя, учнів та ІКТ, то варто зазначити, що така дослідницька діяльність спонукає студентів розвивати власні здібності й інтереси, усвідомлювати рівень готовності до майбутньої фахової діяльності та покращувати його.

Таблиця 1

Опис навчального процесу на основі таксономії Блума

Навчальні цілі		Ключові слова, що описують		Ключові слова, що містяться в умові завдань, які пропонуються учням
рівень	зміст	діяльність вчителя	навчальні дії учня	
Знання	запам'ятовування навчального матеріалу та його відтворення	виділяє об'єкти вивчення; пояснює; уточнює; демонструє; досліджує; описує	запам'ятовує правила-орієнтири; розпізнає об'єкти вивчення; відшукує відомості та відносить їх до певної категорії знань; відтворює послідовності дій;	розкрити зміст; навести приклад; виокремити етапи розв'язування; пояснити призначення; виділити головне; знайти пояснення в довідковій системі
Розуміння				
Використання				
Аналіз				
Синтез				
Оцінювання				

Розвиток особистості студента у процесі навчання у ВНЗ є складовою його професійної підготовки, умовою ефективної педагогічної діяльності. Основою адаптації майбутнього вчителя у фаховому середовищі, професійно-особистісного розвитку, орієнтиром у вивченні всеможливих педагогічних ситуаціях є знання з усіх без винятку дисциплін. Поєднання фундаментальних знань та професійної спрямованості навчання, орієнтація навчального процесу на формування у студентів професійно-значущих компетентностей певного рівня є головними чинниками якісної підготовки вчителя-професіонала.

Література

1. Борытко Н. М. Профессионально-педагогическая компетентность педагога [Электронный ресурс] / Н. М. Борытко. – 2007. – Режим доступа : <http://www.eidos.ru/journal/2007/0930-10.htm>
2. Жалдак М. І. Система підготовки вчителя до використання інформаційно-комунікаційних технологій в навчальному процесі / М. І. Жалдак // Науковий часопис НПУ імені М.П. Драгоманова. Серія №2. Комп'ютерно-орієнтовані системи навчання : зб. наук. праць / Редрада. – К. : НПУ імені М.П. Драгоманова, 2011. – №11 (18). – С. 3-16.
3. Жалдак М. І. Модель системи соціально-професійних компетентностей вчителя інформатики / М. І. Жалдак, Ю. С. Рамський, М. В. Рафальська // Науковий часопис НПУ імені М.П. Драгоманова. Серія № 2. Комп'ютерно-орієнтовані системи навчання : зб. наук. праць / Редрада. – К. : НПУ імені М.П. Драгоманова, 2009. – № 7(14). – С. 3-10.
4. Маркова А. К. Психологический анализ профессиональной компетентности учителя / Маркова А. К. // Сов. педагогика. – 1990. – №8. – С. 82-88
5. Морзе Н. В. Основы методичної підготовки вчителя інформатики : монографія / Н.В. Морзе. – К. : Курс, 2003. – 372 с.
6. Психологічний словник [за ред. Войтка В. І.]. – К. : Вища школа, 1982. – 216 с.
7. Семеріков С. О. Фундаменталізація інформатичної освіти / С. О. Семеріков // Науковий часопис НПУ імені М.П. Драгоманова. Серія № 2. Комп'ютерно-орієнтовані системи навчання : зб. наук. праць / Редрада. – К. : НПУ ім. М.П. Драгоманова, 2009.– № 7(14). – С. 40-49.
8. Хуторской А. В. Технология проектирования ключевых и предметных компетенций / А. В. Хуторской. – 2005. – Режим доступа <http://eidos.ru/journal/2005/1212.htm>

ПЕРСПЕКТИВИ ФУНДАМЕНТАЛІЗАЦІЇ ШКІЛЬНОГО КУРСУ ІНФОРМАТИКИ

О. П. Поліщук, І. О. Теплицький, С. О. Семеріков
Україна, м. Кривий Ріг, Криворізький національний університет
semerikov@gmail.com

Інформатична освіта має практико-орієнтовану спрямованість, яка полягає в тому, що практика є не лише джерелом нових задач, а й критерієм для вибору можливих напрямів досліджень. Це означає, що прогрес інформатики відбувається як під впливом внутрішніх потреб розвитку, так і під впливом запитів практики (задач, що виникають в математиці, економіці, природознавстві, інженерії, всередині самої інформатики тощо). М. І. Жалдак наголошує, що фундаментальні знання мають важливе значення для прикладних досліджень, а потреби повсякденної виробничої практики викликають і стимулюють пізнавальну діяльність, спрямовану на розкриття законів фундаментального характеру, що в свою чергу є одним з аспектів гуманітаризації освіти [1]. Тому фундаменталізація інформатичної освіти у вищій педагогічній школі сприятиме проникненню ідей фундаменталізації й у шкільний курс інформатики. Як зазначає Ю. В. Триус та О. В. Копаєв, «... навіть з технологічної точки зору на уроках інформатики в середній школі необхідно знайомити учнів з найбільш загальними принципами функціонування систем, в тому числі – й програмних» [2]. М. П. Лапчик наголошує, що «... школі потрібен учитель інформатики з фундаментальними знаннями в галузі інформатики» [3].

Сьогодні більшість школярів, маючи високий рівень мотивації, постійний доступ до сучасних апаратних і програмних засобів, спеціалізованої літератури та ресурсам мережі Інтернет, досить добре володіють багатьма інформаційними технологіями. Нерідкою є ситуація, коли з ряду причин знання або вміння учнів з окремих тем курсу інформатики виявляються більш актуальними або навіть більш детальними, ніж у вчителя, тому навчання інформатики необхідно для систематизації знань школярів, підкреслення фундаментальної складової курсу, знайомства із загальними принципами та підходами до подання та обробки повідомлень.

Фундаментальний характер інформатики як науки суттєво впливає на розвиток змісту навчання інформатики в школі. На думку І. В. Левченко, фундаменталізація шкільного курсу інформатики «повинна означати не переорієнтацію на вивчення в школі основ фундаментальної інформатики як науки, а виділення фундаментальних основ науки та їх

дидактичну переробку для освіти школярів за допомогою інформатики для оволодіння школярами соціального досвіду людства» [4, 12].

Фундаменталізація навчання інформатики в школі можлива при реалізації вчителем професійно-педагогічної діяльності з урахуванням ряду умов, в числі яких адекватне відображення сучасного стану інформатики як фундаментальної науки, подання цілісного курсу інформатики на основі інтеграції змісту навчання навколо системотвірних стержнів, формування та розвиток мислення учнів, навчання ефективним способам роботи з інформацією, активне використання внутрішньо-предметних і міжпредметних зв'язків курсу інформатики, навчання узагальнених способів застосування сформованих знань і умінь на практиці.

Важливою частиною змісту шкільного курсу інформатики є вивчення інформаційних і комунікаційних технологій (ІКТ), оволодіння школярами вміннями й навичками застосування засобів ІКТ для розв'язування навчальних і практичних завдань. Для багатьох учителів, методистів і авторів підручників інформатики формування вмінь використання засобів ІКТ – взагалі головна (якщо не єдина) ціль цього курсу, що не відповідає загальноосвітньому характеру цього курсу та суперечить державному стандарту. Витоки такої позиції знаходяться в історії введення інформатики в школу під гаслом необхідності «забезпечення комп'ютерної грамотності молоді», що саме по собі вже орієнтувало цей курс переважно на формування вмінь працювати з комп'ютером. Наприкінці 80-х – початку 90-х років ХХ ст., коли масовими загальнодоступними засобами ІКТ стали так звані «офісні пакети», зміст курсу інформатики орієнтувався більшою мірою на роботу із програмними засобами, що входять до складу цих пакетів. Підручники інформатики середини 90-х рр. усе більше нагадували збірник інструкцій для користувача.

Однак уже через кілька років знову постало питання про використання загальноосвітнього потенціалу інформатики, її внеску у світогляд, розвиток особистості, соціалізацію школярів і т.д. У стандарті [5] й типових програмах з інформатики (див., наприклад, [6]), розроблених під керівництвом М. І. Жалдака, було чітко визначено місце ІКТ в інформаційній освіті, з'явилися питання єдності інформаційних процесів у біологічних, соціальних і технічних системах, інформаційного моделювання, соціальної інформатики та ін. Так, автори експериментального навчального посібника з інформатики для 7 класу [7] спеціально зосереджували увагу на основних типах та головних функціях програмного забезпечення замість детального розгляду певного програмного продукту, тому такий підручник є фундаментальним.

Відзначимо також, що прийняттю нових позицій щодо цілей і зміс-

ту курсу інформатики сприяла ще одна обставина. Критика радянської системи освіти, що розгорнулася на початку 90-х років ХХ ст., зараз уже перестала носити тотальний і безапеляційний характер, а стала більш конкретною та обґрунтованою. Зокрема, багато в чому справедливі докори відносно надмірно «академічного» характеру шкільної освіти та пропозиції відмовитися від принципів фундаментальності, системності й повноти його змісту, протиставлення їх компетентнісному підходу поступилися місцем більше стриманій позиції. Якщо компетентності – це обізнаність, «знання в дії» [8, 3], то діяльність, *дії не можуть бути ефективними, якщо вони не мають системного характеру, не відповідають вимогам повноти й не спираються на фундаментальні знання*. Так само актуальна зараз вимога мобільності освіти може бути реалізована тільки за рахунок фундаментальності освіти. Саме ця якість освіти дає можливість у короткий термін опановувати нові технології та способи діяльності, зробити людину мобільною, затребуваною на ринку праці.

Ці тенденції стають все більш виразними в останні роки, коли зміна поколінь засобів ІКТ відбувається настільки стрімко, що знання, уміння й навички в галузі конкретних версій цих технологій, одержувані в середній школі, втрачають свою актуальність і досить швидко стають незатребуваними.

Сьогодні вже не можна будувати вивчення ІКТ в основному на тренінгу типових умінь роботи з основними засобами цих технологій, орієнтуючи його на численні вправи та розв'язування завдань репродуктивного характеру. Не можуть залишитися осторонь і такі принципи навчання, як фундаментальність, системність, свідомість у навчанні. Повною мірою необхідно задіяти й внутрішньопредметні зв'язки шкільного курсу інформатики, особливо розділи, пов'язані з поданням повідомлень, формалізацією й моделюванням, властивостями алгоритмів тощо.

Відповідно до теорії діяльності, головним змістом навчання повинні бути загальні способи дій для розв'язування широких класів завдань, щоб діяльність учнів була спрямована на оволодіння цими загальними способами. П. Я. Гальперін відзначав, що всі надбання в процесі навчання можна розділити на дві нерівні частини: одну становлять нові загальні схеми речей, які обумовлюють нове їхнє бачення й нове мислення про них, іншу – конкретні факти й закони досліджуваної галузі, конкретний матеріал науки [9]. Освоєння загальних схем вимагає універсальних способів дій, у той час як конкретний матеріал пов'язаний з вузькопредметними, переважно виконавчими діями. Не заперечуючи необхідності формування конкретних дій, найбільшу увагу потрібно приділяти загальним способам дій, пов'язаним із використанням фундаментальних знань, які носять інваріантний характер.

Отже, доцільно було б побудувати зміст навчання ІКТ у курсі інформатики на основі виділення інваріантної частини (наукові основи ІКТ) та варіативної частини (навички роботи з конкретними версіями засобів ІКТ). При цьому варіативна частина могла б скласти основний зміст практичних (лабораторних) робіт із цього курсу. На думку Ю. В. Триуса, саме інваріантна частина повинна бути відображена у підручниках з інформатики, а варіативна – у посібниках, робочих зошитах, практикумах тощо.

Як основні *принципи добору змісту фундаментальних, наукових основ ІКТ* можна виділити такі: єдність подання повідомлень для всіх технологій; єдність у методах і засобах опрацювання повідомлень (і даних); побудова ІКТ на основі алгоритмів, що забезпечують автоматизацію опрацювання повідомлень (даних). Таким чином, в якості системоутворюючого поняття при фундаменталізації змісту навчання ІКТ може бути використане поняття «інформаційний процес». Відібрані дидактичні одиниці змісту навчання інформатики, в числі яких вимірювання і кодування інформації, інформаційне моделювання, основи логіки, алгоритмізації, управління та інші є значущими й для фундаменталізації методичної підготовки вчителя інформатики.

І. В. Левченко [4, 8–9] визначено *умови професійно-педагогічної діяльності вчителя в контексті фундаменталізації навчання інформатики*: формування цілісного курсу інформатики на основі інтеграції змісту навчання навколо системоутворюючих стержнів, навчання ефективним способам роботи з інформацією, наповнення навчального матеріалу гуманітарною складовою, адекватне відображення в шкільному курсі сучасного стану фундаментальної науки інформатики, розкриття емоційно-ціннісних та моральних відносин, формування та розвиток мислення учнів, активне використання внутрішньопредметних і міжпредметних зв'язків курсу інформатики, навчання узагальнених способів застосування сформованих знань та вмінь на практиці. Націлювання методичної підготовки вчителів інформатики на створення умов для розкриття учням походження й сутності фундаментальних теоретичних знань, залучення учнів в різноманітні форми спілкування і діяльності, оволодіння узагальненими способами дій, самостійних відкриттів учнями в процесі пошукової діяльності, структурування розумових процесів, прояви багатства і складності розумових операцій (опора на власні думки, самостійність в узагальненнях, критична оцінка одержуваної інформації, активне застосування розумових операцій і т.п.) сприяє підвищенню ефективності навчання школярів інформатики.

Таким чином, можливі два основні напрями фундаменталізації шкільного курсу інформатики:

1) математизація змісту навчання й розвиток формального компонента діяльності (центральними поняттями інформатики стають алгоритм і комп'ютер);

2) побудова курсів інформатики від феномена інформації та інформаційних процесів до методів їх вивчення за допомогою інформаційних моделей шляхом використання комп'ютера як засобу управління інформаційними процесами.

Ці два підходи цілком об'єктивні й відображають процеси, що відбуваються в усьому світі, але далеко не рівноправні з погляду знань, що формуються. Разом з тим, найбільш перспективним є курс, що об'єднує ці підходи на основі широкого застосування комп'ютерного моделювання. Яскраві приклади фундаменталізації шкільного курсу інформатики наведені у роботах, присвячених впровадженню інтегративного курсу моделювання (зокрема, [10; 11; 12]). На жаль, кількість таких робіт досить незначна, тому фундаменталізація шкільного курсу інформатики в Україні все ще залишається практично нерозробленим напрямом, незважаючи на те, що збірник 1987 року «Вивчення основ інформатики та обчислювальної техніки в середній школі: досвід та перспективи» закінчувався тим, що «навчальні програми з усіх предметів в школах ФРН базуються ... на фундаментальних знаннях ..., і інформатика ... не є виключенням. Застосовуваний у ФРН ... підхід у навчанні інформатики заслуговує найпильнішої уваги. Необхідно відзначити також, що в США, ФРН та ... Великобританії спостерігається тенденція до виявлення фундаментальних понять ... у шкільних курсах інформатики. Аналіз зарубіжного досвіду навчання інформатики слід враховувати і при побудові курсу інформатики та обчислювальної техніки у радянській школі» [13, 190–191].

Література

1. Жалдак М. І. Формування інформаційної культури вчителя [Електронний ресурс] / Жалдак М. І., Хомік О. А. – [30 листопада 1998]. – Режим доступу : www.icfcst.kiev.ua/SYMPOSIUM/Proceedings/Galdak.doc

2. Копаєв О. В. Фундаментальний аспект базового курсу інформатики / Копаєв О. В., Триус Ю. В. // Сучасний стан і перспективи шкільних курсів математики та інформатики у зв'язку з реформуванням у галузі освіти (Дрогобич, 14–16 листопада 2000 р.) : [всеукраїнська науково-практична конференція] : тези доповідей. – Дрогобич : ДДПУ, 2000. – С. 138–140.

3. Лапчик М. П. Структура и методическая система подготовки кадров информатизации школы в педагогических вузах : дис. на соискание ученой степени доктора пед. наук в форме научн. докл. : 13.00.02 / Лап-

чик М. П. – М., 1999. – 82 с.

4. Левченко И. В. Развитие системы методической подготовки учителей информатики в условиях фундаментализации образования вузах : дис. на соискание ученой степени доктора пед. наук : 13.00.02 – теория и методика обучения и воспитания (информатика) / Левченко Ирина Витальевна. – М., 2009. – 46 с.

5. Державний стандарт освітньої галузі “Технології” (проект) для загальноосвітньої середньої школи / Биков В. Ю., Жалдак М. І., Морзе Н. В., Мостіпан О. І., Рамський Ю. С. // Освіта України. – 2003. – № 3-4. – 10 с.

6. Навчальні програми для профільного навчання профільного навчання. Інформатика / Жалдак М. І., Морзе Н. В., Мостіпан О. І. [і ін.] – К. : 2003. – 320 с.

7. Жалдак М. І. Інформатика – 7 : експериментальний навчальний посібник для учнів 7 класу загальноосвітньої школи / Жалдак М. І., Морзе Н. В. – К. : ДіаСофт, 2000. – 207 с.

8. Суворова Т. Н. Совершенствование методики изучения информационных технологий в школьном курсе информатики : автореф. дис. на соискание ученой степени канд. пед. наук : 13.00.02 – теория и методика обучения информатике / Суворова Татьяна Николаевна ; Вятский гос. гуманитар. ун-т – М., 2007. – 22 с.

9. Гальперин П. Я. Развитие исследований по формированию умственных действий / Гальперин П. Я. // Психологическая наука в СССР. – М. : Изд-во АПН РСФСР, 1959. – Т. 1. – 599 с.

10. Поліщук О. П. Систематичне навчання моделюванню в підготовці майбутнього вчителя / О. П. Поліщук, І. О. Теплицький, С. О. Семеріков // Комп’ютерне моделювання в освіті : матеріали Всеукраїнського науково-методичного семінару. – Кривий Ріг, 26 квітня 2006 р. – Кривий Ріг : КДПУ, 2006. – С. 48-49.

11. Теплицький І. О. Елементи комп’ютерного моделювання : навчальний посібник. – Кривий Ріг : КДПУ, 2010. – 264 с., іл.

12. Семеріков С. О. Теоретико-методичні основи фундаменталізації навчання інформатичних дисциплін у вищих навчальних закладах : дис... д-ра пед. наук : 13.00.02 – теорія і методика навчання (інформатика) / Семеріков Сергій Олексійович ; Національний педагогічний ун-т ім. М. П. Драгоманова. – К., 2009. – 536 арк. – Бібліогр.: арк. 470–536.

13. Штырлина И. А. Зарубежный опыт изучения программирования и информатики в средней школе капиталистических стран / Штырлина И. А. // Изучение основ информатики и вычислительной техники в средней школе : опыт и перспективы / Сост. В. М. Монахов [и др.] – М. : Просвещение, 1987. – 192 с. : ил. – (Б-ка учителя математики)

О ПРЕПОДАВАНИИ ТЕОРИИ ГРАФОВ ПРИ ПОДГОТОВКЕ СПЕЦИАЛИСТОВ В ОБЛАСТИ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

С. А. Поттосина^{1α}, Ю. В. Поттосин^{2β}

¹ Беларусь, г. Минск, Белорусский государственный университет
информатики и радиоэлектроники

² Беларусь, г. Минск, Объединенный институт проблем информатики
НАН Беларуси

^α s.pottosina@gmail.com

^β pott@newman.bas-net.by

Введение

Теория графов давно уже стала самостоятельным разделом дискретной математики, который особенно бурное развитие получил за последние более чем полвека. Об этом свидетельствуют многочисленные монографии, изданные как за рубежом, так и в нашей стране (см., например, [1–6]). В своем столь бурном развитии теория графов обязана, прежде всего, быстрому и значительному расширению сфер ее приложений за данный период, а также сравнительной простоте и наглядности ее терминологии. Методы теории графов широко применяются в логическом проектировании дискретных управляющих и вычислительных устройств. Поэтому в курсе дискретной математики для студентов, специализирующихся в области автоматизации проектирования систем управления, особое место должна занимать теория графов. О приложениях теории графов в логическом проектировании и близких к нему областях написано довольно много работ, перечислить которые было бы сложно, поэтому ограничимся только указанием некоторых монографий [7–9] и обзорных статей [10; 11].

На этапе логического проектирования дискретных устройств управления возникают разнообразные логико-комбинаторные задачи [12–14], решение которых приводит к оптимальным вариантам логического проекта. Среди многочисленных задач такого рода значительное место занимают задачи, сформулированные в терминах теории графов. В настоящей работе предлагается краткий обзор тех задач и методов теории графов, с которыми целесообразно знакомить студентов при подготовке специалистов в области информатики и вычислительной техники.

В рассматриваемых задачах понятие графа используется в разных интерпретациях. Чаще всего это графическое представление бинарного отношения типа совместимости, т.е. отношения, обладающего свойствами рефлексивности и симметричности. Такое отношение представля-

ется произвольным неориентированным графом. В некоторых случаях граф представляет отношение квазипорядка или отношение следования между некоторыми событиями. В отдельных задачах граф представляет логическую сеть. Рассматриваемые в этих случаях графы являются ориентированными.

1. Раскраска графа

Значительное место в логическом проектировании занимает задача раскраски графа, которая заключается в получении такого разбиения множества вершин заданного графа на непересекающиеся подмножества (блоки), что никакие две вершины, принадлежащие одному и тому же подмножеству (т.е. раскрашенные в один и тот же цвет), не являются смежными. Число таких подмножеств должно быть минимальным.

Раскраска графа может быть использована при решении задачи декомпозиции булевых функций [13; 14], занимающей важное место в синтезе комбинационных схем в базе СБИС. Из всех задач логического проектирования, пожалуй, именно она наиболее часто сводится к раскраске графа.

Одна из постановок задачи декомпозиции формулируется следующим образом. Задана система булевых функций как векторная булева функция $y = f(x)$, и заданы векторные булевы переменные w и z , составленные из компонент векторной переменной x . Необходимо представить заданную систему в виде $f(x) = h(w, g(z))$. Функции h и g должны быть более простыми, чем f . В данном случае это означает, что число s компонент вектора z и сумма чисел компонент векторов w и $u = g(z)$ меньше n . Обычно стремятся получить минимум этой суммы.

Общий метод решения данной задачи предполагает построение графа G , вершинами которого являются значения z^* векторной переменной z , а ребра определяются следующим образом. Пусть x^*_i – значение векторной переменной x , а z^*_i и w^* – значения векторных переменных z и w соответственно, причем компоненты x^* совпадают с соответствующими компонентами z^*_i и w^* . Вершины z^*_i и z^*_j графа G связаны ребром, если найдется значение w^* вектора w такое, что значения функции f при соответствующих x^*_i и x^*_j различны.

Легко доказывается, что заданную векторную функцию можно представить в виде $f(x) = h(w, u)$, где векторный аргумент u имеет k компонент, тогда и только тогда, когда хроматическое число графа G не превосходит 2^k .

Для получения окончательного решения достаточно получить минимальную раскраску графа G и закодировать полученные цвета булевыми векторами, которые представят значения функции g при соответствующих значениях вектора z .

Ввиду того, что задача раскраски графа является NP-трудной [15] и точное ее решение не всегда практически достижимо, в работе [13] предлагается эвристический метод, который позволяет получать раскраску, довольно близкую к минимальной за практически приемлемое время.

Точный метод раскраски, т.е. метод, гарантирующий получение минимального количества цветов, представлен в работах [13; 16], где предлагается обход дерева поиска, с некоторыми не тупиковыми вершинами которого связаны множества возможных вариантов выбора цвета для очередных вершин раскрашиваемого графа. При этом учитывается возможность однозначности определения цвета для вершин, когда имеет место один из указанных выше случаев.

Дальнейшее усовершенствование данного метода описано в статье [17]. Там же приведен простой и быстрый алгоритм получения приближенно минимальной раскраски графа, эффективно работающий на графах с большим числом вершин, относительно небольшим числом ребер и большим разбросом степеней вершин. Согласно этому алгоритму вершины графа раскрашиваются в порядке, который заранее определяется путем нахождения и удаления вершины с минимальной степенью. При этом указывается возможность определить верхнюю оценку хроматического числа графа, которая может быть использована при поиске минимального решения.

Двойственной задачей по отношению к раскраске графа является задача разбиения графа на полные подграфы. Всякая раскраска любого графа представляет собой разбиение на полные подграфы графа, дополнительного по отношению к исходному графу. Иногда удобнее решать именно эту задачу, чем задачу раскраски графа, например в случае, когда число ребер графа значительно больше, чем число ребер его дополнительного графа.

Точное решение задачи раскраски графа можно получить путем сведения ее к задаче о кратчайшем покрытии [13]. Для этого надо найти в графе все максимальные независимые множества, т. е. такие множества вершин, где никакие две из них не связаны ребром, и получить наименьшую совокупность этих множеств такую, что любая вершина графа принадлежит хотя бы одному из этих множеств. Удалив вершины так, чтобы каждая вершина входила только в одно множество, получим искомого разбиение. Этот метод может быть эффективным при небольшом количестве максимальных независимых множеств в графе. Для некоторых классов графов, однако, это число экспоненциально растет с ростом числа вершин.

2. Кодирование графа. Двудольные подграфы

Как было сказано выше, при декомпозиции булевых функций в конечном счете требуется закодировать булевыми или троичными векторами цвета, полученные после раскраски графа. Данные векторы считаются значениями векторной переменной \mathbf{u} , обладающей тем свойством, что каждому ребру графа должна соответствовать ее компонента, принимающая противоположные значения (0 и 1) на концах этого ребра. Таким образом, каждой компоненте вектора \mathbf{u} можно поставить в соответствие двудольный подграф рассматриваемого графа, а данную задачу можно рассматривать как задачу покрытия графа двудольными подграфами. Такой подход к решению задачи декомпозиции булевых функций описан в работах [17; 18].

Задача покрытия графа двудольными подграфами возникает при поиске оптимального кодирования состояний синхронного автомата, описанном в работах [13; 14]. Предполагается задание входа и выхода автомата в структурном алфавите. Такая модель названа в работе [16] автоматом с абстрактным состоянием. Автомат представляется в виде двух троичных матриц U и V . Столбцам матрицы U соответствуют входные булевы переменные x_1, x_2, \dots, x_n , столбцам матрицы V – выходные булевы переменные y_1, y_2, \dots, y_m . Строки обеих матриц помечены состояниями автомата так, что пара их одноименных строк представляет переход из состояния, являющегося меткой строки матрицы U , в другое состояние, являющееся меткой строки матрицы V . Строка матрицы U представляет при этом условие перехода, а строка матрицы V – двоичные выходные сигналы, сопровождающие этот переход.

Метод заключается в следующем. Для некоторого столбца u_i матрицы V строится граф G , вершинам которого соответствуют состояния заданного автомата. В данном столбце u_i отыскиваются пары элементов с противоположными значениями (0 и 1), которым соответствуют пары неортогональных строк матрицы U . Вершины графа G , соответствующие меткам строк всякой такой пары, связываются ребром и затем кодируются, как показано выше. Полученные коды переносятся в матрицы U и V в виде значений вновь введенных внутренних переменных z_1, z_2, \dots, z_k . Эта процедура повторяется для каждого столбца матрицы V , причем каждый раз выбирается столбец с минимальным числом пар указанных элементов. Когда граф G для каждого столбца оказывается пустым (после введения новых переменных строки становятся ортогональными), процесс заканчивается, а преобразованные матрицы U и V представляют систему булевых функций, описывающую комбинационную часть логической сети, реализующей заданный автомат. Некоторые совместимые состояния оказываются закодированными неортогональ-

ными или даже одинаковыми кодами. В этом случае фактическое число состояний автомата оказывается уменьшенным.

В работе [19] описан метод выделения в заданном графе G остовного двудольного подграфа (т. е. такого подграфа, множество вершин которого совпадает с множеством вершин заданного графа) с множествами вершин V^0 и V^1 и максимальным числом ребер. Алгоритм, основанный на данном методе, состоит из двух этапов. На первом этапе находят все циклы нечетной длины, на втором этапе решается задача покрытия множества полученных циклов ребрами графа. Ребра, составляющие найденное покрытие, затем удаляются, в результате чего получается искомым подграф. В зависимости от конкретной реализации данного метода решение может быть как точным, так и приближенным.

Эвристический метод решения этой задачи предложен в работе [18] и уточнен в статье [17]. Согласно этому методу начальное значение V^0 представляет одноэлементное множество, а V^1 содержит все остальные вершины. Затем вершины из множества V^1 последовательно переносятся в множество V^0 до тех пор, пока не перестанет увеличиваться число ребер, связывающих вершины из V^0 с вершинами из V^1 .

3. Независимые множества. Полные подграфы

Задачи логического проектирования, формулируемые как задачи разбиения заданного множества на совместимые в некотором смысле подмножества, удается свести к раскраске графа, если попарной совместимости элементов подмножества достаточно для их групповой совместимости. В противном случае приходится вводить какие-то дополнительные условия одноцветности вершин. Другой подход к решению таких задач предполагает нахождение в заданном графе наибольшего независимого множества, т.е. множества попарно несмежных вершин, а затем группирование около них остальных вершин заданного графа.

С помощью этого подхода можно решать такие задачи, как минимизация длины кода состояний асинхронного автомата, сжатие таблицы переходов автомата (по строкам и столбцам), параллельная декомпозиция автомата, упрощение системы булевых функций, описывающих асинхронный автомат. Этот прием особенно подходит для решения таких задач абстрактного синтеза автомата, как минимизация числа состояний синхронного автомата и его декомпозиция. В этих задачах искомая совокупность совместимых подмножеств должна обладать свойством замкнутости, которое заключается в том, что наличие каких-либо элементов в одном совместимом подмножестве требует включения некоторых других элементов в другое совместимое подмножество из получаемой совокупности.

Двойственной задачей по отношению к задаче нахождения незави-

симых множеств является задача нахождения полных подграфов, поскольку полный подграф некоторого графа порождается независимым множеством его дополнения.

Для получения всех максимальных независимых множеств можно с успехом применять классический метод Полла-Ангера построения всех максимальных совместимых множеств состояний автомата. Алгоритм, основанный на этом методе, описан в терминах теории графов в работах [13; 14]. В нем используется прием, при котором заданный граф разлагается на последовательность подграфов и решение данной задачи для предыдущего подграфа преобразуется в решение для последующего подграфа. В этих же работах описан алгоритм получения всех максимальных независимых множеств, использующий лексикографический перебор.

В ряде задач требуется найти одно наибольшее независимое множество или один наибольший полный подграф. Как известно, независимое множество и вершинное покрытие графа являются дополнениями друг друга. Вершинное покрытие представляет собой множество вершин такое, что каждое ребро заданного графа имеет в этом множестве хотя бы один свой конец. Наименьшее вершинное покрытие, являющееся дополнением наибольшего независимого множества, получается как кратчайшее строчное покрытие матрицы инцидентности.

4. Задачи на графах в синтезе структур СБИС

Довольно значительное многообразие задач на графах встречается при оптимизации структур сверхбольших интегральных схем (СБИС). Задачи оптимизации структур СБИС лежат на стыке этапов логического и технического проектирования. При оптимизации структуры программируемой логической матрицы (ПЛМ) особое место занимает задача свертки ПЛМ, которая заключается в использовании одной и той же линии (горизонтальной или вертикальной) для укладки различных проводников, что приводит к сокращению площади кристалла, занимаемой проектируемой схемой.

К раскраске графа в работе [20] сводится получение многократной свертки ПЛМ, при которой на одной линии укладывается более чем два проводника, а внешние соединения переносятся в «третье измерение». Вершины рассматриваемого графа соответствуют проводникам, и две вершины связаны ребром, если и только если соответствующие проводники не могут быть уложены на одной линии. Раскраска графа указывает способ укладки проводников.

При заданном разбиении множества внешних сигналов на два подмножества, соответствующие сигналам, подводимым к ПЛМ с разных сторон, задача нахождения максимального числа свертываемых пар сво-

дится к задаче нахождения максимального паросочетания в двудольном графе [20]. Доли этого графа соответствуют упомянутым подмножествам, и две вершины из разных долей связаны ребром, если соответствующие сигналы образуют пару свертки.

При свертывании горизонтальных линий ПЛМ также решается задача размыкания контуров в двудольном орграфе, после чего надо должным образом упорядочить вертикальные шины. Это упорядочение сводится к задаче топологической сортировки вершин бесконтурного орграфа [20]. Решением этой задачи является последовательность вершин, в которой для каждой дуги вершина, являющаяся концом этой дуги, расположена после вершины, являющейся ее началом.

Заключение

Приведенный обзор не охватывает многочисленных работ по приложениям теории графов на всех этапах проектирования дискретных устройств, но показывает, насколько широко методы теории графов могут быть использованы для решения задач на всех стадиях логического проектирования. Рассмотренные задачи теории графов должны быть представлены в курсе дискретной математики для студентов, специализирующихся в области проектирования дискретных устройств.

Литература

1. Берж К. Теория графов и ее применения / К. Берж. – М. : Иностранная литература, 1962. – 319 с.
2. Харари Ф. Теория графов / Ф. Харари. – М. : Мир, 1973. – 300 с.
3. Кристофидес Н. Теория графов. Алгоритмический подход / Н. Кристофидес. – М. : Мир, 1978. – 432 с.
4. Оре О. Теория графов / О. Оре. – М. : Наука, 1980. – 336 с.
5. Зыков А. А. Основы теории графов / А. А. Зыков. – М. : Наука, 1987. – 381 с.
6. Лекции по теории графов / В. А. Емеличев, О. И. Мельников, В. И. Сарванов, Р. И. Тышкевич. – М. : Наука, 1990. – 384 с.
7. Мелихов А. Н. Ориентированные графы и конечные автоматы / А. Н. Мелихов. – М. : Наука, 1971. – 416 с.
8. Мелихов А. Н. Применение графов для проектирования дискретных устройств / А. Н. Мелихов, Л. С. Берштейн, В. М. Курейчик. – М. : Наука, 1974. – 304 с.
9. Алгоритмы и программы решения задач на графах и сетях / Нечепуренко М. И., Попков В. К., Майнагашев С. М. и др. – Новосибирск : Наука, 1990. – 515 с.
10. Закревский А. Д. Приложения теории графов к задачам логического проектирования дискретных устройств / Закревский А. Д., Потто-

син Ю. В., Шнейдер А. А. // Исследования по прикладной теории графов. – Новосибирск : Наука, Сибирское отделение, 1986. – С. 3–9.

11. Поттосин Ю. В. Задачи теории графов в логическом проектировании / Поттосин Ю. В. // Логическое проектирование, вып.6. – Минск : Ин-т техн. кибернетики НАН Беларуси, 2001. – С. 106-130.

12. Агибалов Г. П. Технология решения комбинаторно-логических задач методом сокращенного обхода дерева поиска / Г. П. Агибалов, В. А. Беляев. – Томск : Изд-во Томского ун-та, 1981. – 126 с.

13. Закревский А. Д. Логические основы проектирования дискретных устройств / А. Д. Закревский, Ю. В. Поттосин, Л. Д. Черемисинова. – М. : Физматлит, 2007. – 592 с.

14. Поттосин Ю. В. Основы теории проектирования цифровых устройств / Ю. В. Поттосин. – Saarbrücken : LAP LAMBERT Academic Publishing, 2011. – 336 с.

15. Гэри М. Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон. – М. : Мир, 1982. – 416 с.

16. Закревский А. Д. Логический синтез каскадных схем / А. Д. Закревский. – М. : Наука, 1981. – 416 с.

17. Закревский А. Д. Раскраска графов при декомпозиции булевых функций / Закревский А. Д. // Логическое проектирование. – Минск : Ин-т техн. кибернетики НАН Беларуси, 2000. – Вып. 5. – С. 83-97.

18. Бибило П. Н. Синтез комбинационных схем методами функциональной декомпозиции / П. Н. Бибило, С. В. Енин ; ред. А. Д. Закревский ; АН БССР, Ин-т техн. кибернетики. – Минск : Наука и техника, 1987. – 190 с.

19. Поттосин Ю. В. Выделение максимальной двудольной части в неориентированном графе / Ю. В. Поттосин // Автоматизация решения логико-комбинаторных задач. – Минск : Ин-т техн. кибернетики АН БССР, 1985. – С. 22-28.

20. Бибило П. Н. Синтез комбинационных ПЛМ-структур для СБИС / П. Н. Бибило ; АН Беларуси, Ин-т техн. кибернетики. – Минск : Наука и техника, 1992. – 232 с.

**ДЕЯКІ АСПЕКТИ ВИКЛАДАННЯ ДИСЦИПЛІНИ
«ІНФОРМАТИКА ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»
СТУДЕНТАМ ТЕХНОЛОГІЧНОГО НАПРЯМКУ
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ХАРЧОВИХ ТЕХНОЛОГІЙ**

О. Л. Сєдих^α, С. В. Маковецька^β

Україна, м. Київ, Національний університет харчових технологій

^α olgased@ukr.net

^β Svetlana-konko@yandex.ru

Сучасна економічна система вимагає від вищої школи підготовки таких фахівців, які вміють швидко приймати якісні та конструктивні рішення щодо стратегічних економічних завдань. При цьому вони повинні мати здатність не стільки визначати, які саме з набутих знань, навичок та відомих інструментів необхідно застосовувати в тій чи іншій ситуації, скільки творчо вирішувати нестандартні задачі та генерувати нові знання, виходячи з поставленої мети. Тому фундаментальною задачею інформатики як науки є розробка та впровадження інноваційних технологій навчання, які б забезпечували підготовку фахівців, що відповідають зазначеним вимогам.

При цьому, на думку викладачів кафедри інформатики НУХТ, першочергове значення набуває завдання формування змісту навчального курсу відповідно до фахової спрямованості майбутнього випускника, вдосконалення сучасних технологій навчання, які б забезпечували поряд з істотним підвищенням теоретичної та практичної підготовки студентів, подальшу методологічну орієнтацію процесу навчання на підтримання та розвиток особистісного потенціалу кожного окремого студента.

Метою вивчення дисципліни «Інформатика та інформаційні технології» є формування у майбутніх фахівців необхідного рівня інформаційної культури, необхідної для конкурентоспроможності випускників ВНЗ на ринку праці і успішної роботи в різних областях сучасного бізнесу, набуття практичних навичок використання сучасних інформаційних технологій для розв'язання різноманітних задач у процесі навчання у ВНЗ та роботі за фахом.

Завданням вивчення дисципліни є теоретична та практична підготовка майбутніх фахівців з таких питань:

1. Поняття інформатики та інформаційних технологій, їх роль та місце в сучасному суспільстві.

2. Архітектура та принципи функціонування комп'ютерів, їх програмне забезпечення.

3. Технологія роботи у середовищі сучасних операційних систем.
4. Технологія опрацювання текстових документів.
5. Технологія створення, редагування та форматування електронних таблиць та діаграм для обробки технологічної інформації.
6. Технологія використання інструмента «Поиск решения» для розв'язання задач оптимізації.
7. Створення баз даних в.
8. Глобальна комп'ютерна мережа Internet.
9. Експорт, імпорт та зв'язування об'єктів у середовищі інтегрованого пакету.
10. Використання математичного пакету Mathcad при виконанні наукових та інженерних розрахунків.
11. Розробка проектів інженерних задач на мові Visual Basic.

Вища професійна освіта – процес взаємодії двох складових викладача і студента: перший повинен уміти передавати знання, другий - їх одержувати і творчо засвоювати. Зміст освіти відображає стан суспільства, перехід від одного його стану до іншого. В даний час – це перехід від індустріального суспільства ХХ століття до постіндустріального або інформаційного ХХІ століття. Розвиток і функціонування освіти обумовлений усіма чинниками і умовами існування суспільства: економічними, політичними, соціальними, культурними і іншими. Разом з тим мета освіти – розвиток людини, що відповідає вимогам того суспільства, в якому він живе, що знаходить своє віддзеркалення в зв'язку освіти і культури. «...Необхідно уміти розкріпачити мислення студента – навчити його швидко і кваліфіковано вибирати потрібні методи і необхідне програмне забезпечення, творчо обґрунтовувати найвдаліші рішення з метою отримання конкретного результату» [1, 80]. На сучасному етапі розвитку освіти як один з перспективних напрямів розвитку і модернізації вищої школи розглядається інформатизація, що передбачає розробку і впровадження в педагогічну практику сучасних інформаційних і телекомунікаційних засобів, а також передових технологій навчання. Все це стосується і використання програмного забезпечення навчальних курсів, що викладаються у вищій школі.

При вивченні інформатики в НУХТ можна виділити такі основні послідовні етапи викладення матеріалу:

- проведення лекцій (формування апарату понять і категорій, зв'язок з іншими дисциплінами);
- виконання лабораторних робіт (розвиток самостійної роботи студентів, практичне використання знань);
- самостійне вивчення та закріплення матеріалу: написання рефератів (вмінню працювати з літературою);

– опитування і спільне обговорення запропонованої теми (розвиток мислення, вміння викладати вивчений матеріал, формування системного підходу);

– тестові завдання та модульний контроль (формування оптимальних варіантів прийняття рішень, стимулювання до навчання, розвиток навчальних та психологічних основ до іспиту – вміння орієнтуватися у вивченому матеріалі);

– участь у науково-практичних студентських конференціях (формування навиків роботи з літературою, вміння синтезувати та систематизувати отримані знання, розвиток творчості).

З 2005 року НУХТ використовує кредитно-модульну систему. Рейтингова система оцінювання знань передусім активізує самостійну роботу студента, робить її ритмічною і систематичною протягом семестру; краще мотивується навчально-пізнавальна діяльність, підвищується роль стимулювання самостійності, ініціативи, відповідальності, творчості у навчально-пізнавальній діяльності; підвищується об'єктивність оцінювання, значно зменшується психологічне навантаження під час іспиту. Крім того, вона дає змогу викладачеві здійснювати індивідуалізацію навчання та диференційований підхід до студентів, сприяє раціональному вибудовуванню студентом навчання протягом семестру, особливо під час заліково-екзаменаційних сесій.

Таким чином, рейтинг – це той зовнішній мотиваційний стимул, що спонукає амбітних студентів до більш значних зрушень і досягнень. Кожен студент з будь-яким рівнем підготовки має змогу проявити себе, відчутти свій прогрес, зробити свій поступовий крок до кращого, а відчуття успіху водночас є найкращим стимулом для подальшої роботи і навчання [2, 85-86].

Рейтингова система оцінювання може бути дуже ефективною, якщо вона зрозуміла для студентів, чітка і прогнозована. Для цього студенти мають бути ознайомлені з вимогами курсу та критеріями оцінювання, тому на першій лекції кожній академічній групі видається зміст модулів навчальної дисципліни, їх оцінювання в балах, форми і термін контролю навчальної роботи студентів.

Перші роки впровадження рейтингової системи йшов пошук найбільш ефективного використання оцінки знань студентів. У всіх варіантах рейтингової системи максимальна кількість балів, яку студент може набрати за вивчення навчальної дисципліни протягом семестру та під час іспиту 100 балів. Вибіралося оптимальні варіанти оцінювання лабораторних робіт та модульного контролю кожного розділу робочої програми дисципліни. Було визначено, що зі 100 балів студент протягом семестру може набрати максимальну кількість балів – 60 і під час іспи-

тів – 40.

Викладачі кафедри інформатики іноді запропонують студентам додаткові види роботи творчого характеру для отримання додаткових, бонусних балів. Адже додаткові завдання не є обов'язковими, і студенти обирають їх з двох причин: 1) цікавість, інтерес; 2) бажання підвищити свій рейтинг.

Актуальна проблема викладання інформатики у ВНЗ – це різний початковий рівень студентів-першокурсників. Для її розв'язання викладачами кафедри був розроблений і створений ВЕК (Відео Електронний Консультант) по ряду тем циклу лабораторних робіт. ВЕК може використовуватися як при самостійній роботі студентів, так і під час лабораторних занять. Цей засіб дозволяє в короткий термін опрацювати матеріал та підвищити рівень комп'ютерної підготовки студентів.

Одним із видів контролю та оцінювання знань студентів є комп'ютерні тести, які використовуються для поточного контролю знань. Це дозволяє зробити процес оцінювання знань студентів більш об'єктивним. Використання тестів забезпечує одночасну перевірку знань студентів всієї групи та формує у них мотивацію та зацікавленість до підготовки до кожного заняття, дисциплінує їх.

При вивченні дисципліни «Інформаційні технології в інженерних розрахунках» згідно освітньо-професійної програми підготовки фахівців технологічних спеціальностей також передбачено використання математичного пакету. Проте повноцінне використання новітніх комп'ютерних технологій неможливе без програмування розв'язування інженерних задач – необхідного кроку до ефективного використання сучасних технологій. Одним із розділів робочої програми дисципліни «Інформатика та інформаційні технології» є розділ, пов'язаний з алгоритмізацією та програмуванням інженерних задач, що формує у студентів алгоритмічне мислення. Мовою програмування була вибрана Visual Basic. Технологія роботи у середовищі Visual Basic базується на ідеях об'єктно-орієнтованого та візуального програмування. Об'єктно-орієнтоване програмування – це методика розробки програм, в основі якої лежать поняття об'єкту. Середовище візуального програмування Visual Basic – це графічна автоматизована оболонка над об'єктно-орієнтованою версією мови Basic. Якщо у мові Basic структурними одиницями є дані та команди, то тут такою структурною одиницею є візуальний об'єкт, який називається компонентом. Автоматизація програмування досягається завдяки можливості переносити компонент на форму (у програму) з палітри компонентів і змінювати його властивості, не вносячи вручну змін до програмного коду.

Сучасний курс інформатики повинен дати знання, які будуть базою

для розуміння можливостей та обмежень використання персональних комп'ютерів та програмного забезпечення. Вивчення курсу зумовлює отримання фундаментальних знань в області інформатики. Програма курсу інформатики розрахована на якісне знання обов'язкового мінімуму шкільної інформатики. Курс інформатики у вищій школі націлений на подальше поглиблення знань з основ інформаційних технологій та мови програмування на основі базових знань, які отримані першокурсниками у школі. Насправді ж, всі базові поняття доводиться вивчати із самого початку у зв'язку з тим, що студенти не володіють необхідними знаннями для подальшого вивчення предмета.

Література

1. Иосилевский Л. Острые проблемы современного высшего образования / Иосилевский Л. // Высшее образование в России. – 1997. – №1. – С. 79-84.
2. Сілкова О. В. Застосування автоматизованих атестаційних систем та класифікація тестових завдань при використанні комп'ютерних технологій / Сілкова О. В. // Комп'ютерно-орієнтовані системи навчання : зб. наук. праць. – К. : НПУ ім. М. П. Драгоманова, 2001. – Випуск 3.

МЕЖДИСЦИПЛИНАРНЫЕ СВЯЗИ В ДИСКРЕТНОЙ МАТЕМАТИКЕ: КОМАНДНАЯ ПРОЕКТНАЯ ДЕЯТЕЛЬНОСТЬ

С. М. Сейдаметова^α, А. С. Аджикельдиев^β

Украина, г. Симферополь, Крымский инженерно-педагогический университет

^α lilya.lds@gmail.com

^β adzhykeldiev@gmail.com

Подготовка квалифицированных специалистов в университетах является сложным процессом. Его условно можно разделить на несколько последовательных этапов, причем с каждым этапом сложность обучения возрастает. В связи с этим процесс обучения должен строиться таким образом, чтобы на каждом этапе студенту давалось то количество сведений, которое необходимо для дальнейшего изучения выбранной им специальности. Поэтому особенно важно на начальном этапе обучения, чтобы студенты изучали те дисциплины, которые являются фундаментальными для данной специальности, то есть являются основой, предоставляющей знания для дальнейшего изучения студентом выбранной им специальности.

«Дискретная математика» – это именно такая фундаментальная дисциплина, изучение которой является необходимой основой для дальнейшего обучения студентов специальности «Информатика». Она является важным звеном математического образования, потому что исследования в различных областях математики, связанные с ней, занимают значимое место в этой науке. Для информатики дискретная математика необходима в связи с тем, что с помощью её инструментария можно наглядно изучить принципы работы и эксплуатации всех современных электронных вычислительных машин, а также автоматизированных систем управления и проектирования.

Кроме этого, аппарат дискретной математики позволяет достичь общеобразовательных целей, таких как привитие студенту навыков логического и математического мышления, развитие творческого и критического мышления.

Изучение дискретной математики базируется на знаниях курса алгебры и начал анализа, геометрии и физики в пределах средней школы и преподаётся параллельно с дисциплиной «Математическая логика», в связи с чем разделы «Множества и отношения», «Комбинаторика», «Исчисление высказываний» и «Исчисление предикатов» в курсе «Дискретной математики» не рассматриваются (рис. 1).

В рамках учебной дисциплины «Дискретная математика» предпола-

гается обучить студентов основам дисциплины, где дискретность понимается как антипод непрерывности. Изучение дисциплины «Дискретная математика» предусматривает обучение методам и мышлению, характерным для дискретной математики, а также ознакомление с основами математической логики, теории алгоритмов, теории графов и их применение к задачам математической кибернетики. Дискретная математика служит основой для таких предметов, как «Архитектура вычислительных систем», «Алгоритмы и структуры данных», «Программирование», «Теория информации и кодирование».



Рис. 1. Пререквизиты и постреквизиты дисциплины «Дискретная математика»

Весь курс преподавания дисциплины «Дискретная математика» условно можно разделить на несколько частей (модулей, разделов). Это «Двузначная логика», « k -значная логика», «Теория графов» и «Теория кодирования».

Разделы «Двузначная логика», « k -значная логика» являются основой для формирования дискретного математического мышления, поэтому их изучению должно уделяться достаточно времени. Главными задачами первых двух частей являются:

– предоставление студентам первоначальных сведений о таких понятиях, как дискретная функция, суперпозиция, функционально полная система;

– обучение разным способам задания дискретных функций (табличный способ, задание с помощью формул, с помощью полиномов и нормальных форм, геометрическое задание с использованием n -мерного куба);

– ознакомление с приёмами исследования полноты и замкнутости систем функций, а также с приёмами минимизации булевых функций;

– предоставление сведений о важнейших замкнутых классах алгебры логики и о каноническом разложении k -значных функций;

– изучение методов исследования систем на полноту и замкнутость, а также эквивалентных преобразований формул;

– определение основных отличий между k -значной логикой ($k > 2$) и булевой алгеброй.

Изучению разделов «Теория графов» и «Теория кодирования» также должно уделяется много внимания, так как методы теории алгоритмов широко используются при описании и исследовании структурных свойств объектов в самых разных областях науки. Изучение данных разделов направлено на:

– ознакомление с основными понятиями, методами и языком теории графов;

– обучение построению графов на плоскости, выделению подграфов в графе;

– умение находить в графе циклы и маршруты;

– приобретение навыков построения матриц смежности, бинарных матриц и матриц инцидентности;

– умение определять метрические характеристики графов;

– навыки осуществления поиска в ширину и глубину;

– умение укладывать граф в заданное пространство и на сферу;

– способность различать плоские и планарные графы;

– знание основных определений «Теории кодирования»;

– навыки работы с самокорректирующимися кодами и алфавитными кодами, а также с кодами с минимальной избыточностью.

Современный уровень развития информационных технологий внёс значительные перемены в жизнь общества. Эти перемены коснулись и образования, способствовали появлению совершенно новой педагогики в целом, а также новых педагогических подходов в частности. Но это не означает, что произошел отказ от предыдущих методик преподавания: если учесть многолетний накопленный преподавателями опыт и совместить его с современными методиками, то можно достичь значительных

положительных результатов. Дискретная математика имеет полный спектр необходимого инструментария и методик для организации такого способа обучения.

Например, раньше при организации командного подхода в обучении дискретной математики практически не возможно было использовать компьютер. Это значительно уменьшало выбор тематик и методов реализации этого подхода. Также значительно труднее было организовать обратную связь со студентами. Однако в нынешнее время таких проблем не возникает, так как компьютер уже есть практически у каждого.

Самыми распространенными способами проверки уровня знаний и умений студентов является текущий и итоговый контроль. Немаловажным способом оценки уровня подготовленности студентов является повседневное наблюдение за их учебной работой. Этот подход позволяет преподавателю составить представление о том, как ведут себя учащиеся на занятиях; как они воспринимают и осмысливают изучаемый материал, в какой мере они проявляют сообразительность и самостоятельность при выработке практических умений и навыков, каковы их учебные склонности, интересы и способности, степень их усидчивости и систематичности в овладении знаниями. Это позволяет определять индивидуальные особенности учащихся, учитывать их в работе и, следовательно, более объективно подходить к проверке и оценке знаний студентов [1].

Однако, кроме традиционных методов оценки уровня подготовки студентов возможно применение нетрадиционных методов, таких как проектная деятельность, проблемное обучение, командная работа.

В рамках дисциплины «Дискретная математика» возможно организовать контроль успеваемости студентов на основе выполнения ими междисциплинарной командной проектной деятельности. Данный вид контроля позволит выявить уровень знаний, приобретенных студентами за период изучения дисциплины, а также сформировать навыки работы в коллективе, приучить к самообразованию и ответственности. Применение междисциплинарного подхода к проведению контроля позволяет выявить умения применять полученные знания при изучении других предметов.

В качестве задания для организации командной деятельности можно взять тему «Применение теории графов в программировании». Группу необходимо разделить таким образом, чтобы части ее не сильно отличались по успеваемости. Также желательно учитывать взаимоотношения студентов в группе, так как возможны ситуации, когда между учащимися может быть личная неприязнь. По количеству студентов группа не должна быть слишком маленькой, но также не должна быть и слишком

ком большой, поскольку в этом случае произвести оценку знаний каждого участника группы будет тяжело. Оптимальным количеством студентов в группах является 4–5 человек. После формирования состава групп каждая из них должна выбрать определённый алгоритм, основанный на теории графов и реализовать его применение в какой-либо произвольной программе. Внутри групп должны быть разделены обязанности между каждым её участником. В группе обязательно должен быть лидер, который назначается преподавателем и должен отчитываться перед ним за проделанную группой работу. Остальные роли могут быть разделены самими участниками между собой. Например, в группе может быть один участник ответственный за программирование, несколько тестируемых, которые производят проверку составленной программы на ошибки, участник, ответственный за сбор и анализ методического материала по выбранному группой алгоритму.

После истечения времени, отведённого преподавателем на выполнение проекта, каждая группа проводит защиту своей программы, в которую входит презентация готовой программы, а также приводятся доказательства и аргументация, почему группа реализовала именно такой алгоритм. После защиты составленные группами программы должны быть записаны на диски и сданы преподавателю вместе с составленными отчётами о проделанной работе, причём каждый студент составляет свой собственный отчёт, в котором он указывает объём проделанной им работы, а также приводит оценку работы своих одноклассников. На основании отчетов о проделанной работе и ее открытой защиты перед всей группой преподавателем выставляется итоговая оценка. Данный пример организации проектной деятельности можно использовать для оценки знаний студентов по разделу «Теория графов» и применять в качестве результата итогового контроля.

Таким образом, «Дискретная математика» – это фундаментальная дисциплина, изучение которой является необходимой основой для дальнейшего обучения студентов специальности «Информатика». Для специальности «Информатика» дискретная математика необходима как инструментальный, с помощью которого можно наглядно изучить принципы работы и эксплуатации всех современных компьютеров, а также автоматизированных систем управления и проектирования. Умения и навыки, приобретенные на этапе изучения данной дисциплины, облегчают понимание и изучение других предметов.

Литература

1. Педагогика: педагогические теории, системы, технологии : учебное пособие / Под ред. С. А. Смирнова. – М. : Академия, 2000. – 512 с.

МЕТОДИЧЕСКИЕ АСПЕКТЫ ПРЕПОДАВАНИЯ ДИСЦИПЛИНЫ «ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

З. С. Сейдаметова^{1а}, В. А. Темненко^{2б}

¹ Украина, г. Симферополь, Крымский инженерно-педагогический университет

² Украина, г. Симферополь, Таврический национальный университет им. В. И. Вернадского

^а z.seidametova@acm.org

^б valery.temnenko@gmail.com

Подготовка инженеров-программистов в университетах представляет собой многоэтапный процесс. В самом начале обучения студентов необходимо научить студентов основам программирования, ознакомить с одним или несколькими языками программирования (C/C++, Python, Java и т.п.). Этот этап – первый, и его можно назвать «основанием». На следующем, втором этапе студенты должны начать изучать: а) основы системного программирования, это обеспечивается цепочкой дисциплин: «Специализированные языки программирования» ⇒ «Операционные системы» ⇒ «Системное программирование»; б) основы разработки баз данных и инструментарий для реализации баз данных; в) алгоритмы и структуры данных, а также реализацию классических алгоритмов сортировки, выбора и структур данных с помощью изученных ранее языков программирования.

Третий этап включает изучение дисциплин, связанных с разработкой приложений для Интернет, применением прикладного программного обеспечения, различных платформ программирования. После прохождения предыдущих этапов можно приступить к изучению технологии разработки программного обеспечения – планирования работы над проектом, формулировки требований, методологий разработки, проектирования, кодирования, обеспечения качества продукта, сопровождения, сертифицирования, лицензирования.

При разработке программных приложений (software) необходимо учитывать, что программный продукт должен быть высококачественным, быстрым и экономичным по отношению к ресурсам системы, обеспечивать быстрый и безопасный обмен данными. Поэтому очень важно при подготовке инженеров-программистов обучать студентов владению современными методами разработки программного обеспечения как для настольного компьютера, использования в сети, в Интернет, так и для запуска на мейнфреме (mainframe) или облаке. В монографии [1] описа-

ны современные подходы разработки программных продуктов, а также направления развития программной инженерии. В материалах цюрихской конференции SEAFOOD [2] представлены 13 статей, описывающих изменения индустрии оффшорной и аутсорсинговой разработки программных продуктов, лучшие практики в этой индустрии, измерения и оценки, выявление рисков, разработка спецификаций, а также новые горизонты развития оффшорной программной инженерии.

Одним из важных принципов разработки программного обеспечения является «проектирование, ориентированное на удобство использования и на клиента». Согласно этим принципам необходимо создавать программный продукт, который не просто работает, а с которым удобно работать. Именно этот принцип берется за основу систематического подхода к процессу проектирования. Результатом применения этого принципа является программный продукт, соответствующий реальным запросам клиентов и пользователей. В ставших уже классическими работах [3], [4], [5] авторы обращают внимание именно на эти стороны разработки современных программных продуктов.

В достаточно давней статье [6] «программная инженерия» охарактеризована как множество формализмов, методов и методик для разработки надежных, экономичных программных систем, отвечающих требуемым спецификациям [6, 55]. Эта лапидарная формулировка высказана более полутора десятилетий – огромный срок в IT-сфере, – но она, несомненно, сохраняет ценность и актуальность. Инженеры-разработчики считают, что главной парадигмой программной инженерии является проектирование. Они используют слово «проектирование», чтобы описать текущий процесс, который начинается с определения проблемы, конструирования требований и спецификации разрабатываемой системы. Как инженеры по способу мышления, разработчики программного обеспечения используют проектирование процесса для построения программной системы – т.е. разрабатывают модель жизненного цикла программного обеспечения. Разработчики программного обеспечения стремятся развить систематические методы создания программных продуктов типа «reesstmts» (**R**eliable, **E**conomical, **E**fficient software systems **T**hat **M**eet **T**heir **S**pecifications), т.е. продукта надежного, экономичного, эффективного, который соответствует требованиям и спецификациям. Задача не является простой. Этому необходимо учить студентов.

В рамках учебной дисциплины «Технология разработки программного обеспечения» предполагается научить студентов разрабатывать программные приложения по принятым в сфере компьютеринга стандартам с применением современных средств проектирования, разработки, тестирования, отслеживания версий. Пререквизитами этой дисциплины

являются (рис. 1): «Программирование для начинающих» (1 семестр), «Программирование» (1–3 семестры), «Дискретная математика» (1, 2 семестры), «Базы данных и информационные системы» (3-4 семестры), «Программирование на языке Python» (4 семестр), «Программирование на языке Java» (5 семестр), «Алгоритмы и структуры данных» (6 семестр), «Объектно-ориентированное проектирование» (6 семестр), «Основы математического моделирования и системный анализ» (7 семестр).



Рис. 1. Пререквизиты дисциплины «Технология разработки программного обеспечения»

Наш педагогический опыт преподавания дисциплины «Технология разработки программного обеспечения» позволяет сформулировать следующий список таких задач:

- продемонстрировать необходимость разработки программного обеспечения, ориентированного на практическое использование;
- указать ключевые принципы разработки графического интерфейса пользователя GUI (Graphic User Interface);
- показать влияние фундаментальных принципов проектирования на структуру графического интерфейса пользователя;
- обсудить критерии оценки программного обеспечения;
- описать стандарты разработки программных продуктов;
- представить свойства проектирования «хорошего» программного обеспечения;
- сопоставить объектно-ориентированный анализ и проектирование с

подходами структурного анализа и проектирования программного обеспечения;

- показать, как оценить качество проектов на основе ключевых принципов и концепций проектирования;
- научить выбирать и применять шаблоны проектирования, подходящие для разработки приложения;
- показать, как оценить архитектуру программного проекта, программный продукт на уровне компонент, а также проект с точки зрения повторного использования;
- показать, как выбрать и обосновать набор инструментальных средств для поддержки программных продуктов;
- продемонстрировать навыки в использовании инструментария для поддержки процесса разработки программного обеспечения;
- поэтапно описать жизненный цикл программного обеспечения;
- показать, как выбрать релевантную модель разработки программного продукта;
- показать, как применить ключевые элементы и типовые методы выявления и анализа требования для построения набора требований к программной системе;
- научить понимать отличия между различными типами и уровнями тестирования программных продуктов;
- изучить основные принципы создания команды разработчиков и управления коллективом;
- научить создавать проектный план для проекта разработки программного продукта, включающий оценку размера и трудозатрат, календарный график проекта, распределение ресурсов, управление конфигурацией, управление изменениями, а также выявление рисков, связанных с проектом и управление ими.

Развитие современных коммуникационных технологий и аппаратного обеспечения изменили не только жизнь людей, общества, но и внесли перемены в организацию взаимоотношений сотрудников, а также в значительной степени изменили образовательные технологии и образование, подтолкнули к созданию новой педагогики, педагогических подходов.

Несомненно, коммуникационные инновации повлияли и продолжают влиять на способы, методики обучения. Однако технологии не заменяют преподавателей, они дают инструментарий, который интенсифицирует процесс приобретения и передачи знаний.

В таблице 1 представлены навыки, характерные для каждого уровня традиционной таксономии Блума [7, 236-242], [8], предложенной в 1956 году. На низших уровнях находятся знание и понимание, на более высо-

ких – применение, умение анализировать, создавать и оценивать. Успешность достижения конечного педагогического результата зависит от прочности усвоения студентом каждого уровня. Чем выше уровень усвоения, тем глубже знания, приобретенные студентом. Для достижения в процессе обучения высоких уровней таксономии Блума обычно применяется проектный метод.

Таблица 1

Уровни традиционной таксономии Блума

Уровень	Навыки и типы заданий
Знание	Повторение, воспроизведение или распознавание информации. Типы заданий: дать определение, повторить, перечислить, назвать, рассказать, акцентировать, выделить, показать.
Понимание	Схватывание (понимание) смысла информационных материалов, выделение и объяснение важной информации. Типы заданий: описать, объяснить, определить признаки, формулировать по-другому, обсудить, распознавать, выразить, обнаруживать, сообщить, рецензировать.
Использование	Применение в сходной ситуации, решение проблем. Типы заданий: применить, проиллюстрировать, решить, интерпретировать, употреблять, действовать, использовать, демонстрировать, инсценировать, разработать план, описать в общих чертах.
Анализ	Определить части и структуру. Типы заданий: проанализировать, проверить, провести эксперимент, организовать, сравнить, выявить различия, найти ошибки и упущения в логике, оценить значимость фактов.
Синтез	Соединить части по-новому. Типы заданий: создать, придумать дизайн, разработать, составить план, составить схемы.
Оценка	Оценить значимость на основе критериев. Типы заданий: представить аргументы, защитить точку зрения, доказать, спрогнозировать, оценить значимость того или иного продукта деятельности.

В рамках дисциплины «Технология разработки программного обеспечения» предполагается, что студенты овладевают материалом на более высоком уровне таксономии Блума. Поэтому обязательным условием дисциплины является разработка индивидуального проекта по заданной теме с использованием инструментария разработки проектов. Такой подход позволяет достигать следующих уровней таксономии Блума –

использование, анализ и синтез.

В наступившем двадцать первом веке с учетом изменений, произошедших в педагогических подходах, была представлена обновленная версия таксономии Блума [9], [10] (см. таблицу 2). Сравнение традиционной и обновленной версий таксономии Блума показывает изменение названий категорий, содержание навыков и типов заданий.

Таблица 2

Обновленная версия таксономии Блума

Уровень	Навыки и типы заданий
Запоминание	<p>Может ли студент повторить или запомнить материал занятия?</p> <p>Типы заданий: формулировать, преумножать, перечислить, запомнить, вспомнить, повторить, воспроизводить состояние.</p>
Понимание	<p>Может ли студент объяснить идеи или концепции?</p> <p>Типы заданий: классифицировать, описывать, обсуждать, объяснять, идентифицировать, обнаруживать, распознавать, информировать, выбирать, переводить, перефразировать.</p>
Применение	<p>Может ли студент применить полученную на занятиях информацию в новом направлении, новым способом?</p> <p>Типы заданий: выбирать, применить, инсценировать, иллюстрировать, размещать, интерпретировать, разрабатывать, разработать план, решать, использовать, записывать.</p>
Анализирование	<p>Может ли студент выделить различия между разными частями?</p> <p>Типы заданий: оценивать, сравнивать, сопоставлять, критиковать, дифференцировать, различать, выделять, исследовать, экспериментировать, задавать вопросы, тестировать.</p>
Оценивание	<p>Может ли студент обосновать свою позицию или решение?</p> <p>Типы заданий: оценивать, спорить, защитить точку зрения, обосновывать, выбирать, поддерживать, оценить значимость того или иного продукта деятельности.</p>
Создание	<p>Может ли студент создать новый продукт или точку зрения?</p> <p>Типы заданий: составлять, строить, создавать, проектировать, разрабатывать, формулировать, писать.</p>

Использование на занятиях по дисциплине «Технология разработки программного обеспечения» обновленной таксономии Блума позволяет студентам изучать теоретический и практический материал на уровнях «Анализирование», «Оценивание» и «Создание».

Представляет интерес дальнейшее изучение целей обучения дисци-

плины «Технология разработки программного обеспечения» с точки зрения двухмерной таксономии Блума [10]. Одна размерность – структурно-знаниевая, вторая – уровневая. Структурно-знаниевая размерность подразделяется на четыре компоненты – фактуальные, концептуальные, процедурные и метакогнитивные знания. Уровневая размерность содержит шесть компонент – запоминание, понимание, применение, анализирование, оценивание и создание. Наложение двухмерной таксономии Блума на материал дисциплины позволяет более тонко настраивать его для обучения студентов.

Представляет также самостоятельный интерес изучение временных изменений таксономии Блума за последние полстолетия.

Литература

1. The Future of Software Engineering [1st Edition]. / Sebastian Nanz, ed. – Springer, 2011. – 186 p.
2. Software Engineering Approaches for Offshore and Outsourced Development / Olly Gotel, Mathai Joseph, Bertrand Meyer (Editors). – 3rd International Conference, SEAFOOD, Zurich, Switzerland, July 2-3, 2009. – Springer, 2009. – 137 p.
3. Брауде Э. Технология разработки программного обеспечения / Э. Брауде. – СПб. : Питер, 2004. – 655 с.
4. Braude E. J. Software Engineering: Modern Approaches : [2nd Edition] / E. J. Braude, M. E. Bernstein. – John Wiley & Sons, 2011. – 800 p.
5. Кватрани Т. Визуальное моделирование с помощью IBM® Rational® Software Architect and UML : пер. с англ. / Т. Кватрани, Дж. Палистрант. – М. : КУДИЦ-ПИРЕСС, 2007. – 192 с.
6. Denning P. J. A Discipline of Software Architecture / P. J. Denning, P. A. Dargan // Interactions. Volume 1, Issue 1, Jan. 1994. – ACM Press: New York. – P. 55–65.
7. Сейдаметова З. С. Подготовка инженеров-программистов по специальности «Информатика» : [монография] / З. С. Сейдаметова. – Симферополь : Крымучпедгиз, 2007. – 480, [1] с.
8. Bloom B. S. Affective Domain: The Classification of Educational Goals (Taxonomy of Educational Objectives) / D. R. Krathwohl, B. S. Bloom. – Longman Schools Division, 1965. – 214 p.
9. Bloom B. S. A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives / L. W. Anderson, D. R. Krathwohl, B. S. Bloom. – Longman, 2001. – 352 p.
10. Churches A. Bloom's digital taxonomy [Electronic resources] / A. Churches, 2009. – 75 p. – Mode of access : <http://edorigami.wikispaces.com/>

ПРОЕКТУВАННЯ КОМБІНОВАНОГО НАВЧАННЯ СИСТЕМНОГО ПРОГРАМУВАННЯ БАКАЛАВРІВ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

А. М. Стрюк

Україна, м. Кривий Ріг, Криворізький національний університет
andrey.n.stryuk@gmail.com

Сутність педагогічного проектування визначається в діяльності викладача ВНЗ як спрямованість на відбір змісту навчальної діяльності з метою формування у студентів предметних спеціальних знань і вмінь, на розробку ефективних технологій, діагностування результатів проектної педагогічної діяльності на основі об'єктивних критеріїв готовності майбутніх фахівців до професійної діяльності [6].

В. І. Гриценко визначає педагогічне проектування як «розробку засобів подання навчального матеріалу, вибір телекомунікаційних та інформаційних технологій і формування навчальних стратегій ... з урахуванням цілей навчання і стилю вивчення потенціальної аудиторії» [5, 156]. Нашою задачею є проектування комбінованого навчання системного програмування бакалаврів програмної інженерії. Під комбінованим навчаннями ми розуміємо спосіб реалізації змісту навчання, що інтегрує аудиторну та позааудиторну навчальну діяльність за умови педагогічно виваженого поєднання технологій традиційного, електронного, дистанційного та мобільного навчання з метою ефективного досягнення навчальних цілей.

Реалізація комбінованого навчання потребує розробки комплексу навчально-методичних матеріалів, що включають в себе робочу програму курсу, навчальні та інструктивні матеріали, перелік практичних, лабораторних робіт і методичних порад щодо їх виконання, план проведення та тематику дискусій, семінарів та інших форм навчальної діяльності, передбачену курсом.

В сучасній педагогічній практиці існує багато підходів до проектування та розробки навчальних матеріалів та систем навчання. Широке розповсюдження отримала модель розробки систем навчання ADDIE (Analysis, Design, Development, Implementation, and Evaluation) [4], згідно якої розробка та використання навчальних матеріалів має складатися з п'яти етапів: *аналізу* – визначення цілей навчання і завдання, які має виконати той, хто навчається, щоб продемонструвати та застосувати отримані знання та навички; *проектування* – конкретизації цілей і завдань окремих розділів курсу, планування занять, визначення структури навчальних матеріалів та засоби, що будуть використовуватись під час

навчання; *розробка* – створення необхідних навчальних матеріалів, інтеграція засобів ІКТ, друк або розміщуються в системі управління навчанням; *реалізація* – безпосередньо здійснення навчального процесу за розробленою програмою з використанням створених матеріалів; *оцінка* – аналіз результатів навчання, за результатами якого вносяться зміни до відповідних навчальних матеріалів та планів.

Р. Райзер та Дж. Демпсі уточнюють модель ADDIE і зазначають, що «область педагогічного проектування ... включає в себе аналіз навчальних та практичних задач, проектування, розробку, впровадження, оцінювання та управління навчальними та супровідними процесами та ресурсами, що залучені для поліпшення навчального процесу» [3, 5]. Дж. Кемп [2] підкреслює, що процес навчання є неперервним циклом, що потребує постійного планування, проектування, розробки і оцінки і пропонує модель, що нараховує дев'ять етапів: аналіз цілей, аналіз аудиторії, аналіз завдань, деталізація цілей, структурування матеріалу, розробка плану, розкладу, реалізація навчання та оцінювання. Подібні етапи виділяє і О. Е. Коваленко [7] в функціональній схемі управління процесом навчання (рис. 1), яка більш системно відображає зв'язки між різними етапами процесу навчання, а також підкреслює важливу роль технологічної складової, що передбачає обґрунтований вибір засобів навчання, а також можливість застосування додаткових засобів, якщо процес навчання потребує корегування.

Перш за все, сформулюємо загальну ціль навчання: сформувати у студентів бакалаврату з програмної інженерії компетенції зі створення системного програмного забезпечення та використання системних виликів та сервісів операційних систем при розробці прикладного програмного забезпечення. Аналіз Галузевого стандарту вищої освіти, SWEBOOK [1] та типових посадових інструкцій інженера програміста і системного програміста дозволив визначити дії, пов'язані з системним програмуванням, які бакалавр програмної інженерії має виконувати під час практичної діяльності:

- проектування програмного забезпечення;
- аналіз, проектування та прототипування людино-машинного інтерфейсу;
- розробка алгоритмів та структур даних для програмних продуктів;
- на основі аналізу математичних моделей і алгоритмів розробка програми, її тестування та налагодження;
- визначення обсягу, структури, схеми введення, опрацювання, збереження і виведення даних, що підлягають обробці засобами обчислювальної техніки;
- конструювання інструментального програмного забезпечення для

розробки системного та прикладного програмного забезпечення (компіляторів, текстових процесорів, оболонок операційних систем);

- конструювання операційних систем та їх оточення;
- використання системних викликів та сервісів операційних систем та їх оточення для розробки нового системного програмного забезпечення;
- забезпечення захищеності програм і даних від несанкціонованих дій;
- верифікація програмного забезпечення;
- розробка інструкцій з використання програм, оформлення необхідної технічної документації.

Аналіз виробничої діячливості дозволив конкретизувати цілі навчання системного програмування і на їх основі розробити комплекс практичних завдань з курсу програмної інженерії (таблиця 1).

Таблиця 1

Структура практичних завдань з дисципліни «Системне програмування»

Етапи практичного завдання	Практична діяльність
На основі загального формулювання задачі створити технічне завдання на розробку програми	проектування програмного забезпечення
Розробити алгоритм програми та відобразити його у вигляді блок-схеми	розробка алгоритмів та структур даних для програмних продуктів
Самостійно обрати формати введення початкових даних та виведення результатів	визначення обсягу, структури, схеми введення, опрацювання, збереження і виведення даних, що підлягають обробці засобами обчислювальної техніки
Розробити програму згідно отриманому завданню та розробленому алгоритму	конструювання інструментального програмного забезпечення для розробки системного та прикладного програмного забезпечення (компіляторів, текстових процесорів, оболонок операційних систем); конструювання операційних систем та їх оточення; використання системних викликів та сервісів операційних систем та їх оточення для розробки нового системного програмного забезпечення; забезпечення захищеності програм і даних від несанкціонованих дій.

Етапи практичного завдання	Практична діяльність
Протестувати розроблене програмне забезпечення	верифікація програмного забезпечення
Підготувати звіт з виконаної практичної роботи	розробка інструкцій з використання програм, оформлення необхідної технічної документації

Також на стадії проектування були визначені методи навчання, що будуть застосовані під час вивчення окремих тематичних розділів системного програмування. Більшість з цих методів орієнтовані в першу чергу на практичну діяльність майбутніх інженерів-програмістів. Д. Меррілл зазначав, що навчання полегшується, якщо той, хто навчається, займається вирішенням реальних проблем; коли існуючі знання є основою для нових знань; коли той, хто навчається, демонструє і застосовує нові знання; коли нові знання інтегруються у світогляд того, хто навчається [8]. На основі зазначених принципів навчання має складатися з чотирьох основних етапів: активація попереднього досвіду; демонстрація навичок; застосування навичок; інтеграція навичок для вирішення реальних практичних завдань. Ці ідеї найкраще узгоджуються з технологією контекстного навчання, яка дозволяє гармонійно поєднати в собі методи проектів, проблемного навчання, та навчання у співпраці [9]. Саме комбінація цих методів дозволяє найбільш повно розкрити всі компетенції з системного програмування під час навчання.

Для вивчення кожного тематичного розділу з системного програмування з метою активізації пізнавальної діяльності студентів використовуються елементи проблемного навчання, яке передбачає формулювання проблемної ситуації, вирішення якої потребує від того, хто навчається самостійно шукати шляхи її вирішення. Під час виконання практичних завдань, пов'язаних зі сформульованою проблемою, підключаються методи проектів, парного програмування та навчання у співпраці, що дозволяє додатково сформувати необхідні комунікативні професійні навички. У зв'язку з цим, було розроблено наступний план вивчення кожного тематичного розділу:

1. Визначення цілей заняття.
2. Розгляд плану заняття.
3. Постановка проблеми для розв'язання.
4. Генерація ідей, активація вже отриманих знань та власного досвіду.
5. Узагальнення ідей, що були отримані під час дискусії та розгляд правил і методів, що можна реалізувати на базі цих ідей.

6. Демонстрація практичного застосування розглянутих правил і методів

7. Розв'язання практичних задач в рамках індивідуального або групового проекту.

8. Перевірка засвоєння теоретичного матеріалу.

9. Аналіз, оцінки, висновки.

Деякі етапи такого плану, наприклад, пошукову діяльність пов'язану з генерацією, обговоренням та узагальненням ідей, важко реалізувати в рамках традиційних форм організації навчального процесу – лекції, практичному або лабораторному занятті – у зв'язку з часовими обмеженням, що мають ці форми занять. Реалізувати ці етапи можна за рахунок більш продуктивного використання часу, що відведено студентам на самостійну роботу, із залученням засобів ІКТ. Найбільш системне використання засобів ІКТ у навчальному процесі вітчизняних ВНЗ досягається за рахунок застосування системи управління комбінованим навчанням, в якості якої може виступати LMS або LCMS, що має достатньо розвинені інструменти комунікації, подання навчальних матеріалів та управління навчальним процесом з урахуванням його комбінованого характеру. Особливості організації роботи студента під час вивчення дисципліни «Системне програмування» з використання системи управління комбінованим навчанням відображено в моделі на рис. 1.

Дана модель передбачає організаційний етап, на якому студент має ознайомитись з особливостями роботи з системою, зареєструватися в системі, отримати доступ до навчальних матеріалів та комунікаційних засобів, що будуть використовуватися під час навчання. На вступному занятті студентам пропонується перевірити навички роботи з системою, після чого можна перейти до вивчення тематичних розділів, яке відбувається за наступним планом: спочатку студентам оголошуються цілі навчання та формулюється проблема, яку належить вирішити в рамках даного розділу.

Оголошення цілей та задач розділу відбувається за допомогою розсилання повідомлень. Після цього починається пошуковий етап, на якому активізується попередній досвід студентів, виконується самостійний пошук методів та засобів вирішення поставленої проблеми, обговорюються та узагальнюються ідеї та обрані методи вирішення задач. На цьому етапі спілкування між студентами та викладачем відбувається за допомогою комунікаційних засобів системи.

Під час пошуку додаткової інформації студентами використовуються навчальні та інші матеріали, частина яких може бути розміщена в системі. Під час лекційного заняття викладач підводить підсумки пошукового етапу, викладає основні теоретичні відомості, демонструє приклади

вирішення поставленої задачі. Після лекційного заняття студентам пропонуються практичні завдання у вигляді індивідуального або колективного проекту, які вони виконують під час лабораторних робіт.

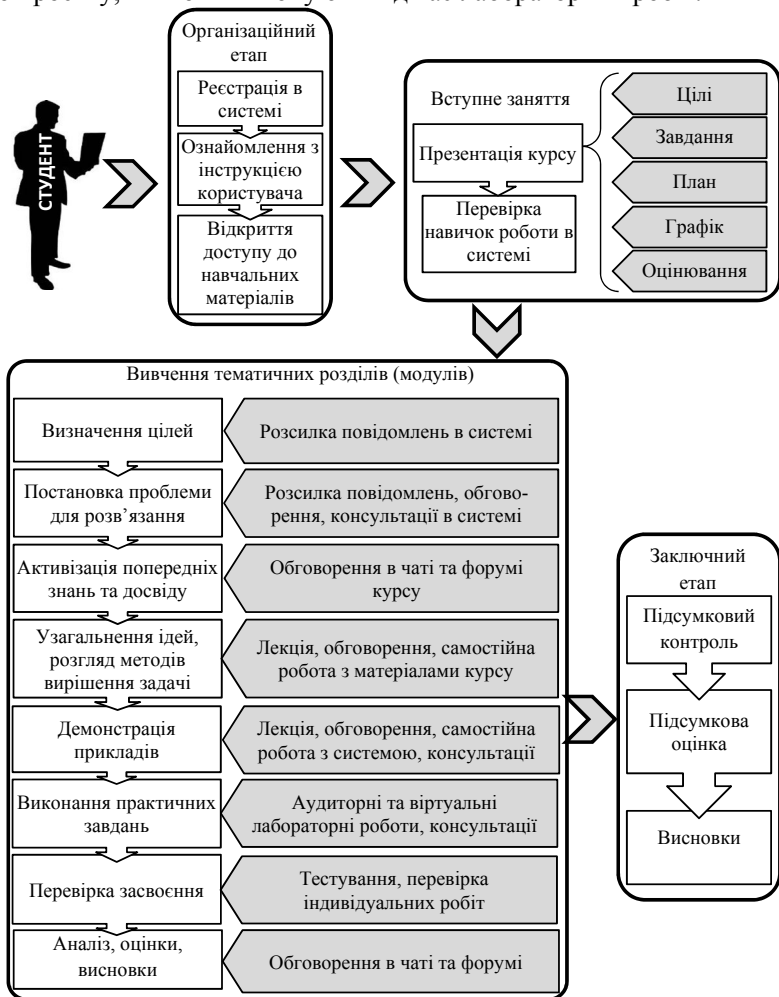


Рис. 1. Модель роботи студента під час вивчення дисципліни «Системне програмування» з використання системи управління комбінованим навчанням

Лабораторні роботи можуть виконуватися у спеціально обладнаних аудиторіях, або поза ними з використанням спеціалізованих віртуальних лабораторій у системі. Звіти з виконаної лабораторної роботи подаються

на перевірку також з використанням відповідних засобів системи. Після успішного виконання практичних робіт, студенти за допомогою тестування студенти перевіряють рівень засвоєння теоретичних знань.

Завершується вивчення розділу підведенням підсумків, аналізом отриманих результатів та їх обговоренням, що відбувається за допомогою комунікаційних засобів системи.

Результати проектування та розроблена модель роботи студента була використана в Криворізькому національному університеті під час навчання системного програмування студентів спеціальності «Програмне забезпечення автоматизованих систем» з використанням системи управління комбінованим навчанням «Агапа». Але узагальнення результатів проектування дозволяє адаптувати подібну модель до будь якої іншої професійно-орієнтованої дисципліни.

Література

1. Guide to the Software Engineering Body of Knowledge / SWEBOK. A project of the IEEE Computer Society Professional Practices Committee ; Eds. Alain Abran, James W. Moore. – IEEE, 2004. – 202 p.
2. Morrison G. R. Designing effective instruction / Gary R. Morrison, Steven M. Ross, Howard K. Kalman, Jerrold E. Kemp. – 6th edition. New York : John Wiley & Sons, 2009. – 491 p.
3. Reiser R. A. Trends and Issues in Instructional Design and Technology. Third edition / Robert A. Reiser, John V. Dempsey. – Boston : Pearson, 2012. – 408 p.
4. Strickland A. W. College of Education – ADDIE [Electronic resource] / A. W. Strickland. – Idaho State University. – Mode of access : <http://ed.isu.edu/addie/index.html>
5. Дистанционное обучение: теория и практика / В. И. Гриценко, С. П. Кудрявцева, В. В. Колос, Е. В. Варенич. – К. : Наукова думка, 2004. – 376 с.
6. Колесникова И. А. Педагогическое проектирование : учеб. пособие для высш. учеб. заведений / И. А. Колесникова, М. П. Горчакова-Сибирская. – М. : Академия, 2005. – 288 с.
7. Коваленко Е. Э. Методика профессионального обучения : учебник / Коваленко Елена Эдуардовна. – Харьков : Штрих, 2003. – 480 с.
8. Merrill M. D. First Principles of Instruction / M. David Merrill // Educational Technology Research and Development. – 2002. – Vol. 50. – Iss. 3. – P. 43–59.
9. Современные образовательные технологии : учебное пособие / под редакцией академика РАО Н. В. Бордовской. – второе издание, стереотипное. – М. : КНОРУС, 2011. – 432 с.

ЦІЛІ ТА ЗМІСТ НАВЧАННЯ ОБ'ЄКТНО-ОРІЄНТОВАНОГО МОДЕЛЮВАННЯ СТУДЕНТІВ ПРИРОДНИЧИХ СПЕЦІАЛЬНОСТЕЙ ПЕДАГОГІЧНИХ УНІВЕРСИТЕТІВ

О. І. Теплицький

Україна, м. Кривий Ріг, Криворізький національний університет
teplitsky5@yandex.ru

Вибір об'єктів моделювання в інформатиці як науці залежить від предметної області, а в інформатиці як навчальній дисципліні – від фахової орієнтації майбутнього вчителя. Сьогодні в Україні підготовка вчителів інформатики здійснюється переважно на природничих спеціальностях «Фізика», «Математика», «Хімія» (зі спеціалізацією «Інформатика»), в яких метод моделювання є провідним методом дослідження, тому в процесі проектування змісту курсу об'єктно-орієнтованого моделювання ми виходили з професійно-орієнтованої функції фундаменталізації інформатичної освіти, що має наступні структурні компоненти: цільовий, змістовий, технологічний та підсумковий.

Враховуючи, що головною метою інформатичної підготовки студентів є формування професійних інформатичних компетентностей, основою *цільової компоненти* обрано суспільне замовлення, державні стандарти вищої освіти та особистий вибір студента. Змістова компонента містить специфічну інформатичну теорію, що відображає професіоналізацію обраної спеціальності.

Зміст навчання є тим стрижнем, який з'єднує всі рівні системи освіти, визначаючи їхню послідовність та наступність. При формуванні змісту важливо встановити баланс між фундаментальністю та професійною спрямованістю інформатичної підготовки, реалізувавши виділений Г. О. Михаліним принцип диференційованої фундаментальності [1].

Так, для майбутніх вчителів математики доцільним є комп'ютерне моделювання математичних об'єктів, для майбутніх вчителів фізики доцільним є комп'ютерне моделювання фізичних об'єктів та процесів, для майбутніх вчителів біології доцільним є комп'ютерне моделювання екологічних процесів, а при навчанні майбутніх вчителів хімії приділити більше уваги квантово-механічним моделям атомів та молекул і т.п. Спільним у всіх випадках є використання таких засобів об'єктно-орієнтованого моделювання, що надають можливість конструювати об'єкти та встановлювати зв'язки між ними, досліджувати явища, процеси, динаміку об'єктів, важкодоступних для спостереження в реальному світі, візуалізуючи рухомі елементи, найбільш важливі з погляду навчальних цілей і завдань характеристики досліджуваних об'єктів і про-

цесів.

Технологічна компонента реалізується як відбір засобів, форм та методів розв'язання задачі фундаменталізації інформатичної освіти. *Підсумкова компонента* для методичної системи навчання об'єктно-орієнтованого моделювання є діагностичною та вказує на рівень сформованості професійних інформатичних компетентностей студентів. Вона набуває свого специфічного вираження в *модельному стилі мислення*. Будемо говорити, що студент має модельний стиль мислення, якщо він може:

- а) структурувати інформацію про об'єкт у просторі та часі;
- б) визначати логічну структуру моделі, створювати графічні образи елементарних явищ, що становлять процес;
- в) виявляти основні зміни стану об'єкта або процесу;
- г) представляти взаємодію об'єктів і процесів у просторі й часі.

Традиційною моделлю методичної системи навчання є п'ятикомпонентна модель, запропонована А. М. Пишкало [2], в якій використовується системний підхід стосовно компонентів процесу навчання (всі компоненти утворюють єдине ціле із визначеними внутрішніми зв'язками). Згідно з цією моделлю, методична система навчання – це сукупність ієрархічно пов'язаних компонентів: цілей навчання, змісту, методів, засобів і форм організації навчання.

«Можна говорити про те, що поява принципово нових засобів навчання, що якісно змінюють можливості передавання інформації і розширюють можливості організації навчального процесу, приводить до перегляду змісту, форм і методів навчання і може опосередковано позначитися на цілях навчання» [4, 7]. Це зауваження майже на 10 років випередило появу комп'ютерів у масовій школі, але з позицій сьогодення, можна стверджувати, що в ньому сконцентровані всі основні ідеї створення й обґрунтування методичної системи навчання інформатики: комп'ютер як потужний сучасний дидактичний засіб у значній мірі обумовлює цілі, зміст, методи й форми організації навчання в сучасній школі.

Розробка повноцінної методичної системи навчання об'єктно-орієнтованого моделювання відіграє ключову роль у її функціонуванні як суттєвої складової інформатичного блоку підготовки вчителів природничих спеціальностей. Тому актуальним є аналіз її компонентів, виявлення найбільш слабких місць і проблем, що здатні помітно погіршити її якості і без подолання яких неможливий її подальший розвиток.

Враховуючи особливості науки інформатики (а, відповідно, і навчальної дисципліни «Об'єктно-орієнтоване моделювання»), в структурі методичної системи навчання було виділено технологічну підсистему,

що надало можливість максимально відобразити взаємовпливи всіх її компонентів: цільового, змістового та технологічного.

Створюючи методичну систему навчання об'єктно-орієнтованого моделювання, ми намагалися:

- врахувати професійну спрямованість підготовки студентів природничих спеціальностей шляхом диференціації змісту навчання;
- спрогнозувати результати педагогічного впливу, передбачаючи, які компетентності з моделювання має набути студент, який розвиваючий вплив на нього повинен здійснити зміст навчання;
- забезпечити варіативність форм організації, методів і засобів навчання з опорою на дослідницький підхід у навчанні.

Виходячи з визначеної структури, визначають цільовий, змістовий та технологічний компоненти методичної системи навчання.

Мета (ціль) навчання – ідеальне передбачення кінцевих результатів навчання; те, до чого прагнуть студенти, викладачі. За традиційним підходом до визначення процесу навчання через знання, уміння та навички, він переслідує три основні групи взаємопов'язаних цілей: 1) освітня – формування у студентів наукових знань, спеціальних і загальнонавчальних умінь і навичок; 2) розвивальна – розвиток мови, мислення, пам'яті, творчих здібностей, рухової та сенсорної систем; 3) виховна – формування світогляду, моралі, естетичної культури тощо.

Головною метою нашої методичної системи є формування компетентності в об'єктно-орієнтованому моделюванні через сукупність спеціальних знань, умінь та навичок, що забезпечують студентам можливість застосовувати засоби обчислювальної техніки спочатку у навчальній, а в перспективі й у професійній дослідницькій діяльності. Зміст курсу містить сукупність двох взаємопов'язаних складових: теоретичної та практичної. Теоретична складова спрямована на формування у студентів модельного мислення, навичок об'єктно-орієнтованого аналізу предметної області та проектування об'єктно-орієнтованої моделі, на ознайомлення з методологією об'єктно-орієнтованого моделювання і особливостями її комп'ютерної реалізації мовою об'єктно-орієнтованого програмування. Практичний аспект пов'язується з набуттям умінь добору середовища об'єктно-орієнтованого моделювання, адекватного розв'язуваній задачі та формі організації навчального дослідження (індивідуального чи колективного), навичок роботи у різних середовищах, проведення обчислювального експерименту та прийняття рішення про адекватність моделі об'єкту дослідження. У загальній структурі курсу об'єктно-орієнтованого моделювання обсяг практичних занять має переважати над обсягом теоретичних.

Цілі навчання об'єктно-орієнтованого моделювання:

- формування навичок об'єктно-орієнтованого моделювання як найбільш природного способу дослідження систем різної складності;
- ознайомлення з основними принципами конструювання та дослідження об'єктно-орієнтованих моделей;
- формування навичок індивідуальних та колективних навчальних досліджень.

Обговорюючи питання змістового компоненту методичної системи навчання, дамо визначення цьому поняттю згідно Закону України «Про вищу освіту» [4]. *Зміст навчання* – структура, зміст і обсяг навчальної інформації, засвоєння якої забезпечує особі можливість здобуття вищої освіти і певної кваліфікації.

Зміст навчання на рівні певної навчальної дисципліни – обумовлена цілями та потребами суспільства система знань, умінь і навичок, професійних, світоглядних і громадянських якостей, що має бути сформована в процесі навчання з урахуванням перспектив розвитку суспільства, науки, техніки, технологій, культури та мистецтва.

Зміст освіти є одним із факторів економічного і соціального прогресу суспільства і повинен бути орієнтований: на забезпечення самовизначення особистості, створення умов для її самореалізації; розвиток суспільства; посилення та вдосконалення правової держави.

Визначення змісту навчання об'єктно-орієнтованого моделювання необхідно здійснювати з урахуванням принципів, спільних для будь-яких навчальних дисциплін, так і властивих насамперед для інформатичних [5, 70]:

1. *Принцип відповідності навчальним цілям.* Цілі навчання об'єктно-орієнтованого моделювання визначаються, виходячи із загальних цілей освіти – формування компетентності особистості, причому не лише в моделюванні, а й у інших спеціальних професійних, загально професійних та ключових компетентностей.

2. *Принцип науковості.* Вимога науковості передбачає взаємозв'язок теорії, розробки, аналізу й оцінювання ефективності, реалізації та застосування моделей. Зміст курсу повинен складатися з тих розділів і тем, які важливі для практики моделювання незалежно від обраного підходу до навчання самого об'єктно-орієнтованого моделювання.

3. *Принцип фундаментальності,* реалізація якого передбачає: опанування методологічно важливих та інваріантних знань, що мають довгий термін життя, необхідних для професійної діяльності вчителя; розвиток творчої і пізнавальної активності та самостійності студентів; розвиток методичних систем навчання інформатичних дисциплін з урахуванням перспектив розвитку «економіки знань» та інформаційного суспільства; системність засвоєння інформатичних дисциплін на основі

глибокого розуміння сучасного стану та існуючих проблем інформатики [5, 86].

4. *Принцип відкритості.* Цей принцип передбачає можливість корекції змісту курсу залежно від освітнього напрямку підготовки, без порушення цілісності фундаментального ядра дисципліни.

5. *Принцип сучасності.* Швидкий розвиток інформаційно-комунікаційних технологій вимагає регулярного перегляду навчальної програми дисципліни з метою модернізації застарілих компонентів, що для майбутніх учителів є особливо актуальним з огляду на особливості їхньої професійної діяльності в умовах широкого впровадження засобів ІКТ у навчальний процес.

6. *Принцип перспективності.* Цей принцип передбачає формування у студентів готовності до подальшого навчання протягом усього життя, що надасть їм можливість бути здатними до вирішення професійних проблем у майбутньому.

7. *Принцип вирівнювання знань.* Зміст курсу об'єктно-орієнтованого моделювання повинен включати модуль, протягом якого буде здійснено початкове опанування середовища та мови об'єктно-орієнтованого моделювання.

Окрім цього, добір змісту навчального матеріалу має здійснюватися з урахуванням основних дидактичних принципів навчання: посиленої складності, системності, послідовності і систематичності навчання, наочності змісту і діяльності, активності і самостійності, свідомості, індивідуалізації і колективності навчання та ін.

Змістова частина методичної системи (навчальний матеріал) включає широкий спектр задач із різних предметних галузей і передбачає опанування технології об'єктно-орієнтованого моделювання у середовищах для навчальних досліджень. Основний зміст навчання об'єктно-орієнтованого моделювання складає конструювання та дослідження динамічних та імітаційних моделей. З урахування принципу вирівнювання знань, зміст курсу складають 3 модулі.

У *першому модулі* «Вступ до об'єктно-орієнтованого моделювання» розглядаються базові поняття та уявлення курсу (поняття про моделювання, види моделей, об'єктно-орієнтоване моделювання; об'єктно-орієнтоване програмування та об'єктно-орієнтовані мови; абстракція, інкапсуляція, спадкування, поліморфізм як основа об'єктно-орієнтованої методології; етапи об'єктно-орієнтованого моделювання: об'єктно-орієнтований аналіз, проектування, обчислювальний експеримент та аналіз його результатів) і виконується огляд середовищ об'єктно-орієнтованого моделювання (зокрема, виділяються універсальні середовища моделювання, середовища для конструювання динамічних моде-

лей та середовища для конструювання імітаційних моделей). На підставі аналізу придатності середовищ для моделювання різних класів моделей пропонується у навчанні студентів спеціальностей «Хімія», «Біологія» та «Фізика»:

а) при розгляді динамічних моделей послуговуватися середовищами об'єктно-орієнтованого моделювання VPython та Squeak як основними, а Sage, Alice та NetLogo – як додатковими;

б) при розгляді імітаційних моделей скористатися середовищами об'єктно-орієнтованого моделювання Alice та NetLogo як основними, а Sage, VPython та Squeak – як додатковими [6].

У процес навчання об'єктно-орієнтованого моделювання студентів спеціальностей «Математика», «Інформатика» середовище Sage із додаткового переноситься у основні. Це не вимагає додаткових витрат часу через спільність мови ООП у середовищах VPython та Sage [7].

Другий модуль «Об'єктно-орієнтовані динамічні моделі» присвячений розгляду динамічних моделей математичної екології (динаміка одновидової популяції, модель «Хижак-жертва», вікові моделі Леслі), класичної механіки (динаміка коливних систем, рух тіл в полі сили тяжіння, моделювання аеродинамічних об'єктів та явищ), молекулярної фізики і фізики твердого тіла (атомна та молекулярна динаміка) та електродинаміки (рух заряду в електричному та магнітному полях).

У третій модуль «Об'єктно-орієнтовані імітаційні моделі» включені моделі кліткових автоматів (модель поширення чуток, модель «Хижак-жертва», модель поширення пожежі, гра «Життя»), стохастичні моделі (модель броунівського руху, модель відмов обладнання, модель росту кристалу), моделі фрактальних об'єктів та процесів (моделі регулярних фракталів, задача перколяції, моделі електролізу, модель утворення берегової лінії).

Підсумковий контроль знань за курсом – екзамен у формі захисту індивідуальних та колективних дослідницьких проєктів.

Література

1. Михалін Г. О. Професійна підготовка вчителя математики у процесі навчання математичного аналізу / Михалін Г. О. – К. : ДНІТ, 2003. – 320 с.

2. Пышкало А. М. Методическая система обучения геометрии в начальной школе : авторский доклад по монографии «Методика обучения элементам геометрии в начальных классах», представленной на соискание ученой степени докт. пед. наук / А. М. Пышкало ; НИИ СиМО АПН СССР – М., 1975.

3. Преемственность в обучении математике : сб. статей : [пособие

- для учителѐй] / Сост. А. М. Пышкало. – М. : Просвещение, 1978. – 239 с.
4. Закон України «Про вищу освіту» / Верховна Рада України. Інститут законодавства. – К., 2002. – 96 с.
5. Лапчик М. П. Теория и методика обучения информатике : учебник / М. П. Лапчик, И. Г. Семакин, Е. К. Хеннер ; под общей ред. М. П. Лапчика. – М. : Академия, 2008. – 592 с.
6. Теплицький О. І. Об'єктно-орієнтоване моделювання в Alice. Частина 1 / О. І. Теплицький ; за науковою редакцією академіка НАПН України М. І. Жалдака. – К. : НПУ імені М. П. Драгоманова, 2011. – 56 с.
7. Теплицький О. І. Розробка WAP-інтерфейсу до Web-СКМ Sage / О. І. Теплицький, І. І. Ліннік, М. В. Глуходід // Тези доповідей VII Всеукраїнської науково-практичної конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2010). 4-6 травня 2010 року. – Том 2. – Черкаси : Черкаський державний технологічний університет, 2010. – С. 34.

СОЦИАЛЬНОЕ ТВОРЧЕСТВО – МЕТОДИЧЕСКИЙ ПРИЕМ ФОРМИРОВАНИЯ МОТИВАЦИИ ПРИ ИЗУЧЕНИИ ИНФОРМАТИКИ

С. В. Терещенко

Украина, г. Белогорск, Белогорский УВК «Школа-лицей» №2
stereshenko@list.ru

Опыт показывает, что учащиеся достигают большего, чем запланировано, при условии, если они сами могут сформулировать цель изучения предмета. «Для чего это мне понадобится?» – рассматривая этот вопрос на уроках информатики при изучении разных тем программы, учащиеся стараются получить знания и навыки, которые можно применять в повседневной жизни. Привлечение школьников к социальному творчеству на основе участия в проектной деятельности различных общественных организаций является одним из методических приемов мотивации, которая направлена на создание учебных проблемных ситуаций с акцентом на практическом общественно-социальном смысле полученных знаний.

Цель статьи – рассмотреть социальное творчество, как методический прием формирования мотивации при изучении информатики в профильной школе.

Динамика социокультурного развития требует от подрастающего поколения формирования принципиально нового опыта, направленности личности на непрерывное саморазвитие, самосовершенствование, умения прогнозировать и моделировать процесс развития собственной личности, ближайшего окружения и социума в целом, готовности ее к постоянному самостоятельному поиску духовных идеалов и эффективных способов воплощения их в практике.

Социальное творчество – наиболее зрелая форма социальной активности, высшая форма социальной деятельности, созидательный процесс, направленный на преобразование сложившихся и создание качественно новых форм социальных отношений и социальной действительности. Оно предполагает мобилизацию всех интеллектуальных, духовных и физических сил, позволяет личности предельно полно раскрыть свои способности, направить энергию на достижение интересов общества, коллектива [3]. Социальное творчество направлено на самообразование, освоение социально-культурного опыта предшествующих поколений, на формирование культурной среды обитания, а также на совершенствование и создание новых форм общественных отношений. Оно проявляется в самостоятельной и творческой инициативе личности.

Социальное творчество может быть классифицировано по следующим направлениям: историко-культурное; экологическое; моделирование и изучение устройств, принципов их действия; создание программ для компьютеров; естественнонаучное; художественное [6].

Одним из путей формирования качественных знаний, умений и навыков в курсе информатики у каждого школьника является объединение его личных способностей, интересов и потребностей с демонстрацией применения ИКТ в конкретной профессиональной сфере, в частности в его будущей профессиональной деятельности или социально значимой направленности проектной деятельности [5].

Целью современной школы является обучение проектированию, направленному на социальное творчество, как общеучебному универсальному (надпредметному) умению или компетентности (обладание знаниями, позволяющими судить о чем-либо, высказывать мнение).

Часто отождествляют два разных вида деятельности – проектирование и исследование. Исследование выявляет то, что уже существует, проектная деятельность создает новый продукт. То есть проектирование предусматривает планирование, анализ, поиск, реализацию, результат (придумал → спроектировал → осуществил) [4, 78-79].

Социальное творчество можно отнести к практико-ориентированным типам проектов. Результаты деятельности участников четко определены с самого начала, они ориентированы на социальные интересы участников (документ, газета, блог, мероприятие, сайт). При проектировании требуется составление сценария всей деятельности его участников с определением функций каждого из них. Особенно важным является организация работы в виде поэтапного обсуждения и презентации полученных результатов и возможных средств их внедрения в практику.

Портфолио проекта позволяет сформулировать и развить такие жизненные компетентности, которые помогают школьнику организовать и осуществлять процесс самообразования: самостоятельно формулировать проблему; определять цель работы; планировать работу; искать и обрабатывать полученную информацию; критически оценивать собственные действия и давать оценку сделанному; распределять обязанности в группе, обосновывая свой выбор.

В 2009 году в Украине началась активная реализация программы Intel «Шлях до успіху» в рамках общей инициативы Intel и Программы развития ООН. Особенностью проекта в Украине является ориентация на создание условий для эффективной интеграции молодежи, применение информационно-коммуникационных технологий как инструмента развития её социальной активности, создания благоприятной среды для реализации социальных проектов [2].

Программа «Шлях до успіху» направлена на развитие у школьников навыков, необходимых во взрослой жизни, чтоб быть успешным в условиях экономики знаний XXI столетия. Особенностью проекта является его комплексный характер. Учащиеся обучаются не только информационным технологиям, но и получают навыки критического мышления, кооперативного обучения и проектной деятельности, которые являются основами личностно-ориентированного обучения, которое широко используется в международной практике. Все задания и проекты, которые в процессе обучения выполняют учащиеся, ориентированы на потребности и проблемы местных сообществ – школы, района, города – и направлено на активное привлечение детей в жизнь сообщества, поиска путей его развития и усовершенствования. Результатами работы по данной программе в нашем лицее является активное участие учащихся во многих социальных, экологических проектах: «Мы против торговли людьми», «Диалог поколений», «Климатическое движение 350 – Вместе двигаем планету», «Кешингуем по Белогорску», «Час Земли», Экологический фестиваль «Планета людей», «Эко-Арт», «Крым – Буковина – Транзит», «Украина в Европе» и т.д. Учащиеся применяют свои знания полученные на уроках информатики: оформление при помощи текстового редактора тренинг-планов и грантовых заявок, создание при помощи издательской системы школьной газеты, расчет средствами табличного процессора бюджета мероприятий и акций, разработка презентаций о результатах своей деятельности, ведение дневников-блогов.

Участие в учебном телекоммуникационном проекте «Крым – Буковина – Транзит» позволяет в процессе разработки и проведения исследований в своих местных сообществах школьным инициативным группам самоорганизоваться, пропустить полученную информацию через собственное восприятие, развить свои аналитические и поисковые навыки. При реализации таких проектов существует возможность открытия новых, неизвестных ранее аспектов местной истории, а также уменьшает расстояния между учителями и учащимися через неформальное общение в процессе исследования. Для работы над такими проектами учащиеся должны: владеть основами информационной культуры; знать и владеть основными исследовательскими методами; владеть коммуникативными навыками; уметь самостоятельно интегрировать полученные знания разных учебных предметов для решения познавательных задач, которые необходимо решать в рамках телекоммуникационного проекта.

Проектная деятельность, в том числе и социальной направленности, должна иметь письменную часть – отчет о ходе работы (начиная с определения проблемы проекта), все принимавшиеся решения с их обоснованием, все возникшие проблемы и способы их преодоления; анализи-

руются собранная информация, наблюдения, приводятся результаты опросов и т.п.; подводятся итоги, делаются выводы, выясняются перспективы проекта. Для создания такого отчета учащимся необходимы знания по работе с текстовым и табличным процессором, со средствами обработки мультимедийных данных, умения работы со средствами персональной и коллективной коммуникации [1, 47-48].

Непременным условием проекта является его публичная защита или проведение запланированного мероприятия (мероприятие – результат проекта). В ходе презентации автор не только рассказывает о ходе работы и показывает её результаты, но демонстрирует собственные знания и опыт в решении проблемы проекта, приобретенную компетентность. Элемент самопрезентации – важная сторона работы над проектом, которая предполагает рефлексивную оценку автором всей проделанной им работы и приобретенного в её ходе опыта.

Каждый участник телекоммуникационного проекта выполняет свои задачи:

- учитель информатики – осуществление сеансов связи и контактов с другими участниками проекта из других регионов; организация получения, печать и распространение материалов, полученных по электронной почте; помощь учащимся в создании школьных веб-страниц; организация хранения материалов;

- учащиеся – пишут и отправляют письма по электронной почте, «скачивают» информацию из сети, структурируют полученную информацию, работают в электронных конференциях, размещают там собственную информацию и т. д.

Наиболее острой проблемой при участии в таких проектах, является проблема разного стартового уровня знаний, умений и навыков школьников по информатике. Часть школьников на момент начала изучения информатики, уже имеет определенный опыт использования ИКТ в бытовой и учебной деятельности. Для другой части школьников обучение необходимо начинать с основных понятий, определений, с формирования элементарных навыков работы с компьютером. Кроме того, знания детей с высоким уровнем подготовки, как правило, неполные и несистематизированные, а поэтому существует необходимость корректировки и пополнение знаний у таких учащихся. Поэтому участие в социальных проектах учащихся является основным приемом формирования мотивации в изучении предмета, так как именно практическая направленность такой деятельности отвечает на основной вопрос «Для чего это мне понадобится?», когда рабочая группа проекта видит значимость результатов своей деятельности на практике в своем родном городе или селе (выпуск Укрпочтой марки «Белогорск» в рамках реализации проекта

«Приглашаем в Белогорск», інформаційний сайт «Кешингуем по Белогорску», шкільна газета «Лицеист»).

Выводы. Таким образом, работа с социальными, телекоммуникационными проектами – требование времени, так как в современной области человеческой деятельности используются средства ИКТ и телекоммуникации. Поэтому развитие этих компетенций – актуальная проблема современной школы.

При работе с проектами социальной направленности в ходе обучения информатике в урочное и внеурочное время по-новому понимается коммуникативный ресурс. Это, во-первых, возрастание роли личностного коммуникативного ресурса ученика, его способность работать в команде. Во-вторых, это возможность привлечения к решению социальной, учебной задачи или проблемы учеников данного возраста, учащихся других возрастов, школ. Это расширение социального партнерства в области образования.

Меняется понимание результата — им становится компетентность в решении учебных и личностных задач на основе развития опыта социальной, учебной и метапознавательной деятельности.

Литература

1. Забарна А. П. Організація навчання інформатики у профільній школі / А. П. Забарна. – Тернопіль : Мандрівець, 2009. – 128 с.
2. Герман М. Програма Intel® «Шлях до успіху» : методичний посібник для викладачів [Електронний ресурс] / Марк Герман, Шелли Шот. – 2008. – 81 с. – Режим доступу : <http://www.eduwiki.uran.net.ua/wiki/images/1/17/StaffManual1.pdf>
3. Национальная социологическая энциклопедия [Электронный ресурс]. – Режим доступа : <http://voluntary.ru>
4. Садкіна В. І. 101 цікава педагогічна ідея. Як зробити урок / В. І. Садкіна – Харків : Основа, 2009. – 88 с. – (Золота педагогічна колекція)
5. Сейдаметова С. М. Формування інформаційної культури та інформаційної компетентності школярів // С. Сейдаметова, С. Терещенко / Сучасні тенденції розвитку інформаційних технологій в науці, освіті та економіці : всеукр. науч.-прак. конф., 9-11 квіт. 2009 р. : тези доп. – Львів, 2009. – С. 275-277.
6. Енциклопедія педагогічних технологій та інновацій / Автор-укладач Н. П. Наволокова. – Харків : Основа, 2009. – 88 с. – (Золота педагогічна скарбниця).

ЕВОЛЮЦІЯ ТА СУЧАСНИЙ СТАН ПІДГОТОВКИ ФАХІВЦІВ З ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ У ВНЗ УКРАЇНИ

Н. А. Хараджян

Україна, м. Кривий Ріг, Криворізький національний університет
nata_leonova@mail.ru

Соціально-економічні зміни, які відбуваються в Україні, сприяють підвищенню вимог до якості підготовки фахівців з економіки, зміни змісту самого процесу навчання.

Підготовка фахівців з економічної кібернетики у вищих навчальних закладах України почалась з 60-х років минулого століття, паралельно із розвитком нових напрямів в економічній науці. Вперше термін «економічна кібернетика» з'явився в працях С. Біра (Великобританія), Х. Греневського (Польща) та В. С. Немчинова (СРСР). Вони ж окреслили й основні напрями розвитку цієї науки, приділивши особливу увагу взаємозв'язку системного аналізу економіки з теорією регулювання, логікою та теорією інформації. Однак багато фундаментальних теоретичних положень економічної кібернетики було сформульовано значно раніше: уявлення про економіку як цілісну і взаємопов'язану систему міститься вже в економічній таблиці Франсуа Кене (1758 р.); воно було розширене і науково обґрунтоване в працях класиків теорії і практики управління.

Зараз підготовку фахівців з економічної кібернетики здійснюють понад 30 вищих навчальних закладів України, лише в Дніпропетровській області – 11. Так, в Кривому Розі (населення близько 800 тис. мешканців) підготовку фахівців з економічної кібернетики здійснюють 6 ВНЗ, 4 учбових центри, 2 філіали та один відокремлений підрозділ [1]. Значна кількість випускників підвищує значення конкурентоспроможності на ринку праці, що в свою чергу підвищує потребу до фундаментальної підготовки фахівців з економічної кібернетики.

Згідно з наказом Міністерства освіти і науки України № 444 від 07.06.2006 р. підготовка фахівців повинна відбуватися за галузевим стандартом вищої освіти України освітньо-професійної програми (ОПП) підготовки бакалаврів за спеціальностями напряму 0501 – «Економіка і підприємництво», до яких входить спеціальність «Економічна кібернетика» [2]. Стандарт є основою для розроблення вищими навчальними закладами варіативної компоненти ОПП з урахуванням специфіки галузі, регіону, побажань замовника та інтересів студентів [3]. Цей стандарт визначає:

– нормативний зміст навчання у вищих навчальних закладах, обсяги і рівень його засвоювання у процесі підготовки відповідно до вимог

освітньо-кваліфікаційних характеристик (ОКХ) бакалаврів;

- перелік нормативних навчальних дисциплін;
- рекомендований перелік вибіркових дисциплін;
- форми державної атестації [3].

Реалізація даного стандарту в кожному конкретному навчальному закладі може бути різною.

Нормативна частина ОПП охоплює перелік дисциплін, які є обов'язковими для вивчення за такими циклами:

- дисципліни гуманітарної підготовки (які призначені для формування світоглядних, соціально-комунікативних та психолого-педагогічних компетенцій випускників);
- дисципліни природничонаукової та загальноекономічної підготовки (формують загальноекономічні компетенції майбутніх фахівців);
- дисципліни професійної підготовки за спеціальністю (призначені для формування професійних компетенцій і забезпечують теоретичну підготовку та здобуття практичних умінь і навичок за спеціальністю).

Крім переліку дисциплін, стандарт також визначає загальний обсяг навчального часу та його розподіл за дисциплінами.

Варіативна частина ОПП призначена для забезпечення індивідуалізації фахової підготовки майбутніх фахівців відповідно до Болонського процесу та світових тенденцій у вищій освіті. Розподіл дисциплін навчального плану підготовки спеціалістів освітньо-кваліфікаційного рівня «бакалавр» в Черкаському національному університеті ім. Б. Хмельницького наведено в таблиці 1.

Таблиця 1

Розподіл дисциплін навчального плану підготовки спеціалістів освітньо-кваліфікаційного рівня «бакалавр» в Черкаському національному університеті ім. Б. Хмельницького

Цикли дисциплін	Нормативна кількість навчальних годин / кредитів ECTS	В тому числі	
		нормативні дисципліни, годин / кредитів ECTS	дисципліни за вибором (варіативна частина), годин / кредитів ECTS
Гуманітарна підготовка	2160/60	1836/51	324/9
Природничонаукова та загальноекономічна підготовка	4320/120	3600/100 (83%)	720/20 (17%)
Професійна підготовка	2160/60	1296/36 (60%)	864/24 (40%)
Всього	8640/240	6723/187 (78%)	1908/53 (22%)

Останнім часом у вищих навчальних закладах відчутна тенденція до зменшення частки аудиторних занять з фундаментальних дисциплін. Цю тенденцію можна прослідкувати, порівнюючи навчальні і робочі плани, наприклад, з математики (вища математика, теорія ймовірності та математична статистика) спеціальності «Економічна кібернетика» за останні роки (табл. 2). Загальна кількість годин, на перший погляд, збільшується, в той же час стрімко збільшується кількість годин, відведених навчальними планами на самостійне опрацювання матеріалу. Впровадження модульно-рейтингової і кредитно-модульної системи передбачає фактичне скорочення аудиторних занять і перенесення вивчення матеріалу на самостійне опрацювання. Виникає суперечність між переходом освіти на новий рівень, де самостійна робота студентів має займати більший відсоток часу, ніж аудиторна, і неспроможністю студентів самостійно опрацьовувати теоретичний матеріал. Це призводить до погіршення фундаментальної підготовки фахівців, зокрема з економічної кібернетики. Це, в свою чергу, призводить до зниження професіоналізму фахівця – систему професійних знань і вмінь, сукупність якостей спеціаліста (інтелектуальні і ділові здібності, організаційні знання, вміння і навички, можливість швидко переорієнтуватись). Очевидно, що без глибокого знання основ науки, системи знань, всебічного розвитку особистості досягти цього вкрай важко. Таку проблему можна розв'язати за допомогою розширення фундаментальної підготовки фахівців.

Таблиця 2

Розподіл годин на математичні дисципліни в різні роки

Рік затвердження навчального плану	Всього, год.	Аудиторна		Самостійна	
		год.	%	год.	%
2006	324	160	49%	164	51%
2008	360	144	40%	177	60%
2010	432	174	30%	258	70%

Спеціальність «Економічна кібернетика» готує фахівців з поглибленими знаннями сучасних інформаційних технологій, математичних методів моделювання та аналізу, ґрунтовних економічних знань. Випускники спеціальності «Економічна кібернетика» можуть здійснювати діяльність з:

- аналізу і прогнозування фінансово-економічних ситуацій;
- управління економічними об'єктами в ринкових умовах;
- проектування, розробки, супроводження і використання сучасних комп'ютерних систем управління виробництвом, капіталом та ринком.

Згідно з ОКХ, фахівці з економічної кібернетики на своїх посадах

можуть виконувати такі завдання, що безпосередньо пов'язані з моделюванням:

- збирати, систематизувати й аналізувати інформацію про систему управління будь-яким господарським об'єктом;

- аналізувати виробничо-господарську діяльність, інформаційні зв'язки між адміністративними, виробничими та іншими процесами з метою встановлення критеріїв і обмежень під час управління цими процесами;

- аналізувати функціонування діючої інформаційної системи і визначати можливості її модернізації;

- досліджувати й аналізувати інформаційні потреби та проблеми користувачів;

- досліджувати й аналізувати інформаційні потоки на об'єкті;

- аналізувати алгоритми розв'язання економічних задач/комплексів, вибирати та обґрунтовувати ті з них, які адекватні умовам конкретного об'єкта;

- аналізувати результати тестування технічних і програмних засобів передавання, оброблення та збереження інформації з подальшим їх використанням для поліпшення роботи інформаційної системи;

- складати й аналізувати економіко-математичні моделі об'єктів і систем управління;

- аналізувати та моделювати економічний ризик.

Тому для підготовки таких фахівців у ВНЗ в навчальних планах згідно з галузевим стандартом запроваджені такі нормативні навчальні дисципліни: у циклі природничонаукової та загальноекономічної підготовки – економіко-математичне моделювання; у циклі професійної підготовки – моделювання економіки; у циклі природничонаукової та загальноекономічної підготовки (вибірковий цикл на 20 кредитів) – економіко-математичні моделі в управлінні та економіці; у циклі професійної підготовки (вибірковий цикл на 24 кредитів) або математичні моделі синергетичної економіки або/та топологія економічних структур: аналіз та моделювання.

Як бачимо з переліку предметів, майбутні фахівці з економічної кібернетики протягом навчання у ВНЗ вчать застосовувати методологію моделювання в аналізі складних економічних систем.

Згідно з галузевим стандартом вищої освіти України ОПП підготовки бакалаврів із спеціальності «Економічна кібернетика», студенти починають вивчати моделювання, не вивчаючи самих середовищ для моделювання. В ОПП наводиться анотації змісту нормативних дисциплін, з яких видно, що в жодній дисципліні не розглядаються середовища для моделювання. Тому для підвищення рівня професійної підготовки фахі-

вців з економічної кібернетики засобами комп'ютерного моделювання пропонується в рамках дисципліни «Економічна інформатика» ввести теми для вивчення технології та засобів комп'ютерного моделювання за авторською програмою [4]. Курс «Економічної інформатики» традиційно відкриває підготовку студентів-економістів у галузі інформатики та аналізу соціально-економічних систем і виконує, крім іншого, функції курсу вступу до спеціальності.

Аналізуючи вимоги до інформатичної підготовки економістів і вимоги до рівня підготовки, сформульовані в ОКХ, можна зробити висновок, що фахівець економічного профілю повинен:

- уміти систематизувати і узагальнювати інформацію, готувати довідки й огляди з питань професійної діяльності, редагувати, рецензувати, складати реферати статей;

- володіти комп'ютерними методами збирання, зберігання й опрацювання відомостей, що використовуються у сфері професійної діяльності, використовувати сучасні інформаційні технології;

- знати не менше двох систем опрацювання даних: автономну мікрокомп'ютерну і розгалужену системи;

- уміти отримувати доступ і вести пошук інформації в мережових базах даних, користуватися електронною поштою, текстовим процесором, електронними таблицями, системою управління базами даних.

Аналіз інформаційної діяльності економістів дозволив визначити основні інформаційні задачі, що доводиться вирішувати фахівцю в процесі професійної діяльності:

- опрацювання ділової інформації (економічної, статистичної, поданої у цифровій формі);

- підготовка документів на основі опрацьованих даних, що містять таблиці, діаграми, графіки тощо;

- прийняття рішень на основі аналізу фактичних даних, складання прогнозів;

- пошук необхідної для прийняття рішень інформації в будь-яких доступних джерелах, у тому числі й за допомогою інформаційно-комунікаційних технологій;

- обмін даними на відстані.

Саме тому актуальними стають соціальна та професійна мобільність, здатність адаптуватись до змінних умов та вимог, активність, ініціативність, самостійність, прагнення до самоосвіти, висока відповідальність. Цілком очевидно, що вказані якості повинні ґрунтуватися на професійній компетентності фахівця, яка, в свою чергу, може бути забезпечена лише на основі формування в нього фундаментальних знань та всебічної фахової підготовки.

Література

1. Виевская М. Г. Состояние системы подготовки кадров в высших экономических учебных заведениях Украины / М. Г. Виевская, Л. И. Красовская // Новости науки и технологии. – 2008. – № 3. – С. 10–18.
2. Освітньо-професійна програма перепідготовки спеціаліста напряму 0501 – «Економіка і підприємництво» / Кол. авт. під заг. керів. А. Ф. Павленка. – К. : КНЕУ, 2003. – 134 с.
3. Освітньо-кваліфікаційні характеристики бакалавра, спеціаліста і магістра спеціальності «Економічна кібернетика» напряму 0501 – «Економіка і підприємництво» / Кол. авт. під заг. керівн. А. Ф. Павленка. – К. : КНЕУ імені Вадима Гетьмана, 2004. – 55 с.
4. Хараджян Н. А. Педагогічні умови підготовки фахівців з економічної кібернетики засобами комп'ютерного моделювання : дис. ... на здобуття наук ступеня канд. пед. наук : 13.00.04 – теорія та методика професійної освіти / Хараджян Наталя Анатоліївна ; Черкаський національний університет імені Б. Хмельницького. – Черкаси, 2011. – 287 с.

АППРОКСИМАЦИЯ ФУНКЦИЙ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ

С. Н. Чаплыгина

Украина, г. Днепропетровск, Приднепровская государственная академия
строительства и архитектуры
marinapn@ukr.net

Нейронные сети можно рассматривать как современные вычислительные системы, которые преобразуют информацию по образу процессов, происходящих в мозгу человека.

Базовым элементом нервной системы является нервная клетка, называемая нейроном. Каждый нейрон имеет несколько отростков, по которым в него поступает информация, и один выходной отросток, по которому он может передавать импульс нескольким другим нейронам.

Искусственный нейрон (рис. 1) имитирует поведение биологического нейрона выполнением простой операции взвешенного суммирования входных сигналов. Каждое соединение имеет коэффициент w_i , показывающий фактор веса, и определяющий влияние соседнего нейрона на рассматриваемый. Состояние нейрона определяется выражением

$\sum_{i=1}^n u_i w_i$, где u_i – входные сигналы, приходящие от соседних нейронов.

Полученная взвешенная сумма входных сигналов преобразуется в выход нейрона с помощью активационной функции $\sigma(x)$.

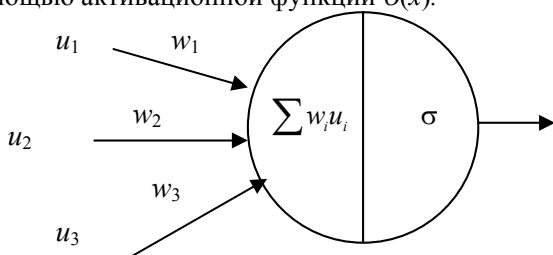


Рис. 1. Искусственный нейрон

В настоящее время создано несколько типов нейронных сетей. Одним из наиболее широко используемых типов являются многослойные сети прямого распространения, которые позволяют создавать сложные нелинейные связи между входными данными и результатами на выходе.

Входные ячейки u_1 и u_2 просто задают входные значения, скрытые ячейки u_3 и u_4 и выходная ячейка u_5 эту информацию преобразуют (рис. 2).

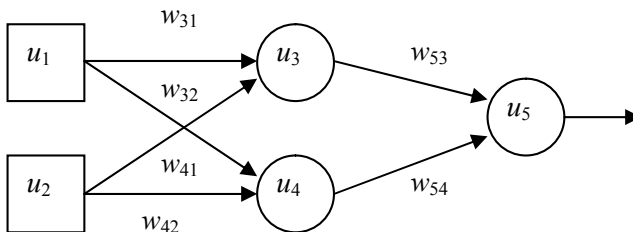


Рис. 2. Сеть с одним скрытым слоем

Одна из главных особенностей нейронных сетей заключается в том, что они предполагают наличие правил, с помощью которых сети могут программироваться автоматически – обучаться. Целью обучения является изменение весовых значений таким образом, чтобы в результате получить требуемые характеристики поведения сети. Для обучения необходимо иметь наборы входных данных, выходы для которых известны. Обычно в начале обучения весовые коэффициенты устанавливаются равными случайным малым значениям, так что в первый раз при предъявлении сети учебного набора оказывается маловероятным, чтобы сеть произвела верный вывод. Расхождение между тем, что даст сеть, и тем, что для данного учебного набора должно быть получено, составляет ошибку, которая может быть использована для корректировки весов. Примером правила коррекции является дельта-правило. Если вывод сети обозначить y , а правильное решение – C , то ошибка составляет $\delta=C-y$. Если сигнал, приходящий к выходной ячейке, обозначить x , то, в соответствии с дельта-правилом, коррекция веса определяется формулой

$$\Delta w = \eta \delta x,$$

где η – действительное число, называемое коэффициентом обучения. Он минимизирует процент изменения, которое может произойти с весом. Новый весовой коэффициент будет

$$w^* = w + \Delta w.$$

В процессе обучения на вход сети подается набор за набором учебных данных, и в результате их обработки весовые коэффициенты корректируются до тех пор, пока для всех вводимых данных ошибки не станут меньше требуемой точности.

Существует несколько алгоритмов обучения нейронных сетей. Одним из них является алгоритм обратного распространения ошибки. Согласно этому алгоритму, погрешности распространяются по сети от выходного слоя через скрытые к входному. В соответствии с этим алгоритмом сначала рассчитываются погрешности в выходных ячейках

$$\delta_{0i} = (C_i - y_i) y_i (1 - y_i),$$

а затем для всех скрытых ячеек

$$\delta_j = \left(\sum_{k=1}^n w_{kj} \delta_{0k} \right) u_j (1 - u_j).$$

Здесь n – количество выходных ячеек, u_j – выходное значение j -той ячейки. После расчета всех погрешностей веса обновляются соответственно соотношениям

$$w_{ij}^* = w_{ij} + \eta \delta_{0i} u_j$$

для весов соединений между скрытым слоем и выходным;

$$w_{ij}^* = w_{ij} + \eta \delta_i u_j^{\text{ex}}$$

для весов между скрытым слоем и входным. Последнее соотношение используется также для обновления весов между скрытыми слоями, если их несколько, в этом случае u_j^{ex} – входящий сигнал в соответствующую ячейку.

С обновленными весами производится расчет для следующего набора данных и таким образом перебираются все N учебных наборов. Суммарная погрешность определяется как

$$E = \sum_{k=1}^N E_k, \quad E_k = \frac{1}{2} \sum_{i=1}^n (C_i - y_i)^2.$$

Если $E > E_0$, весь процесс для всех наборов N повторяется.

Нейронные сети нашли применение в таких областях, как аппроксимация функций, распознавание образов, обработка визуальной информации, медицинские приложения, оптимизация, управление, экспертные системы и других.

В данном случае нейронные сети применялись к задаче аппроксимации функций.

Доказано, что любая непрерывная нелинейная функция может быть сколь угодно хорошо аппроксимирована на ограниченном множестве многослойной сетью, включающей в себя один или более скрытых слоев. Пусть $f(\mathbf{x})$ – непрерывная функция. Для сети прямого распространения с фиксированным числом нейронов и с заданной архитектурой веса представляются вектором \mathbf{w} . Тогда выход, реализуемый сетью, может быть представлен функцией $F(\mathbf{x}, \mathbf{w})$. В процессе обучения производится поиск такого вектора \mathbf{w} , который обеспечивает наилучшую аппроксимацию функции $f(\mathbf{x})$, то есть необходимо выбрать вектор весов таким образом, чтобы выход $F(\mathbf{x}, \mathbf{w})$ был близок к желаемому результату $f(\mathbf{x})$.

Учебные задачи аппроксимации проводились для функций, являющихся комбинациями тригонометрических функций. В качестве активационной функции использовался сигмоид. При этом исследовались такие параметры, как число итераций, число ячеек в скрытом слое и коли-

чество слоев. Рассматривались задачи аппроксимации функций двух переменных, заданных таблично.

В заключение можно отметить следующее:

- успех применения нейронных сетей обычно требует множества экспериментов, то есть нужно иметь достаточно много наборов данных для обучения сети;
- качество работы нейронной сети сильно зависит от предъявляемого ей в процессе обучения набора данных; они должны быть типичными для задачи, решению которой обучается сеть;
- обучение часто оказывается уникальным процессом, когда приемлемые решения могут быть получены только в процессе многочисленных экспериментов;
- для каждого нового типа задач нет универсального способа нахождения решения.

Таким образом, применение нейронных сетей является экспериментальным подходом к прикладным вопросам.

Литература

1. Каллан Р. Основные концепции нейронных сетей / Роберт Каллан. – М. : Вильямс, 2001. – 290 с.

СТВОРЕННЯ ЕЛЕКТРОННОГО ПІДРУЧНИКА З МОВИ ПРОГРАМУВАННЯ QBasic

С. В. Шаров

Україна, м. Мелітополь, Мелітопольський державний педагогічний
університет імені Богдана Хмельницького
seg_sh@ukr.net

Актуальність. В умовах інформатизації суспільства програмування як вид навчальної діяльності залишається важливою складовою підготовки майбутніх фахівців у галузі комп'ютерних технологій та системних наук. Якість підготовки студентів з програмування багато у чому залежить від накопичення досвіду з розробки комп'ютерних програм, у процесі якого студенти ознайомлюються з алгоритмами їх створення, інструментальним середовищем програмування, особливостями типів даних тощо. Більшою мірою студенти опановують знаннями та навичками програмування під час самостійної роботи, на яку витрачається значний час. Тому актуальною є проблема підвищення якості самостійної роботи студентів при опануванні знаннями та вдосконаленні навичками з програмування.

Аналіз останніх досліджень і публікацій. Значний вплив на організацію самостійної роботи студентів має використання інформаційних технологій, які в цьому аспекті розглядалися багатьма дослідниками. Одним з напрямів інтенсифікації навчального процесу, зокрема самостійної роботи при вивченні студентами програмування, є використання електронних підручників, які є потужним дидактичним засобом у процесі підготовки конкурентоспроможних фахівців.

Метою статті є визначення значення електронного підручника у навчальному процесі та основних вимог до їх створення, повідомлення про структуру електронного підручника з мови програмування Qbasic.

Виклад основного матеріалу. Сучасний стан навчально-виховного процесу вищої та середньої освіти характеризується активним впровадженням і використанням нових інформаційних технологій, зокрема програмних засобів навчального призначення. Одним із таких засобів є електронний підручник (ЕП), методично-правильне використання якого дозволяє отримати такі переваги: оперативне оновлення та актуалізація навчального матеріалу; використання різних видів сприйняття інформації (слухове, зорове, емоційне) дозволяє значно полегшити розуміння та запам'ятовування студентами найбільш істотних понять, термінів тощо; моделювання та візуалізація навчальної інформації про існуючі або змодельовані об'єкти та явища; оптимізація діяльності викладача за рахунок

передачі деяких функцій комп'ютеру; організація зворотного зв'язку за допомогою різних видів комп'ютерного контролю знань [2; 5].

Аналіз літературних джерел та власний практичний досвід дозволили виявити, що більшість електронних продуктів, які створюються у значній кількості та називаються «електронні підручники», за своєю суттю є електронними копіями друкованих видань, в яких не застосовуються широкі комп'ютерні можливості подачі навчального матеріалу та контролю знань. Тому є потреба у визначенні основних вимог, які висувуються до створення ЕП. Слід зазначити, що загальноприйнятого визначення поняття «електронний підручник» немає, однак багато дослідників розглядають його в комплексі та визначають це поняття як систему компонентів, призначених для навчання та контролю знань. Так, ЕП вважається: педагогічний програмний засіб, призначений, в першу чергу, для пред'явлення нової інформації, яка доповнює друкарські видання та служить для групового або індивідуального навчання та контролю знань і умінь [2]; навчальні програмні системи комплексного призначення, що надають теоретичний матеріал, контроль рівня знань, інформаційно-пошукову діяльність, математичне моделювання, комп'ютерну візуалізацію та сервісні функції [3] та ін.

ЕП повинні відповідати певним вимогам. Ефективність використання ЕП залежить від того, наскільки він виконує довідково-інформаційні, пояснюючі та контролюючі функції, тобто ті функції, які властиві звичайному підручнику та викладачу. Основна мета використання ЕП полягає у формуванні у студентів необхідних знань, умінь та навичок. Аналіз праць щодо використання ЕП та власний практичний досвід дають підстави визначити вимоги до їх створення:

1. Інформація за вибраним навчальним курсом повинна бути добре структурована та представляти собою закінчені фрагменти з обмеженим числом нових понять та зручним переглядом ієрархії об'єктів. Повинна бути можливість вибору довільної послідовності розділів, що вивчаються.
2. Зміст підручника повинен повністю відповідати змісту навчальної дисципліни, яка визначається базовою та робочою програмами. ЕП повинен забезпечувати безперервність та повноту процесу навчання за умови здійснення зворотного зв'язку.
3. ЕП повинен мати механізм для швидкого пошуку потрібної інформації у змісті підручника. Частіше для цього використовуються гіперпосилання, що дозволяють скоротити час на пошук необхідної інформації.
4. Повинна забезпечуватися простота використання ЕП у поєднанні з інформаційними освітніми технологіями (мультимедійний проектор,

- інтерактивна дошка). ЕП повинен бути зручний у користуванні.
5. У разі необхідності повинний бути тезаурус термінів, понять, історичних даних тощо, які зустрічаються у змісті підручника.
 6. ЕП повинен мати низькі системні вимоги до використання, можливість застосування на різних комп'ютерних платформах.
 7. Добре розроблений ЕП повинен враховувати індивідуальні особливості студентів. Зміст такого ЕП повинен бути представлений на різних рівнях складності для індивідуалізації навчання [1; 3; 4; 7].

Одними із початкових мов програмування, які вивчаються студентами вищих навчальних закладів на першому та другому курсах, є процедурні мови програмування Turbo Pascal та Qbasic. З метою забезпечення самостійної роботи студентів нами було створено ЕП з програмування на мові Qbasic (рис. 1). Ознайомитися із підручником можна за адресою <http://lib.mdpu.org.ua/e-book/sharov/index.htm>. Зміст ЕП відповідає робочій програмі з курсу «Програмування» та містить інформацію про базові конструкції програмування, завдання для виконання різних рівнів складності, лабораторні роботи, приклади завдань до модульного контролю та інші структурні одиниці.

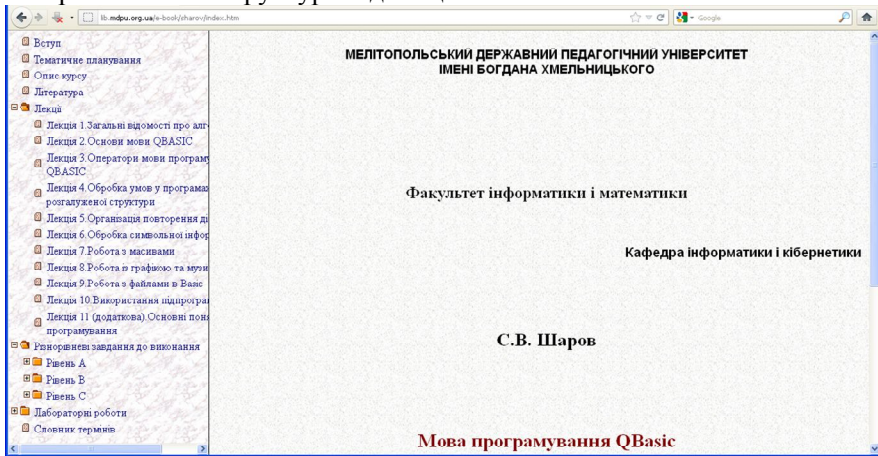


Рис. 1. Головне вікно ЕП

Структура ЕП розпочинається з пояснювальної записки, де студенти можуть познайомитися із метою навчальної дисципліни та її структурою. Далі подано тематичне планування курсу, яке дозволяє користувачу ознайомитися з кількістю годин, які відводяться на вивчення кожної теми. Наступний розділ ЕП містить перелік літератури, яким студенти можуть користуватися під час підготовки до курсу «Програмування».

ЕП з мови програмування Qbasic містить у собі весь лекційний та

практичний матеріал, а також інформацію, необхідну для самостійної роботи. Як зразок, подається приклад екзаменаційного білету з цього курсу. Слід додати, що така структура ЕП є досить зручною для засвоєння студентами навчального змісту. Студенти можуть ознайомитися із змістом тем, зазначених у тематичному плані, виконати завдання з кожної теми, перевірити свої вміння та знання.

Для швидкого та зручного пошуку необхідної інформації у змісті підручника була передбачена гіпертекстова структура. У лекційному навчальному матеріалі присутні гіпертекстові посилання, які дозволяють студенту дізнатися про те чи інше поняття, що присутнє на сторінці термінів та понять. До структури ЕП з програмування на мові Qbasic входить тезаурус, який містить перелік термінів та понять, що використовуються у процесі засвоєння навчальної дисципліни та можуть бути незрозумілими або невідомими для студентів.

Проблема врахування індивідуальних особливостей студентів під час опанування навичками програмування була вирішена за рахунок використання диференційованого підходу у навчанні, а саме створення системи різнорівневих навчальних завдань. Застосування диференційованих завдань у ЕП передбачало врахування таких вимог, як: відповідність умови завдань змісту та меті навчальної дисципліни; наявність достатньої кількості завдань, які відрізняються змістом і рівнем складності; врахування навчальних можливостей студентів під час вибору рівня складності. У цьому випадку поступовий перехід від виконання простих завдань до більш складних дозволить швидко розвинути у студентів відповідні вміння та підвищити їх загальну інформаційну культуру.

Диференційовані завдання було поділено на три рівні складності ("А", "В", "С"). Завдання рівня "А" передбачають вільне володіння студентами програмним та додатковим навчальним матеріалом, наявність розвинутих умінь до програмування, добре сформоване логічне мислення. Робота на рівні "В" передбачає осмислення студентами навчального матеріалу, наявність розвинутих навичок до програмування, застосування знань за розділами програмування, які логічно передують поточній темі. Диференційовані завдання за рівнем складності "С" передбачають наявності у студентів мінімальних знань та умінь програмування і не потребують значних зусиль для їх виконання [6]. Слід зазначити, що під час апробації ЕП складність деяких завдань змінювалася.

Також у підручнику були використані виконані початкові завдання, програмний код яких можна було переглянути за допомогою гіперпосилань у загальному переліку завдань. Найбільше виконаних завдань було представлено на рівні "С", який орієнтований для менш успішних студентів. Це дозволить студентам переглянути стандартні алгоритми ви-

конання поданих завдань та зробити інші завдання за аналогією.

Крім того, у ЕП подані лабораторні роботи за наступними темами: алгоритми та програми лінійної структури; алгоритми та програми розгалуженої структури; алгоритми та програми з повторенням дій; програми обробки рядків; програми обробки двомірних масивів.

Контроль за якістю засвоєння знань студентів є не менш важливою складовою частиною при вивченні програмування, оскільки знання синтаксису конкретної мови програмування допомагає зробити програму найбільш функціональною за короткий час. В ЕП нами були застосовані тестові завдання, які передбачають одну правильну відповідь на декілька запропонованих. Після проходження тестування студенти в змозі переглянути вірні відповіді.

На ЕП з мови програмування Qbasic отримано свідоцтво на авторське право у Державному департаменті інтелектуальної власності (свідоцтво № 40702 від 27 жовтня 2011 р.).

Література

1. Використання сучасних інформаційних технологій при розробці електронних посібників [Електронний ресурс]. – Режим доступу : <http://shpargalka.org.ua/r/2/2016.html>.
2. Гуркова О. М. Электронный учебник как эффективное средство для повышения качества образования [Электронный ресурс] / О. М. Гуркова. – Режим доступа : <http://www.kgau.ru/img/konferenc/2009/115.doc>
3. Зими́на О. В. Рекомендации по созданию электронного ученика [Электронный ресурс] / О. В. Зими́на, А. И. Кирилов. – Режим доступа : http://www.academiaxxi.ru/Meth_Papers/АО_recom_t.htm
4. Классификация электронных средств учебного назначения [Электронный ресурс]. – Режим доступа : <http://www.ido.tsu.ru/ss/?unit=223&page=650>.
5. Левшин М. М. Електронний підручник у системі навчально-методичного забезпечення ВНЗ / М. М. Левшин, Ю. З. Прохур, О. П. Муковіз // Вища освіта України, 2007. – №1. – С. 60–67.
6. Шаров С. В. Використання диференційованих навчальних завдань під час самостійної роботи студентів / С. Шаров // Педагогіка, психологія та медико-біологічні проблеми фізичного виховання і спорту : наукова монографія за редакцією проф. Єрмакова С.С. – Харків : ХДАДМ, 2011. – №3. – С. 151–153.
7. Электронный учебник [Электронный ресурс]. – Режим доступа : <http://www.distance-learning.ru/db/el/212366DD14BD912BC3256C5B0057941A/doc.html>

З ДОСВІДУ ОРГАНІЗАЦІЇ ІНФОРМАТИЧНОЇ ПІДГОТОВКИ МАЙБУТНІХ УЧИТЕЛІВ ІНФОРМАТИКИ ПОЧАТКОВОЇ ШКОЛИ

О. І. Шиман

Україна, м. Бердянськ, Бердянський державний педагогічний
університет

aleksandra.shiman@gmail.com

Постановка проблеми. У країнах зарубіжжя, як і в Україні, до сьогоднішнього дня не вщухали дискусії, розпочаті у 80-х роках ХХ століття, щодо заборони використання комп'ютерів у початковій освіті, мотивовані, перш за все, турботою про здоров'я дітей. Наступний навчальний рік повинен остаточно поставити крапку в цих спорах, бо початкова школа в Україні розпочинає нову дорогу в майбутнє.

Аналіз останніх досліджень і публікацій. Новий стандарт початкової школи, який буде запроваджено вже з 1 вересня 2012 року, носить низку абсолютно нових рішень. Про це під час засідання ІІІ Всеукраїнського з'їзду працівників сказав міністр освіти і науки, молоді та спорту Д. Табачник: «Цілі поставлені вкрай амбітні – іноземна мова з 1 класу, інформатика з 2-го ...». Д. Табачник зазначив, що надзвичайно важливо, аби педагогічні ВНЗ та інститути підвищення кваліфікації оперативно та творчо розпочали підготовку до розбудови нової початкової школи. Також він наголосив, що до кінця поточного року всі вчителі мають ліквідувати свою «комп'ютерну неграмотність»... [1]. За словами міністра, головним у затвердженій Урядом Державній програмі «100 відсотків» на період до 2015 року є підключення всіх шкіл до всевітньої мережі Інтернет, підготовка вчителів математики, інформатики, стовідсоткове завершення комп'ютеризації в середніх загальних закладах України, поступове оновлення парку комп'ютерної техніки.

Проблемам застосування комп'ютерної техніки в освітніх закладах, створення методичної підтримки їх використання присвячені дослідження вчених В. Бикова, А. Гуржія, М. Жалдака, Ю. Жука, В. Лапінського, Ю. Машбиця, В. Монахова, Н. Морзе, І. Підласого та ін. Питаннями підготовки фахівців початкової ланки в умовах інформатизації освіти займаються А. Горячев, С. Гунько, С. Гурьев, Г. Лавренєва, О. Кивлюк, Л. Макаренко, О. Суховірський, В. Шакотько, С. Шапкін та ін. Науковий та методичний супровід впровадження ІКТ у навчально-виховний процес початкової школи здійснюють С. Колесніков, М. Левшин, Г. Ломаковська, Й. Ривкінд, Ф. Ривкінд, В. Шевченко (Україна), О. Горячев, Ю. Первін (Росія), Б. Хантер (США). Дослідженням впливу

комп'ютерних засобів на здоров'я дитини присвячені роботи В. Бондаровської та Н. Польки.

Постановка завдання. Але поставлені міністром завдання вимагають активізації наукових досліджень у напрямку підготовки саме вчителів інформатики початкової школи, тому що введення цього пропедевтичного курсу в початкову ланку ЗНЗ надає проблемі організації процесу формування інформаційно-комунікаційних компетентностей учнів 1-4 класів нових рис і розширює поле наукових розробок. Тож теоретичні й методичні засади підготовки вчителів початкової школи спеціалізації «Інформатика» потребують детального розгляду й активної розробки.

В основу концепції підготовки вчителів інформатики в педагогічних ВНЗ повинні бути покладені інші цілі і завдання навчання інформатики в початковій школі, ніж на інших освітніх рівнях. Професійно-педагогічна діяльність учителя інформатики повинна бути спрямована на розвиток, навчання і виховання молодших школярів як суб'єктів освітнього процесу засобами предмета «Інформатика», а також впровадження нових інформаційно-комунікаційних технологій у навчання і управління навчально-виховним процесом в початковій ланці. Метою професійної підготовки фахівців у галузі інформатики повинна бути пропедевтика розвитку комп'ютерної грамотності учнів початкової школи, що передбачає огляд методів використання комп'ютерів у навчанні (психолого-педагогічних теорій комп'ютерно-орієнтованого навчання, розробки і впровадження навчальних програм, функціонування навчальних середовищ і систем гіпер- та мультимедіа).

Виклад основного матеріалу дослідження. У 2006 році в Бердянському державному педагогічному університеті на факультеті підготовки вчителів початкової школи було здійснено перший набір студентів на спеціалізацію «Інформатика» (спеціальність 6.010102 Початкова освіта). Автору цієї статті, яка в 2005 році захистила кандидатську дисертацію «Формування основ інформаційної культури майбутніх учителів початкової школи» за спеціальністю 13.00.02 – теорія і методика навчання (інформатика) [2], було доручено розробку всіх дисциплін циклу професійно-педагогічної підготовки (його варіативної частини). Тому, насамперед, треба було визначитися з назвами дисциплін, їх часовим навантаженням, змістовим наповненням, послідовністю вивчення. Це непросте завдання вирішувалося протягом останніх 5-ти років, і тепер, після першого випуску студентів (учителів початкової школи та інформатики) можна підводити перші підсумки, ділитися досвідом і окреслювати шляхи подальшого розвитку й удосконалення.

У межах даної публікації хотілося б ознайомити освітянську спіль-

ноту з переліком дисциплін, їх коротким оглядом і розподілом за семестрами в плані навчального процесу підготовки студентів освітньо-кваліфікаційного рівня «бакалавр».

Перш за все, ми спиралися на концепцію нового стандарту вищої освіти, яка заснована на збалансованому включенні в зміст підготовки вчителів інформатики фундаментальних, прикладних і методичних дисциплін, що визначають необхідний професійний рівень сучасного фахівця. Це повинно реалізуватися через наступні змістовні лінії: теоретико-методологічну, програмно-технологічну й предметно-методичну. Засвоєння навчальних дисциплін, що становлять ці лінії, забезпечує формування в майбутнього вчителя інформатики відповідного інформаційного світогляду й необхідного професійного інструментарію, розрахованого на тривалу перспективу й досить інваріантного стосовно можливих локальних змін в області інформатики і комп'ютерної техніки.

1. Головною тенденцією у формуванні змісту теоретико-методологічної лінії є підвищення рівня фундаментальних знань в області наукових дисциплін, досліджуваних у відповідних курсах. До таких курсів відноситься, насамперед, дисципліна «Теоретична інформатика», яка вивчається майбутніми вчителями інформатики початкової школи протягом перших 3-х семестрів.

Теоретична інформатика є світоглядною наукою, тому й предметом її вивчення є одна з вихідних категорій світобудови – інформація. Інформатика – фундаментальна природнича наука, що вивчає загальні властивості інформації, процеси, методи й засоби її обробки. Віднесення інформатики до фундаментальних наук означає, що вона має загальнонаукову значимість, тобто її поняття, закони й методи застосовувані не тільки в рамках самої науки, але й в інших наукових і прикладних дисциплінах. Майбутні вчителі повинні усвідомити фундаментальність і універсальність законів інформатики. У результаті вивчення цієї дисципліни у студентів повинне бути сформоване представлення про загальні проблеми й завдання теоретичної інформатики й про основні принципи й етапи інформаційних процесів в суспільстві та освіті. Вони повинні знати основні методи одержання, зберігання, обробки, передачі й використання навчальної інформації, а також уміти застосовувати для цього апарат комп'ютерних інформаційних систем, найбільше широко використовувані класи інформаційних моделей.

Теоретична інформатика включає наступні змістові лінії: теорію інформації, апаратні та програмні засоби автоматизації обробки інформації, теорію кодування, теорію систем і моделей, теорію алгоритмів, методи технологізації інформаційних процесів. Структурування змісту курсу «Теоретична інформатика» здійснювалося, виходячи із реального

навчального процесу у педагогічному університеті, з підсиленням спрямованості курсу на формування методичної культури студентів, із пріоритету і взаємозв'язку з профільюючими предметами в траєкторії професійного становлення майбутнього вчителя інформатики початкової школи, із постійної рефлексії співвідношення теоретичної підготовки з результатами педагогічної практичної діяльності студентів.

Тож, дисципліна «Теоретична інформатика» – важливий системотвірний компонент методичної системи навчання майбутніх вчителів інформатики в педагогічних університетах, який визначає дидактичні умови цілісного процесу професійного становлення майбутнього педагога. Подальша методична підготовка майбутнього вчителя інформатики базуватиметься на теоретичних основах інформатики та на навичках проектування методичної системи навчання інформатики в школі.

2. Другою за значимістю і найбільшою за обсягом є дисципліна «Використання сучасних інформаційних технологій» (вивчається протягом всього періоду підготовки бакалаврів – 8-ми семестрів). В сьогоденних умовах інформатизації суспільства заклади освіти покликані виховувати особистість, що здатна творчо мислити та приймати нестандартні рішення, швидко здійснювати обробку поточної інформації. Це вимагає змін у підходах і методах навчання. Нововведення полягають у використанні нових педагогічних та інформаційних технологій (ІТ).

У сфері освіти основні завдання ІТ – реалізувати різноманітні форми проведення занять, задіяти сучасні засоби, забезпечити індивідуальний підхід до навчання. У процесі навчання майбутніх учителів інформатики початкової школи поряд з вивченням теоретичних розділів інформаційних технологій більше уваги необхідно приділити тим інформаційним технологіям, які є базовими складовими майбутньої сфери їх професійної діяльності. Причому якість навчання повинна визначатися ступенем закріплених стійких знань, умінь та навичок роботи в середовищах розповсюджених інформаційних технологій при розв'язуванні типових задач педагогічної діяльності.

Зміст курсу «Використання сучасних інформаційних технологій» (ВСІТ) включає сукупність двох взаємопов'язаних компонентів: теоретичного і практичного. В межах визначеної кількості годин потрібно надати більш адаптовані до життєдіяльності в умовах інформатизації освіти теоретичні знання, а практичні навички роботи з програмними пакетами для рішення практичних завдань обробки навчальної інформації спрямувати на майбутню професію педагога початкової ланки. Окреслені напрямки практичного використання комп'ютера як універсального дидактичного засобу навчання передбачають його застосування майбутніми вчителями для підготовки і демонстрації фрагментів навчально-

го матеріалу як у друкованому вигляді, так і в мультимедійному поданні, а також для комп'ютерно-орієнтованої підтримки навчання молодших школярів і контролю за його результатами. В процесі засвоєння курсу студенти мають оволодіти такими інформаційними технологіями:

- опрацювання текстів;
- опрацювання графічних зображень;
- опрацювання числових даних;
- роботи з мультимедіа;
- роботи в комп'ютерних мережах;
- роботи з педагогічними програмними засобами.

В кінцевому результаті завдяки такій наповненості навчальної дисципліни ВСІТ у педагогічному ВНЗ перевіряється початкова комп'ютерна грамотність (набуття досвіду і знань виконання практичних операцій на комп'ютері, оволодіння інструментарієм Windows- та офісних додатків), розвивається загальна комп'ютерна грамотність (використання баз даних і знань, прикладних програмних засобів та засобів телекомунікацій), формується фахова комп'ютерна грамотність (аналіз спеціальних педагогічних програмних засобів, виготовлення навчально-методичних матеріалів в електронному вигляді). Засвоєння пропонованого курсу стимулює у студентів розвиток професійно-значущих знань:

- загально-інформаційних (уявлення про різні види представлення навчального матеріалу в електронному вигляді, про відповідні методи його науково-педагогічного опрацювання);
- користувацьких (формування вміння і навичок реалізувати сучасні інформаційні технології для супроводу, аналізу, коригування навчального процесу в початковій школі, орієнтуватися в сучасному програмному забезпеченні);
- педагогічних (уміння добирати найраціональніші методи і засоби навчання, враховуючи вікові особливості молодших школярів, їх нахили і здібності).

3. До програмно-технологічної змістовної лінії підготовки вчителів інформатики відноситься дисципліна «Основи роботи в операційній системі» (вивчається протягом 3-х семестрів, починаючи з 2-го курсу).

Найважливіші програми на комп'ютері – це програми операційної системи, які забезпечують взаємодію пристроїв і програм при виконанні всіх видів роботи, створюють засоби керування комп'ютером. Знання основ програмного забезпечення операційних систем та вміння працювати з системними програмами є важливими для студентів даної спеціалізації та закладають необхідну базу для гнучкого і широкого застосування ними прикладного програмного забезпечення в їх подальшій навчальній та професійній діяльності. Нині сучасна освіта потребує спеці-

алістів, які вміють ефективно використовувати всі можливості сучасних ОС, опанували засоби системного програмування і мають достатню кваліфікацію для застосування нових ОС. Саме для формування такої підготовки і призначена дисципліна «Основи роботи в операційній системі».

Системне програмне забезпечення охоплює операційні системи (ОС), їх інструментальні засоби, різні види утиліт і оболонок, що розширюють можливості ОС, а також інтерпретатори і компілятори мов програмування. Хоча на сьогодні використовуються різні види ОС, проте більшість з них має багато спільного щодо використання. Це уможливорює вивчення сучасних ОС та відповідного системного програмного забезпечення на прикладах, які вже дістали статус класичних. Зокрема, це стосується загальних питань проектування та використання ОС, особливостей таких популярних операційних систем, як Windows, завдань використання системних програм.

Основним завданням вивчення дисципліни «Основи роботи в операційній системі» є ознайомлення студентів з такими поняттями як системне, службове та прикладне програмне забезпечення, драйвери та утиліти, інтерфейс користувача, використання автономної та онлайнної довідки, організація пошуку даних, навігація, стискання, архівування та розархівування даних, комп'ютерні віруси, антивірусні програми, програми для запису інформації на оптичні носії, програми-перекладачі, програми розпізнавання текстів, встановлення й видалення програмного забезпечення тощо. У методичних рекомендаціях до засвоєння дисципліни зроблено спробу змістити акцент з користувачького підходу навчання інформатики до вивчення загальнотеоретичних, інваріантних принципів функціонування операційних систем, що дозволить забезпечити фундаментальні знання незалежно від використовуваного програмного забезпечення.

Структурування змісту курсу «Основи роботи в операційній системі» здійснювалося, виходячи із реального навчального процесу у педагогічному університеті, з підсиленням спрямованості курсу на формування методичної культури студента, із взаємозв'язку з іншими предметами в траєкторії професійного становлення майбутнього вчителя інформатики початкової школи, із постійної рефлексії співвідношення теоретичної підготовки з результатами педагогічної практичної діяльності студентів.

4. Дисципліна «Основи програмування» теж відноситься до програмно-технологічної змістовної лінії підготовки вчителів інформатики, її вивчення проходить протягом наступних 3-х семестрів, починаючи з другої половини 3-го курсу. Метою вивчення є формування теоретичної

бази знань студентів з основ програмування та практичних навичок використання середовища програмування для створення власних проєктів.

Аналіз чинних державних стандартів і навчальних планів педагогічних ВНЗ України з напрямів підготовки „Інформатика» свідчить, що навчання програмування майбутніх педагогів переважно включає вступний курс з програмування в середовищі офісних додатків та курс із об'єктно-орієнтованого програмування.

Хронологічно після детального розгляду технологій роботи з офісними додатками (в рамках дисципліни ВСІТ) пропонується ознайомлення з можливостями офісного програмування в межах пакету MS Office. Visual Basic for Applications (VBA) – розвинена система візуального програмування, призначена для створення прикладних програм у середовищі MS Office та автоматизації роботи в додатках цього пакету. За допомогою VBA можна створювати об'єкти управління графічного інтерфейсу користувача, змінювати властивості об'єктів, приєднувати до них програмний код, автоматизувати роботу за допомогою макросів тощо. Розробка макросів для офісних додатків на основі базових понять візуального програмування, таких як змінні, типи даних, оператори, алгоритмічні структури, підпрограми й масиви, сприймається студентами легше й цікавіше, ніж створення класичних алгоритмів і програм.

Також в рамках дисципліни «Основи програмування» розглядається програмування мовою Лого. Останнім часом мова програмування Лого завойовує все більшу популярність в початковій і середній школі, у коледжах і університетах, та й просто в середовищі аматорів інтелектуального дозвілля. У зв'язку із цим стає очевидним актуальність пропонованого напрямку, тому доцільно використовувати цю мову при навчанні програмування майбутніх учителів інформатики в початковій школі. Мова програмування Лого є оптимальним вибором для вказаної спеціалізації, знайомство з різноманітними можливостями якої сприяє формуванню пізнавальної, інформаційної і комунікативної компетенції студентів.

Вибір саме цього програмного продукту обґрунтований наступними вимогами до середовища, що розвиває абстрактне й будь-яке інше мислення:

- наявність очевидних інструментів маніпулювання поняттями;
- можливість ускладнення інструментів у міру ускладнення понять;
- наявність засобів, що збільшують емоційне сприйняття (мультимедіа вкладення);
- можливість використання об'єктного (системного) підходу.

Це пропонується для ознайомлення інтерпретована динамічна візу-

альна мова програмування Scratch, яка останнім часом рекомендується саме для навчання дітей поняттю програмування. При роботі в середовищі Scratch проявляються багато ідей програмування, властивих середовищу Лого та Лего-Лого. Але тут вони втілені на більш високому рівні. У Scratch можна створювати фільми, гратися з різними об'єктами, видозмінювати їх вигляд, переміщувати їх по екрану, встановлювати форми взаємодії між об'єктами. Це об'єктно-орієнтоване середовище, в якому блоки програм збираються з різнокольорових «цеглинок»-команд так само, як різні конструкції збираються з різнокольорових блоків у конструкторах Лего.

Так доцільно організоване навчання основ програмування з використанням різних середовищ розвиває алгоритмічне й логічне мислення в природній для цього обстановці; дає досвід роботи з різними моделями; знайомить із загальними принципами й методами програмування, що дозволяє майбутнім учителям інформатики початкової школи адаптувати придбані навички при освоєнні інших комп'ютерних середовищ.

5. Професійно-методична підготовка вчителя інформатики – актуальна проблема сьогодення. Розвиток засобів інформатизації, інформаційних і телекомунікаційних технологій приводить до перегляду вікових і змістових нормативів навчання інформатики, що вимагає кардинального переосмислення цілей, змісту, засобів, методів і форм підготовки з інформатики на сучасному рівні і повинне знайти відображення як у системі загальної освіти, так і у підготовці вчителів інформатики для відповідних освітніх ланок.

Внаслідок наявності різних концепцій шкільного курсу інформатики, спираючись на досягнення сучасної педагогічної науки, необхідно надати вчителю інформатики таку методичну підготовку, яка дозволила б йому самому розв'язувати проблеми побудови шкільного навчального предмета, особливо, пропедевтичного курсу в початковій школі.

Дисципліна «Методика викладання інформатики в початковій школі» відноситься до предметно-методичної змістовної лінії підготовки вчителів інформатики і вивчається протягом 2-х семестрів, на 4-му курсі. Саме ця дисципліна потребуватиме найближчим часом значного вдосконалення (і можливо збільшення часового навантаження) у зв'язку з введенням Нового стандарту початкової школи.

Перш за все курс методики навчання інформатики в початковій школі повинен забезпечити:

– орієнтування студентів в можливостях навчання інформатики саме молодших школярів, щоб майбутні вчителі розуміли, як повинна змінюватися методика навчання в залежності від особливостей навчального процесу, спрямування навчання, віку дітей, і могли застосовувати

ці знання в практичній роботі;

– формування знань і вмінь, необхідних для розуміння підходів до профільної і рівневої диференціації навчання, що ставить нові вимоги до навчання інформатики (підходи до навчання учнів з гуманітарними нахилами, повинні бути іншими, ніж учнів, що схильні до природничого чи математичного напрямку;

– встановлення зв'язків між змістом навчання інформатики та інших предметів початкової школи, на яких формуються основні інформаційні уявлення, навички та вміння молодших школярів;

– формування у майбутнього вчителя вмінь організовувати різні види позакласної роботи з комп'ютерною підтримкою.

Обов'язково слід знайомити студентів з існуючими варіантами програм початкового курсу інформатики: їх змістовими лініями (інформація і інформаційні процеси; знайомство з комп'ютером і формування навичок роботи з ПК; формування логічного та алгоритмічного мислення).

Висновки. Неможливо передбачити все різноманіття можливих форм використання комп'ютера в навчальному процесі початкової школи, тому підготовка студента у вищому педагогічному навчальному закладі повинна бути відкрита для здійснення будь-якої корекції процесу навчання, а це означає, що предметна підготовка вчителя початкової школи і інформатики повинна стати настільки гнучкою і варіативною, щоб забезпечити інтеграцію змінюваних змісту і методів навчання залежно від форм використання комп'ютера і інформаційно-комунікаційних технологій та відповідного програмного забезпечення.

Перспективи подальших розробок у цьому напрямку. В наступних публікаціях планується більш детальний розгляд методики навчання окремих дисциплін інформатичної підготовки майбутніх учителів інформатики початкової школи.

Література

1. Нові стандарти в початковій школі запровадять наступного року [Електронний ресурс] – Режим доступу : <http://ukr.obozrevatel.com/news/novi-standarti-v-pochatkovij-shkoli-zaprovadyat-nastupnogo-roku.htm>
2. Шиман О. І. Формування основ інформаційної культури майбутніх учителів початкової школи : дис. ... канд. пед. наук : 13.00.02 – теорія і методика інформатики» / Шиман Олександра Іванівна ; Національний педагогічний університет імені М. П. Драгоманова. – К., 2005. – 292 с.

ПРОБЛЕМИ В ІНФОРМАТИЧНІЙ ОСВІТІ МАЙБУТНІХ ЕКОНОМІСТІВ ТА ШЛЯХИ ЇХ ВИРІШЕННЯ

О. М. Яцько

Україна, м. Чернівці, Буковинський державний фінансово-економічний
університет
verigao@ukr.net

Інформатика та інформатична освіта в сучасних умовах відіграють особливу роль у підготовці майбутніх фахівців у галузі математики, техніки, комп'ютерних та інформаційних технологій, виробництва, економіки, управління як у плані формування певного рівня інформаційної культури, інтелектуального розвитку, так і в плані формування наукового світогляду, розуміння сутності практичної спрямованості інформатичних дисциплін. Сьогодні ІКТ-компетентності є невід'ємною та важливою частиною загальнолюдської культури, вони є ключем до пізнання оточуючого світу за допомогою комп'ютера та інформаційних технологій, важливою компонентою розвитку особистості. Тому кожна людина має право на якісну інформатичну освіту і обов'язок суспільства – надати кожній особистості можливість скористатися цим правом.

З розвитком цивілізації постійно збільшується роль інформатики та її методів у пізнанні світу, все глибшою стає інтеграція інформатики з іншими науками, інформаційні технології сьогодні є потужним інструментом розв'язування складних задач, що виникають у різних сферах людської діяльності, тобто відбувається поступовий процес інформатизації науки і практики. У зв'язку з цим зазнають суттєвих змін і вимоги до інформатичної освіти майбутніх фахівців у будь-якій сфері, зокрема й у галузі виробництва, економіки, фінансів, управління, тому дослідження проблем, що виникають у вищій інформатичній освіті, та відшукування шляхів їх вирішення, є актуальної соціально значущою задачею.

Проблеми інформатичної освіти, розробки теоретичних і методичних аспектів навчання інформатики в сучасних умовах знайшли відображення в працях Л. І. Білоусової, С. О. Бешенкова, Т. О. Бороненко, О. І. Бочкіна, В. Ю. Бикова, М. І. Жалдака, В. І. Клочка, К. К. Коліна, О. А. Кузнецова, В. В. Лаптева, М. П. Лапчика, Н. В. Макарової, Н. В. Морзе, С. А. Ракова, Ю. С. Рамського, І. В. Роберт, С. О. Семерікова, Ю.В. Триуса та інших.

Безпосередньо проблемами теорії та методики навчання інформатики майбутніх економістів займалися М. С. Головань [1], В. І. Клочко і Н. І. Праворська [2], Ю. М. Красюк [3] та інші.

Розглянемо деякі аспекти навчання інформатики студентів економі-

чних спеціальностей у ВНЗ.

Мета і завдання навчання інформатики. Базовою дисципліною, де формуються *ІКТ-компетентності майбутніх економістів*, є курс «Інформатика». Він є основою для вивчення таких нормативних і вибірко-вих інформатичних дисципліни, як «Інформаційні системи і технології в обліку та аудиті», «Інформаційні системи і технології в економіці», «Інформаційні системи і технології на підприємстві», «Інформаційні системи і технології в оподаткуванні», «Інформаційні системи і технології у фінансах», «Інформаційні системи і технології в банківських установах», «Інформаційні митні технології», передбачених освітньо-професійними програмами напрямів підготовки «Економіка підприємства», «Фінанси і кредит», «Облік і аудит».

Згідно галузевого стандарту вищої освіти з напрямку підготовки «Економіка і підприємництво» [4, 5], *мета навчання інформатики* – формування у студентів знань про принципи побудови та функціонування комп'ютерної техніки, організацію обчислювальних процесів на персональних комп'ютерах та їх алгоритмізацію, програмне забезпечення персональних комп'ютерів і комп'ютерних мереж, а також формування умінь і навичок ефективного використання сучасних інформаційно-комунікаційних технологій у майбутній професійній діяльності; *завдання курсу інформатики* – вивчення студентами теоретичних основ інформатики і набуття ними навичок використання прикладних систем оброблення економічних даних та систем програмування для персональних комп'ютерів і комп'ютерних мереж під час дослідження соціально-економічних систем та розв'язування завдань економічного спрямування.

Аналіз робіт [1], [2], [3] показав, що сьогодні, крім зазначених завдань, основними завдання навчання інформатичних дисциплін у вищих навчальних закладах економічного профілю є: продемонструвати студентам сутність наукового підходу до вивчення процесів і явищ оточуючого світу, роль інформатики у розвитку наукових досліджень і технічному прогресі; сформувати навички самостійного вивчення інформатичної літератури, інших інформатичних джерел та застосування здобутих знань для самоосвіти і саморозвитку.

При цьому вивчення інформатичних дисциплін у ВНЗ повинно забезпечити:

– формування особистості студентів, розвиток їхніх інтелектуальних здібностей, аналітичного та синтетичного мислення, відповідної інформаційної культури та інтуїції;

– оволодіння інформатичним апаратом, необхідним для вивчення фахових дисциплін, розвиток здібностей свідомого сприйняття матеріа-

лу з різних інформаційних ресурсів, характерного для відповідної професії;

– оволодіння основними методами, необхідними для аналізу і моделювання процесів і явищ, які відбуваються в соціальних, економічних, технічних, виробничих та інформаційних системах, пошуку оптимальних рішень з метою підвищення ефективності роботи зазначених систем, вибору найкращих способів реалізації цих рішень, опрацювання і аналізу результатів обчислювальних експериментів;

– формування достатнього рівня інформатичної підготовки випускників ВНЗ для майбутньої професійної діяльності, інформатичної самоосвіти, наукової діяльності з використанням сучасних досягнень у галузях інформатики та інформаційних технологій;

– формування компонентів інформаційної культури, які пов'язані з майбутньою професійною діяльністю в умовах інформаційного суспільства.

Проблеми інформатичної освіти. Розглянемо основні проблеми, що виникають при навчанні інформатичних дисциплін у вищій школі і хвилюють як викладачів, так і студентів.

Аналіз стану навчання інформатичних дисциплін у деяких ВНЗ м. Чернівці (Буковинському державному фінансово-економічному університеті та Чернівецькому національному університеті імені Юрія Федьковича) показав, що результати навчання студентів, рівень їхньої інформаційної культури, пізнавальної активності і самостійності не відповідає вимогам інформаційного суспільства. Все це негативно відбивається на якості знань і умінь студентів, їх інтелектуальному розвитку, рівні фахової підготовки. Проведене дослідження дозволило серед низки проблем, з якими стикаються студенти при вивченні інформатичних дисциплін, виділити кілька основних:

- низький рівень базової теоретичної підготовки з інформатики;
- недостатній рівень практичних умінь та навичок щодо використання цих знань;
- недостатній рівень навчально-пізнавальної діяльності студентів;
- невміння і небажання студентів працювати самостійно;
- недостатня кількість годин, що відведені на вивчення інформатичних дисциплін;
- відсутність якісних сучасних підручників, посібників та інших методичних матеріалів;
- невміння застосовувати інформатичні знання, вміння і навички для розв'язування практичних задач, що виникають у сфері їх майбутньої професійної діяльності.

На рис. 1 і 2 наведені результати відповідей студентів-економістів

(158 респондентів) та викладачів інформатичних дисциплін (132 респонденти) на питання «У чому Ви вбачаєте причини низького рівня Ваших знань або знань Ваших одногрупників з інформатичних дисциплін?» із зазначенням, які з цих причин є головними, які – суттєвими, а які – несуттєвими.

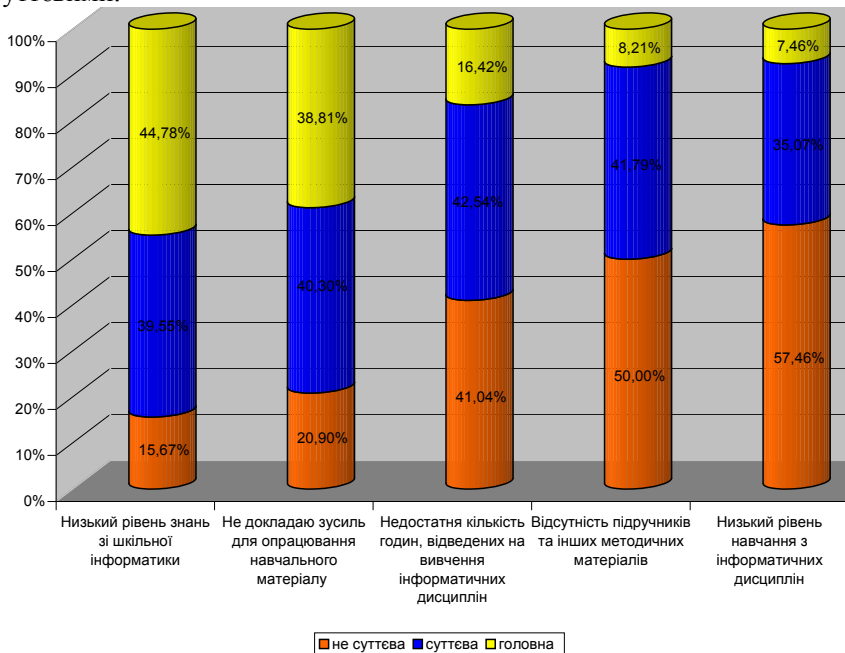


Рис. 1. Причини низького рівня знань з інформатичних дисциплін (на думку студентів)

Як свідчать наведені результати, головними причинами низького рівня знань студентів ВНЗ з інформатичних дисциплін викладачі і, що найголовніше, самі студенти вважають низький рівень підготовки зі шкільної інформатики (відповідно 75,41% і 44,78%), невміння й небажання студентів самостійно і наполегливо працювати з навчальним матеріалом (відповідно 54,10% і 38,81%). Все це відбувається в умовах перенесення у вищій школі більше половини навчального матеріалу на самостійне опрацювання.

Зазначені проблеми і пошук шляхів їх вирішення хвилюють багатьох відомих вчених, викладачів інформатичних дисциплін у ВНЗ. Але однозначної відповіді щодо вирішення цих питань поки що немає.

Серед шляхів подолання проблем, які існують сьогодні у вищій освіті взагалі й інформатичній освіті зокрема, ключове місце належить

активізації навчально-пізнавальної діяльності студентів при вивченні інформатичних дисциплін на основі широкого використання педагогічних та інформаційно-комунікаційних технологій.

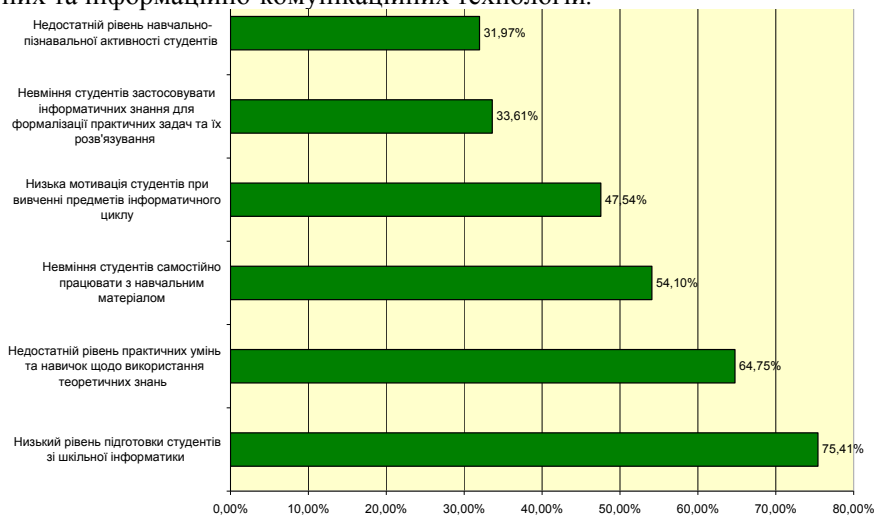


Рис. 2. Причини низького рівня знань з інформатичних дисциплін (на думку викладачів)

Цілеспрямована робота викладачів з формування та розвитку пізнавальної активності студентів – запорука підвищення якості їхньої інформатичної підготовки, розвитку їх мислення. Значні дидактичні можливості для підвищення рівня пізнавальної активності пов'язані з використанням сучасних інформаційно-комунікаційних технологій.

Можна виділити групу найважливіших чинників активізації навчально-пізнавальної діяльності студентів, ефективність яких може бути підсилена за рахунок застосування у навчальному процесі інформаційних-комунікаційних технологій [5]:

- розвиток мотивації, посилення інтересу до навчання, в тому числі до способів здобування знань;
- розвиток мислення, інтелектуальних здібностей студентів;
- індивідуалізація та диференціація навчання;
- розвиток самостійності;
- надання переваги методам активного навчання;
- підвищення наочності навчання;
- збільшення арсеналу засобів пізнавальної діяльності, опанування сучасними методами наукового пізнання, пов'язаними із застосуванням комп'ютерів (комп'ютерний експеримент, іміта-

- ційне моделювання);
- розширення кола задач і вправ, зокрема практичного змісту, та їх розв'язування з використанням комп'ютерних систем, інформаційних технологій;
 - спрощення та збільшення швидкості доступу до навчальних та наукових інформаційних ресурсів через мережу Internet, за допомогою технологій мобільного зв'язку.

Підвищення рівня знань студентів з інформатичних дисциплін, активізація їхньої навчально-пізнавальної діяльності та подолання інших зазначених вище проблем навчання вбачається також у створенні та впровадженні у навчальний процес методичних систем навчання інформатичних дисциплін на основі широкого використання інноваційних педагогічних та інформаційно-комунікаційних технологій.

Таким чином, актуальною є проблема розробки, наукового обґрунтування та експериментальної перевірки ефективності комп'ютерно орієнтованих методичних систем навчання інформатичних дисциплін у вищих навчальних закладах, зокрема, при підготовці фахівців у галузі економіки та фінансів, використання яких надасть можливість активізувати навчально-пізнавальну діяльність студентів і підвищити рівень їхньої інформатичної підготовки за рахунок широкого застосування інформаційно-комунікаційних та педагогічних технологій.

Автором проводиться науково-дослідна робота з створення комп'ютерно орієнтованої методичної системи навчання інформатики студентів економічного і фінансового профілів, про що більш детально йтиметься у доповіді.

Висновки.

Підсумовуючи сказане щодо шляхів подолання негативних тенденцій у інформатичній освіті, сформулюємо деякі пропозиції щодо їх практичної реалізації:

1. Привести у відповідність програми навчання інформатики в школі та у вищих навчальних закладах. Модернізувати інформатичні курси, наповнивши їх зміст відомостями про сучасні досягнення інформатики, інноваційні інформаційно-комунікаційні технології, зокрема у галузі економіки і, таким чином, покращити рівень базової теоретичної підготовки з інформатики;

2. Розробляти та впроваджувати в навчальний процес ВНЗ комп'ютерно орієнтовані методичні системи навчання інформатичних дисциплін на основі інноваційних педагогічних та інформаційно-комунікаційних технологій з використанням навчальних комплексів, електронних підручників та посібників, робочих конспектів, контролюючих і тренувальних комп'ютерних програмних засобів, електронних

систем навчання, що нададуть можливість якщо не повністю, то частково усунути таку проблему інформатичної освіти, як відсутність якісних сучасних підручників, посібників та інших методичних матеріалів;

3. У вищих навчальних закладах необхідно створювати освітньо-наукове інформаційне середовище, яке надасть можливість ефективно використовувати ІКТ для проведення аудиторних, зокрема практичних і лабораторних, занять з інформатичних дисциплін, контрольних заходів, організації науково-дослідної роботи і, особливо, для самостійної роботи студентів денної, заочної та дистанційної форм навчання. Це надасть можливість стимулювати студентів працювати самостійно, частково вирішити проблему із недостатньою кількістю годин, що відведені на вивчення інформатичних дисциплін.

Література

1. Головань М. С. Компетентнісний підхід у навчанні інформатики і комп'ютерної техніки студентів економічного ВНЗ / М. С. Головань // Проблеми інженерно-педагогічної освіти : зб. наук. праць / Українська інженерно-педагогічна академія. – Харків : УПА, 2007. – Вип. 18-19. – С. 19-32.

2. Клочко В. І. Система задач як засіб формування професійно значущих знань з інформатики студентів економічних спеціальностей : монографія / Клочко В. І., Праворська Н. І. – Вінниця : УНІВЕРСУМ-Вінниця, 2008. – 140 с.

3. Красюк Ю. М. Методика навчання інформатики студентів економічних спеціальностей : автореф. дис.... канд. пед. наук : 13.00.02 – теорія та методика навчання інформатики / Красюк Юлія Миколаївна ; НПУ ім. М. П. Драгоманова. – К., 2004. – 22 с.

4. Освітньо-професійна програма підготовки бакалаврів за спеціальностями напрямку 0501 – «Економіка і підприємництво» – К. : МОН України, 2010. – 11 с.

5. Триус Ю. В. Комп'ютерно-орієнтовані методичні системи навчання математики : монографія / Триус Ю. В. – Черкаси : Брама-Україна, 2005. – 400 с.

Наші автори

Аджікельдієв Айдер Серверович, магістрант Кримського інженерно-педагогічного університету (*розробка ПЗ, нових методик навчання*)

Бідайбеков Єсен Ікласович, д. пед. н., професор, завідувач кафедри інформатики та механіко-математичних спеціальностей Казахського національного педагогічного університету імені Абая (*теорія та методика навчання інформатики, інформатизація освіти*)

Войтенко Леся Петрівна, старший викладач кафедри медичної інформатики та комп'ютерних технологій навчання Національного медичного університету імені О. О. Богомольця (*методологія навчання інформатичних дисциплін у вищій школі*)

Воронов Валерій Олексійович, вчитель математики та інформатики Шепетівського навчально-виховного комплексу №1 (*інформаційні технології в навчанні, самоосвіта студентів та викладачів, взаємозв'язок шкільної освіти та вищої школи*)

Глинський Ярослав Миколайович, к. ф.-м. н., доцент, доцент кафедри обчислювальної математики та програмування Національного університету «Львівська політехніка» (*зміст і методика навчання інформатики*)

Гришко Людмила Веніамінівна, к. пед. н., доцент, доцент кафедри прикладної математики Черкаського національного університету імені Богдана Хмельницького

Злобін Григорій Григорович, к. т. н., доцент, доцент кафедри радіофізики і комп'ютерних технологій Львівського національного університету імені Івана Франка (*вільне програмне забезпечення*)

Льясова Фатіме Серверівна, викладач кафедри інформаційно-комп'ютерних технологій Кримського інженерно-педагогічного університету (*методологія навчання фундаментальних дисциплін у ВНЗ*)

Камалова Гульдїна Большевіківна, д. пед. н., доцент, доцент Казахського національного педагогічного університету імені Абая (*теорія та методика навчання інформатики, інформатизація освіти*)

Кисельова Олена Анатоліївна, к. пед. н., старший викладач Казахського національного педагогічного університету імені Абая (*методика навчання інформатики*)

Когут Уляна Петрівна, здобувач Інституту інформаційних технологій і засобів навчання НАПН України (*використання систем комп'ютерної математики при навчанні інформатичних дисциплін*)

Кривонос Олександр Миколайович, старший викладач кафедри прикладної математики та інформатики Житомирського державного університету імені Івана Франка (*методика викладання інформатики, програмування*)

Кулик Галина Ігорівна, к. т. н., доцент, доцент кафедри прикладної математики Придніпровської державної академії будівництва та архітектури (*використання інформаційних технологій в освіті*)

Лукаш Ірина Миколаївна, к. пед. н., доцент, доцент кафедри інформатики і ОТ Чернігівського національного педагогічного університету імені Т. Г. Шевченка (*методика навчання інформатики*)

Маковецька Світлана Василівна, асистент Національного університету харчових технологій (*розробка баз даних технологічних процесів у технології харчових концентратів*)

Медведева Ольга Анатоліївна, старший викладач кафедри прикладної математики Донбаської державної машинобудівної академії (*методика використання сучасних ІКТ у навчанні інформатики*)

Мінтій Ірина Сергіївна, асистент кафедри інформатики та прикладної математики Криворізького національного університету (*методика навчання інформатики, мобільне навчання*)

Моїсеєнко Михайло Вікторович, старший викладач кафедри інформатики та прикладної математики Криворізького національного університету

Моїсеєнко Наталя Володимирівна, к. ф.-м. н., доцент, доцент кафедри інформатики та прикладної математики Криворізького національного університету

Монцібович Борис Романович, к. ф.-м. н., доцент, доцент кафедри програмного забезпечення систем Львівського навчально-наукового інституту Закарпатського державного університету (*чисельні методи, програмне забезпечення, методи апроксимації дослідних даних*)

Мясищев Олександр Анатолійович, д. т. н., професор, професор Хмельницького національного університету (*комп'ютерні мережі, ресурси мережі Інтернет, паралельні обчислення*)

Овчарук Володимир Олексійович, к. т. н., доцент, завідувач кафедри інформатики Національного університету харчових технологій (*викладання інформаційних технологій*)

Овчарук Ірина Вікторівна, к. т. н., доцент, доцент кафедри інформаційних технологій Київської державної академії водного транспорту імені Гетьмана Петра Конашевича-Сагайдачного (*викладання ІТ*)

Олефіренко Надія Василівна, к. пед. н., доцент, доцент кафедри інформатики Харківського національного педагогічного університету імені Г. С. Сковороди (*теорія та методика викладання інформатики*)

Павлова Наталія Степанівна, к. пед. н., доцент кафедри інформаційно-комунікаційних технологій та методики викладання інформатики Рівненського державного гуманітарного університету (*методика викладання інформатики, ІКТ у навчальному процесі*)

Поліщук Олександр Павлович, к. т. н., старший науковий співробітник, доцент кафедри інформатики, теорії і методики навчання інформатики Криворізького національного університету (*методика навчання інформатики, комп'ютерне моделювання динамічних систем*)

Поттосін Юрій Васильович, к. ф.-м. н., доцент, провідний науковий співробітник Об'єднаного інституту проблем інформатики НАН Білорусі (*автоматизація логічного проектування дискретних пристроїв*)

Поттосіна Світлана Анатоліївна, к. ф.-м. н., доцент, доцент Білоруського державного університету інформатики і радіоелектроніки (*дискретна математика, теорія ймовірностей і математична статистика, економетрика*)

Ряжська Вікторія Анатоліївна, к. ф.-м. н., доцент, доцент кафедри обчислювальної математики та програмування Національного університету «Львівська політехніка» (*зміст і методика навчання інформатики*)

Сейдаметова Зарема Сейдаліївна, д. пед. н., професор, завідувач кафедри інформаційно-комп'ютерних технологій Кримського інженерно-педагогічного університету (*освітнє тестування, освітні стандарти*)

Сейдаметова Саніє Мамбетівна, к. пед. н., доцент, доцент Кримського інженерно-педагогічного університету (*методика навчання ІТ-фахівців*)

Семеріков Сергій Олексійович, д. пед. н., професор, професор кафедри фундаментальних дисциплін Криворізького національного університету (*теорія і методика навчання інформатики*)

Сєдих Ольга Леонідівна, старший викладач Національного університету харчових технологій (*розробка та програмування технологічних завдань в Mathcad*)

Стрюк Андрій Миколайович, старший викладач Криворізького національного університету (*ІКТ в навчанні, системне програмування*)

Темненко Валерій Анатолійович, к. ф.-м. н., доцент, доцент кафедри прикладної математики Таврійського національного університету імені В. І. Вернадського (*прикладна математика, динамічні системи*)

Теплицький Ілля Олександрович, к. пед. н., доцент, доцент кафедри інформатики та прикладної математики Криворізького національного університету (*теорія і методика навчання інформатики*)

Теплицький Олександр Ілліч, старший викладач кафедри комп'ютерних систем автоматизованого управління електроприводом Криворізького національного університету (*об'єктно-орієнтоване моделювання*)

Терещенко Світлана Віталіївна, учитель інформатики Білогорського НВК «Школа-ліцей» №2 (*інноваційні технології навчання, синергетична середина навчання, проектна діяльність*)

Хапко Оксана Богданівна, асистент кафедри транспортних технологій Львівської філії Дніпропетровського національного університету залізничного транспорту ім. акад. В. Лазаряна (*математичне моделювання дослідних даних*)

Хараджян Наталя Анатоліївна, к. пед. н., доцент кафедри інформатики та прикладної математики Криворізького національного університету (*комп'ютерне моделювання різних явищ*)

Чаплигіна Світлана Миколаївна, к. ф.-м. н., доцент, доцент кафедри прикладної математики Придніпровської державної академії будівництва та архітектури

Шаров Сергій Володимирович, к. пед. н., старший викладач кафедри інформатики і кібернетики Мелітопольського державного педагогічного університету імені Богдана Хмельницького (*використання інформаційних технологій у навчальному процесі вищої школи, створення та використання електронних підручників, прикладне програмування*)

Шиман Олександра Іванівна, к. пед. н., доцент, доцент кафедри початкового навчання Бердянського державного педагогічного університету (*інформатизація дошкільної та початкової ланки*)

Яцько Оксана Мирославівна, асистент кафедри комп'ютерних дисциплін Буковинського державного фінансово-економічного університету (*комп'ютерно орієнтована методична система навчання економічної інформатики майбутніх фахівців у фінансовій сфері*)

Зміст

<i>Е. Ї. Бидайбеков, Г. Б. Камалова.</i> О фундаментализации подготовки будущих учителей информатики.....	3
<i>Е. Ї. Бидайбеков, Е. А. Киселёва.</i> «Теоретические основы информатики» в системе фундаментальной подготовки учителей информатики	10
<i>Л. П. Войтенко.</i> Змістовно-структурна модель медичної інформатики на додипломному етапі підготовки майбутнього лікаря.....	16
<i>В. О. Воронов.</i> 3D-моделювання як засіб підвищення пізнавальної активності школярів	25
<i>Я. М. Глинський, В. А. Рязьська.</i> Деякі аспекти мобільного навчання програмування у VBA	29
<i>Л. В. Гришко.</i> Функції курсу основ програмування для майбутніх програмістів та принципи навчання.....	40
<i>Г. Г. Злобін.</i> Офісний пакет в курсі інформатики вищого навчального закладу	49
<i>Ф. С. Ильсова.</i> Обучение языку Java как одному из фундаментальных языков программирования для обучения инженеро-программистов	51
<i>У. П. Козут.</i> Модель фундаменталізації навчання інформатичних дисциплін майбутніх бакалаврів інформатики на основі міжпредметних зв'язків	55
<i>О. М. Кривонос.</i> Результати впровадження елементів методичної системи формування інформаційно-комунікаційних компетентностей майбутніх вчителів інформатики в процесі навчання програмування ...	62
<i>Г. И. Кулик.</i> Формирование исследовательских навыков у студентов младших курсов при изучении предмета «Информатика» в техническом вузе.....	67
<i>І. М. Лукаш.</i> Відпрацювання знань і вмінь з інформатики	73
<i>О. А. Медведева.</i> Використання мови програмування Delphi у процесі формування інформатичної компетентності майбутніх інженерів	79
<i>І. С. Мінтій.</i> Рівні сформованості компетентності у програмуванні.....	82
<i>М. В. Моїсеєнко, Н. В. Моїсеєнко.</i> Особливості організації самостійної роботи студентів при вивченні курсу «Подіє-орієнтоване програмування»	87
<i>Б. Р. Монцібович, О. Б. Ханко.</i> Про методи наближення табличних (дослідних) даних засобами VBA MS Excel.....	93
<i>А. А. Мясищев.</i> Сопоставление производительностей GPU и CPU для матричного умножения с двойной точностью	99
<i>І. В. Овчарук, В. О. Овчарук.</i> Використання пакетів Excel та Mathcad при вивченні дисципліни «Економічна інформатика».....	108

<i>Н. В. Олефіренко.</i> Курс інформатики як компонент сучасної початкової освіти.....	112
<i>Н. С. Павлова.</i> Професійний розвиток студента у системі фахової підготовки вчителя інформатики.....	119
<i>О. П. Поліщук, І. О. Теплицький, С. О. Семеріков.</i> Перспективи фундаменталізації шкільного курсу інформатики.....	126
<i>С. А. Поттосина, Ю. В. Поттосин.</i> О преподавании теории графов при подготовке специалистов в области информатики и вычислительной техники.....	132
<i>О. Л. Седих, С. В. Маковецька.</i> Деякі аспекти викладання дисципліни «Інформатика та інформаційні технології» студентам технологічного напрямку Національного університету харчових технологій.....	140
<i>С. М. Сейдаметова, А. С. Аджикельдиев.</i> Междисциплинарные связи в дискретной математике: командная проектная деятельность.....	145
<i>З. С. Сейдаметова, В. А. Темненко.</i> Методические аспекты преподавания дисциплины «Технология разработки программного обеспечения».....	150
<i>А. М. Стрюк.</i> Проектування комбінованого навчання системного програмування бакалаврів програмної інженерії.....	157
<i>О. І. Теплицький.</i> Цілі та зміст навчання об'єктно-орієнтованого моделювання студентів природничих спеціальностей педагогічних університетів.....	164
<i>С. В. Терещенко.</i> Социальное творчество – методический прием формирования мотивации при изучении информатики.....	171
<i>Н. А. Хараджян.</i> Еволюція та сучасний стан підготовки фахівців з економічної кібернетики у ВНЗ України.....	176
<i>С. Н. Чаплыгина.</i> Аппроксимация функций с помощью нейронных сетей.....	182
<i>С. В. Шаров.</i> Створення електронного підручника з мови програмування Qbasic.....	186
<i>О. І. Шиман.</i> З досвіду організації інформатичної підготовки майбутніх учителів інформатики початкової школи.....	191
<i>О. М. Яцько.</i> Проблеми в інформатичній освіті майбутніх економістів та шляхи їх вирішення.....	200
Наші автори.....	207

Наукове видання

**Теорія та методика навчання
математики, фізики, інформатики**

Випуск X

В 3-х томах

Том 3

Підп. до друку 19.03.12
Папір офсетний №1
Ум. друк. арк. 11,0

Формат 80×84 1/16
Зам. №3-1903
Наклад 300 прим.

Жовтнева друкарня
50014, м. Кривий Ріг, вул. Електрична, 5
Тел. (0564) 407-29-02

E-mail: semerikov@gmail.com