

УДК 378.147:004.413

*Кривонос О. М.
Житомирський державний університет
імені Івана Франка*

КОМПЕТЕНТНОСНО-ОРІЄНТОВАНІ ЗАВДАННЯ В КУРСІ “ПРОГРАМУВАННЯ”

У статті розглядається проблема підготовки вчителів інформатики на засадах компетентнісного підходу, зокрема, визначені роль та місце компетентнісно-орієнтованих завдань в курсі програмування. Подана схема створення компетентнісно-орієнтованого завдання з врахуванням методу проєктів. Описано застосування зазначеного підходу в курсі “Програмування” для майбутніх вчителів інформатики.

***Ключові слова.** компетентнісний підхід, навчання програмуванню, метод проєктів.*

У контексті приєднання до Болонського процесу національна система вищої освіти переживає суттєві структурні та змістовні зміни. Сьогодні завдання вищої школи полягає у формуванні в студентів наукового світогляду, системи сучасних професійних знань, розвитку творчих здібностей, професійної кваліфікації. Процес навчання повинен орієнтуватися на поетапне створення в студентів відповідної системи знань, певних умінь та навичок, ключових і професійних компетентностей, зокрема й інформаційно-комунікаційних. Саме компетентності є тими критеріями, які дозволяють визначати готовність випускника до фахової діяльності.

Навчання майбутнього фахівця – це вплив на свідомість та діяльність для підготовки до майбутньої фахової діяльності. Проте результати навчання не повинні обмежуватися формуванням лише професійних знань, умінь та навичок. У процесі навчання повинні формуватися ключові компетентності, інтелектуальні якості – формуватися особистість майбутнього фахівця як єдине ціле [3]. Сучасному вчителю інформатики не достатньо мати навички щодо використання традиційних технологій, а необхідно знати й вивчати освітній сегмент мережі Інтернет, мати навички впровадження ІКТ у навчальний процес, уміти навчати інформатиці за допомогою різних засобів телекомунікації тощо.

Одним із перших, хто в 70-80 роках ХХ століття почав розглядати проблеми методики викладання інформатики, був академік А. П. Єршов. Його загальновідомі праці “Програмування – друга освіта” та “Звідки беруться люди, які можуть створювати надійне програмне забезпечення” створили підґрунтя для методики викладання інформатики як науки в Радянському Союзі. Подальше активне обговорення проблем методики навчання інформатики призвело до створення та введення у практику державних освітніх стандартів, які відобразили існуючі на той час бачення предмета інформатика та відповідних йому знань.

Над удосконаленням методичної системи професійної підготовки вчителя інформатики працюють В. Ю. Биков [2], М. І. Жалдак [7], М. П. Лапчик [10], Н. В. Морзе [11], С. А. Раков [12], Ю. С. Рамський [7], С. О. Семеріков [14], О. В. Співаковський [16], О. М. Спірін [17], Ю. В. Триус [18] та ін.

Аналізуючи ці дослідження, можна стверджувати, що низка аспектів вимагає подальшого вивчення. Зокрема, аналіз стану навчання програмування у вищій школі показує, що рівень сформованості ІКТ-компетентностей майбутніх учителів інформатики після завершення навчання з цього курсу не достатньо відповідає вимогам сьогодення; в умовах кредитно-модульної системи не приділено належної уваги вивченню питання формування компетентностей під час навчання програмування.

Мета статті – визначити роль та місце компетентнісно-орієнтованих завдань у курсі

програмування в процесі фахової підготовки майбутніх учителів інформатики.

Розробка та впровадження окремих елементів системи або нових технологій навчання можливі лише за умови дотримання дидактичних принципів з огляду на те, що “принципи... відображають дидактичні закони і закономірності” [1, с. 396]. На сучасному етапі педагогіка вищої школи взяла орієнтир не на заміну узвичаєних дидактичних принципів новими, а на наповнення їх новим змістом з урахування вимог сучасності.

Серед загальноприйнятих дидактичних принципів виділимо такі:

– загальні принципи організації навчального процесу: науковість та доступність, наступність, систематичність, перспективність і наочність [13], принцип виховуючого навчання, принцип єдності теорії й практики, принцип педагогічної обґрунтованості обсягу навчального матеріалу [1];

– принцип диференціації та індивідуалізації навчального процесу, який передбачає присутність і необхідність урахування вікових та індивідуальних особливостей студентів (різний рівень сформованості знань, умінь, навичок, різний темп сприйняття матеріалу тощо);

– принципи добору змісту навчального контенту для кожної спеціальності зокрема: принцип першочерговості розвивальної функції навчання, принцип інформаційної смності й соціальної ефективності, принцип диференційованої реалізованості, принцип діагностико-прогностичної реалізованості, модульний принцип добору змісту, принцип концентризму, принцип гуманізації та гуманітаризації освіти [4];

– принцип якості засвоєння знань, професійних умінь, інтелектуальних навичок розумової праці потребує ефективної системи виявлення результатів навчання у формі поточного, проміжного, підсумкового контролю [6, с. 24];

– принцип актуальності знань і професійних умінь передбачає насамперед орієнтацію на новітні наукові досягнення в обраній галузі діяльності людини, актуальні та перспективні вимоги ринку праці щодо рівня кваліфікації фахівців, забезпечення відповідних умов організації навчання й неперервного свідомого підвищення кваліфікації протягом життя;

– принцип забезпечення креативності й самостійності студентів у навчальному процесі передбачає перенесення акцентів з лекційного навантаження курсу на самостійну роботу під час проведення практичних, лабораторних робіт, розробки індивідуальних науково-дослідних завдань тощо;

– принцип варіативності в побудові індивідуальної траєкторії, виборі форм, методів і засобів навчання, що є розвитком дидактичного принципу альтернативності [19] й передбачає: можливість обґрунтованої зміни студентом спеціальності, спеціалізації в межах напряму підготовки та, як виняток, зміну напряму підготовки.

За роки свого існування більшість курсів з інформатики переважно ставили акценти на набутті навичок програмування. На розповсюдження такого підходу вплинула низка практичних та історичних чинників:

– уміння програмувати є неодмінним для всіх студентів, котрі навчаються за напрямом “Інформатика”. Оволодіння програмуванням на перших курсах певною мірою гарантує наявність необхідних знань при переході до поглиблених курсів (наприклад, об’єктно-орієнтоване програмування, теорія програмування, логічне програмування тощо);

– інформатика стала академічною дисципліною досить пізно, і тому вона довгий час ототожнювалася з програмуванням;

– модель “з орієнтацією на програмування” була підтримана ранніми програмами з інформатики.

Проте підхід з орієнтацією на програмування має декілька мінусів, а саме:

– зосередження на програмуванні за рахунок вилучення інших розділів інформатики дає студентам вузьке розуміння дисципліни;

– теоретичні питання, які повинні закріпити розуміння практичного матеріалу,

відкладаються на більш пізніх етапах навчання, тоді, коли вони вже не мають необхідного значення;

– курси з програмування, як правило, роблять акцент на синтаксисі та особливостях мови програмування, тим самим надмірно спрощують процес навчання програмуванню;

– такий підхід може привести студентів до враження, що створення програми є єдиним методом розв'язання завдання з використанням комп'ютера.

Незважаючи на вищезгадані мінуси, модель з орієнтацією на програмування довела свою неабияку життєздатність протягом останніх десятиліть.

Одним із найактуальніших питань у навчанні інформатики є роль та місце програмування в навчальному плані. Під час навчання програмуванню виникла необхідність використовувати методи підвищеної мотивації навчання студентів комп'ютерних спеціальностей та майбутніх викладачів інформатики для школи [9].

Програмування як розділ інформатики, спеціалізується на створенні конкретних алгоритмів розв'язування задач. Методика навчання програмування, яка має орієнтацію на самостійність і творчість – це кропіткий процес співвідношення синтаксису та семантики мов програмування, принципів побудови програм, спеціалізованих алгоритмів з методами їх засвоєння. Сьогодні для актуалізації та закріплення знань, умінь і навичок з програмування використовують задачі, які передбачають знаходження алгоритму та його відтворення на мові програмування протягом одного заняття. Проте для формування інформаційно-комунікаційних компетентностей студентів такого замало: бракує умов для самостійності, пошуку, об'єм отриманих знань не знаходить свого використання в конкретних життєвих ситуаціях.

Вихід з цієї ситуації бачимо в застосуванні методу проектів, класичне розуміння якого полягає в широкому використанні під час навчання проектів (задач підвищеної складності, час виконання яких не обмежується одним заняттям). Цей метод орієнтований на гуманістичний, індивідуальний підхід до учня та ґрунтується на ідеях Ж. Ж. Русо, Дж. Дьорі, У. Килпатрика й пропонує індивідуальне пізнання навчального матеріалу шляхом розв'язання проблемної ситуації. Для методу проектів характерні висока мотивація, можливість розвитку творчих здібностей і самостійності.

У процесі навчання програмування використання даного методу набуває особливої значущості ще й тому, що в навчальних планах спеціальностей педагогічного спрямування не передбачене обов'язкове виконання дипломного проекту або кваліфікаційної роботи.

До типових ознак навчального проекту належать: мета створення проекту; методи, які переважають у процесі розробки проекту; творчий характер діяльності; характер координації проекту; характер контактів та кількість учасників проекту; час розробки проекту. Основними складовими проекту є: проблема, актуальність, мета, задачі проекту, тип проекту, структура проекту, методи створення програмного комплексу (специфіка програмування), час роботи над проектом, форма представлення результату (рис. 1).

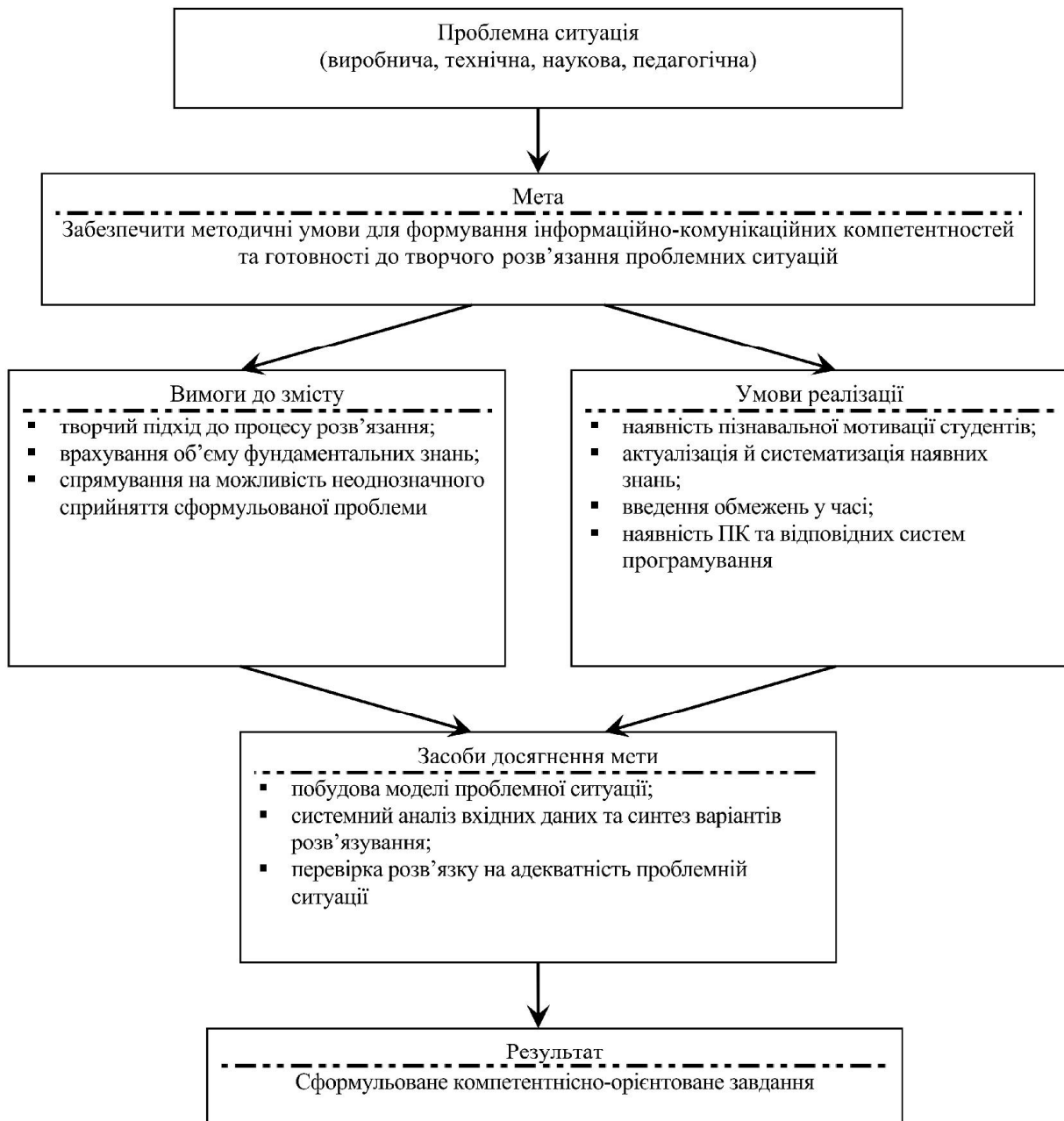


Рис. 1. Схема створення компетентнісно-орієнтованого завдання

Під час роботи над навчальним проектом активно використовується спостереження, висуваються гіпотези, йде експериментальна перевірка, розширюється науковий світогляд. Результат виконання проекту повинен бути "відчутним", тобто, якщо спочатку була теоретична проблема, то повинно бути конкретне її розв'язання, якщо практична – конкретний результат, готовий до використання. Готовий проект повинен пройти захист, і його можна розцінювати як форму контролю.

Кожен проект повинен спиратися на методи побудови алгоритмів, які вивчаються, і передбачати розробку власного алгоритму. Треба, щоб розробка інтерфейсу користувача вимагала від студентів самостійного вивчення роботи із графікою, організації взаємодії з периферійними пристроями на низькому рівні.

Зауважимо, що такий вид навчальної діяльності змінює традиційні ролі студента і викладача. При створенні проекту студенти мають можливість самостійно обирати середовища програмування, встановлювати темп роботи над проектом, шукати та вивчати

необхідну довідкову літературу як у мережі Інтернет, так і в друкованому вигляді, знаходити консультантів на спеціалізованих форумах. Викладач лише коригує роботу студентів, спрямовуючи їхні зусилля в певне русло. Причому дії викладача носять рекомендаційний характер. Крім того, для надання консультації й контролю над процесом роботи встановлено графік консультацій викладача.

У Житомирському державному університеті імені Івана Франка на фізико-математичному факультеті майбутні вчителі інформатики під час вивчення курсу “Програмування” отримують на початку четвертого семестру компетентісно-орієнтоване завдання (його добір проходить відповідно до рис. 1). Це практикується для подолання протиріччя між абстрактним предметом навчально-пізнавальної діяльності студентів і їх майбутньої професійною діяльністю. Саме тому студентам пропонують об’єднатися в мікрогрупи по 2-3 особи для створення власного навчального програмного продукту й продемонструвати свою розробку після завершення вивчення дисципліни.

Програмний продукт – “це не тільки програми, але і вся супутня документація, а також конфігураційні дані, необхідні для коректної роботи програм” [15]. Не наголошується на обов’язковості створення програми у вивчених середовищах програмування. Як показує досвід, більшість студентів розробляють власні програмні продукти у візуальних середовищах програмування, паралельно вивчаючи нові, більш перспективні середовища (Delphi, Micro Media Flash, Microsoft SilverLingt).

Демонстрація проектів та їх захист відбувається публічно, тобто в присутності всіх студентів групи та, за бажанням, інших осіб. Важливо, щоб студенти кожної мікрогрупи мали можливість порівняти власну розробку з розробками інших мікрогруп. Оцінювання якості проекту і внесок кожного студента (розробника) в проект здійснює комісія. До складу комісії, окрім викладачів, які читають курс, доцільно включати студентів старших курсів, котрі мають певний досвід створення програмних продуктів. Члени комісії повинні оцінити проект у цілому: відповідність поставленому завданню; алгоритмічну складність; завершеність проекту; об’єм виконаної роботи; якість призначеного для користувача інтерфейсу; супровідну документацію.

Крім того, комісія визначає підсумковий бал кожного з учасників колективної розробки на основі виконаної частини проекту.

За результатами самооцінювання виконання завдання студенти не тільки визначають свій потенціал щодо рівня використання ІКТ, а й обирають відповідні теми та спрямування, які дозволять їм “заповнити прогалини в знаннях” і забезпечити подальше вивчення вузівських предметів із використанням ІКТ.

У процесі розв’язування пропонованого завдання студенти виявляють навички, які є ключовими для інформаційно-комунікаційно-технологічної компетентності, відомими під назвою “велика сімка” [5]. Вони показують, як універсальні навички пошуку й перетворення даних з допомогою комп’ютера та інших засобів ІКТ можуть бути інтегровані в систематичний процес, орієнтований на розв’язання практичного завдання. До цього переліку належать: визначення (ідентифікація даних) – визначити умову задачі, ідентифікувати необхідні дані; управління – виявити всі можливі джерела даних та відібрати серед знайдених ті, котрі найбільше відповідають проблематиці задачі; пошук даних – знайти необхідне джерело даних та необхідний матеріал всередині джерела; інтеграція – порівняти та зіставити відомості з різних джерел, подавати одержані результати належним чином; оцінка – знайдені дані відповідають критеріям відбору та правильно оцінені ресурси, затрачені на цей пошук; створення – розв’язувати задачу на основі наявного матеріалу або створювати нові дані; передавання даних – передача повідомлень або відомостей із використання сучасних ІКТ.

Перше ознайомлення студентів з компетентісно-орієнтованим завданнями на заняттях із програмування відбувається тоді, коли їм пропонується розв’язати кілька задач на порталі E-olimp [8]. А це вже можна робити після перших лабораторних робіт. У ході виконання цього завдання студенти повинні:

- використовуючи один із наявних браузерів, зайти на портал E-olimp (e-olimp.com.ua);
- ознайомитися зі структурою порталу;
- зареєструватися на цьому мережевому ресурсі (це можливо лише при наявності в студента електронної скриньки);
- знайти задачу за вказаним номером;
- скласти математичну модель задачі;
- знайти алгоритм розв'язання;
- використовуючи середовище програмування, описати алгоритм на певній мові програмування;
- перевірити правильність алгоритму на тестах (один тест запропоновано в тексті задачі, інші студент розробляє самостійно);
- використовуючи буфер обміну, перенести код програми до спеціального поля на сторінці порталу й виконати відправку коду, зазначивши середовище програмування та наявність у коді програми операторів роботи з файлами;
- у разі не проходження відправленою відповіддю усіх тестів системи або виникнення повідомлення про помилку, доопрацювати алгоритм та виконати останні три етапи (повідомлення про помилку буває трьох типів – помилка компіляції; перевищено час роботи програми; перевищено об'єм пам'яті, яка використовується).

У такий спосіб студенти працюють не лише з середовищем програмування, а й використовують інші програмні продукти. Під час роботи їм дозволено спілкуватися та шукати необхідні матеріали на порталі або в мережі Інтернет. А також отримувати допомогу від інших студентів або викладача.

Отже, при розв'язуванні компетентісно-орієнтованого завдання, студенти створюють закінчений програмний продукт та усвідомлюють свої можливості, що є певним стимулом до подальшого навчання й самовдосконалення. При реалізації методу проектів уся проектна діяльність спрямована на студента. Самостійність у виборі навчальної траєкторії дає можливість студенту вийти на новий, більш високий рівень роботи з інформаційними і комунікаційними технологіями й розглядати їх як інструмент пізнання і саморозвитку, що, у свою чергу, сприяє прояву соціальної активності особистості.

Використана література:

1. *Алексюк А. М.* Педагогіка вищої освіти України: Історія. Теорія : підруч. для студ., асп. та мол. викл. вузів / А. М. Алексюк – К. : Либідь, 1998. – 558 с.
2. *Биков В. Ю.* Сучасні завдання інформатизації освіти / В. Ю. Биков // Інформаційні технології і засоби навчання: електронне наукове фахове видання [Електронний ресурс] / Ін-т інформ. технологій і засобів навчання АПН України, Ун-т менеджменту освіти АПН України ; гол. ред. : В. Ю. Биков. – 2010. – № 1 (15). – Режим доступу : URL :<http://www.ime.edu-ua.net/em15/emg.html>. – Назва з екрану.
3. *Буланова-Топоркова М. В.* Педагогика и психология высшей школы : учеб. пос. / М. В. Буланова-Топоркова. – [3-е изд., перераб. и доп.] – Ростов-н/Д : Феникс. – 2006. – 512 с.
4. *Бурда М. І.* Методичні основи диференційованого формування геометричних умінь учнів основної школи : дис. ... доктора пед. наук : 13.00.02 / Бурда Михайло Іванович. – К., 1994. – 347 с.
5. *Бурмакина В. Ф.* Большая Семёрка (Б7). Информационно-коммуникационно-технологическая компетентность. Методическое руководство для подготовки к тестированию учителей [Электронный ресурс] / В. Ф. Бурмакина, М. Зелман, И. Н. Фалина. – М., 2007. – Режим доступу : URL : <http://ifap.ru/library/book360.pdf>. – Название с экрана.
6. *Галузинський В. М.* Основи педагогіки та психології вищої школи в Україні : навч. посіб. для викл. та асп. вузів / В. М. Галузинський, М. Б. Євтух ; ІСДО, Київський лінгвістичний ун-т. – К. : ІНТЕЛ, 1995. – 168 с.
7. *Двадцять років становлення і розвитку методичної системи навчання інформатики в школі та педагогічному університеті* / В. Ю. Биков, М. І. Жалдак, Н. В. Морзе, Ю. С. Рамський // Комп'ютер у школі та сім'ї. – 2005. – № 5. – С. 12–19.

8. Кривонос О. М. Організація самостійної роботи бакалаврів математики з курсу “Програмування” [Електронний ресурс] / О. М. Кривонос // Інформаційні технології і засоби навчання. – 2009. – № 2 (10). – Режим доступу до журн. : <http://journal.iitta.gov.ua>.
9. Кривонос О. М. Особливості викладання програмування у вищому навчальному закладі з врахування вимог сучасності / О. М. Кривонос // Вісник Житомирського державного університету імені Івана Франка. – 2011. – Вип. 57. – С. 131–134.
10. Лапчик М. П. ИКТ-компетентность бакалавров образования / М. П. Лапчик // Информатика и образование. – 2012. – № 2. – С. 29–33.
11. Морзе Н. В. Компетентнісні задачі з інформатики / Н. В. Морзе, О. Г. Кузьмінська // Науковий часопис НПУ імені М. П. Драгоманова. Серія № 2: комп’ютерно-орієнтовані системи навчання. – К. : НПУ імені М. П. Драгоманова, 2008. – № 6 (13). – С. 62–69.
12. Раков С. А. Тест загальної навчальної компетентності: основні засади і результати пілотування / С. А. Раков, О. І. Ляшенко // Педагогіка і психологія. – 2012. – № 2. – С. 27-36.
13. Педагогіка / [под ред. Ю. К. Бабанського]. – М. : Просвещение, 1988. – 475 с.
14. Семеріков С. О. Активізація пізнавальної діяльності студентів при вивченні чисельних методів у об’єктно-орієнтованій технології програмування : автореф. дис. на здобуття наук. ступеня канд. пед. наук : спец. 13.00.02 “Теорія та методика навчання (інформатика)” / С. О. Семеріков. – К., 2001. – 20 с.
15. Смолкин А. М. Методы активного обучения : научн.-метод. пособие / А. М. Смолкин. – М. : Высшая школа, 1991. – 176 с.
16. Співаковський О. В. Теорія і практика використання інформаційних технологій у процесі підготовки студентів математичних спеціальностей : [монографія] / О. В. Співаковський. – Херсон : Айлант, – 2003. – 225 с.
17. Спірін О. М. Теоретичні та методичні засади професійної підготовки майбутніх учителів інформатики за кредитно-модульною системою [монографія] / Олег Михайлович Спірін ; [наук. ред. М. І. Жалдак] / Житомирський держ. ун-т ім. Івана Франка. – Житомир : Видавництво ЖДУ ім. І. Франка, 2007. – 300 с.
18. Триус Ю. В. Особливості створення методичної системи навчання основ програмування для підготовки майбутніх інженерів-програмістів / Ю. В. Триус, О. О. Богатирьов, Л. В. Гришко // Вісник Черкаського університету, серія “Педагогічні науки”. – Випуск 35. – Черкаси, 2002. – С. 135–141.
19. Чобітько М. Г. Особистісно орієнтована професійна підготовка майбутнього вчителя: теоретико-методологічний аспект : монографія / М. Г. Чобітько. – Черкаси : Брама, 2006. – 560 с.

Кривонос А. Н. Компетентнісно-орієнтовані задачі в курсі “Програмування”.

В статті розглядається проблема підготовки учителів інформатики на принципах компетентного підходу, в частині, визначені роль та місце компетентно-орієнтованих задач в курсі програмування. Подана схема побудови компетентно-орієнтованої задачі з урахуванням методу проектів. Описано використання описаного підходу в курсі “Програмування” для майбутніх учителів інформатики.

Ключевые слова: компетентний підхід, навчання програмуванню, метод проектів.

Kryvonos O. M. Competence and Directed Task in Course “Programming”.

This paper addresses the problem of training informatics teachers based on competency approach, including the role and the place of competency-based assignments in programming courses. The present scheme of creating a competency-based objectives with regard to the method of projects. The application of this approach in the course “Programming” for future informatics teachers is described.

Keywords: competence approach, teaching programming, project method.