

5. Ягупов В. В. Методологічні основи розуміння та обґрунтування понять «компетентність» і «компетенція» щодо професійної підготовки майбутніх фахівців. *Нові технології навчання: наук.-метод. зб.* Київ, 2011. Вип. 69. Ч. 1. С. 23–29.
6. Зязюн І. А. Філософія поступу і прогнозу освітньої системи. Педагогічна майстерність: проблеми, пошуки, перспективи: монографія. Київ, Глухів: РВВ ГДПУ, 2005. С. 10–18.
7. Лозовецька В. Т. Формування професійної компетентності фахівця сфери послуг і туризму: навч.-метод. посіб. / ред. В. Т. Лозовецька. Київ, 2010. 382 с.

REFERENCES

1. Zimnyaya I. A. Competence approach. What is its place in the modern approaches to education problems? (Theoretical and methodological aspect). *Higher education today*. 2006. No. 8. P. 20-26.
2. Ovcharuk O.V. Competence approach in modern education: world experience and ukrainian perspectives. *Library of educational policy* / Ed. O.V. Ovcharuk. Kyiv. K.I.C., 2004. P. 112.
3. Nychkalo N. H. The new generation of pedagogical staff to continuous vocational education. *Problems of engineering and pedagogical education*. Kharkiv: Ukrainian Engineering and Pedagogical Academy. 2005. No. 10. P. 15-19.
4. Radkevich V. O. Partnership in workers training. *Vocational - technical education*. 2013. No. 4. P. 15.
5. Yahupov V. V. Methodological bases of understanding and substantiation of the notions of "competency" and "competence" concerning professional training of future specialists. Kyiv: *New Learning Technologies*. 2011. No. 69. P.23-29.
6. Zyazyun I. A. Philosophy of movement and forecast of the educational system. Pedagogical skills: problems, searches, perspectives. Kyiv., Glukhiv: RVV GDPU, 2005. 326 p. Kyiv. 2010. 382 p.
7. Lozovetska V. T. Formation of professional competence of specialist in services and tourism sphere. Tutorial manual. Kyiv. 2010. 382 p.

УДК 378.14

АНАЛІЗ ЗАРУБІЖНИХ ДОСЛІДЖЕНЬ МЕТОДІВ ПІДГОТОВКИ МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ ДО ПРОФЕСІЙНОЇ ДІЯЛЬНОСТІ

Круглик В.С.

*Мелітопольський державний педагогічний університет імені Богдана Хмельницького,
вул. Гетьманська, 20, м. Мелітополь, Запорізька обл., Україна*

kryglikvlad@gmail.com

У статті наведені результати аналізу досліджень зарубіжних вчених, присвячених проблемам фахової підготовки майбутніх інженерів-програмістів. Підкреслюється важливість використання досвіду іноземних колег з метою удосконалення навчання у вітчизняних вишах. Особливу увагу приділено методу учнівства, стратегії когнітивного конфлікту і методу побудови ментальних моделей, методам феноменологічного дослідження, методу візуального моделювання програм, методиці навчання студентів програмуванню з використанням веб-орієнтованого середовища і методу виконання вправ із пропусками, методу парного програмування, методу програмного навчання, методу розв'язання професійно-орієнтованих задач з використанням програмних засобів математичного призначення.

Ключові слова: методи навчання, професійна підготовка інженерів-програмістів, метод парного програмування, феноменологічне дослідження, метод побудови ментальних моделей.

АНАЛИЗ ЗАРУБЕЖНЫХ ИССЛЕДОВАНИЙ МЕТОДОВ ПОДГОТОВКИ БУДУЩИХ ИНЖЕНЕРОВ-ПРОГРАММИСТОВ К ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ

Круглик В.С.

*Мелитопольский государственный педагогический университет имени Богдана
Хмельницкого, ул. Гетманская, 20, г. Мелитополь, Запорожская обл., Украина*

kryglikvlad@gmail.com

В статье приведены результаты анализа исследований зарубежных ученых, посвященных проблемам профессиональной подготовки будущих инженеров-программистов. Подчеркивается важность использования опыта иностранных коллег с целью усовершенствования обучения в отечественных вузах. Особое внимание уделяется методу ученичества, стратегии когнитивного конфликта, методу построения ментальных моделей, методам феноменологического исследования, методу визуального моделирования программ, методике обучения студентов программированию с использованием веб-ориентированной среды, методу выполнения упражнений с пропусками, методу парного программирования, методу программного обучения, методу решения профессионально-ориентированных задач с использованием программных средств математического назначения.

Ключевые слова: методы обучения, профессиональная подготовка инженеров-программистов, метод парного программирования, феноменологическое исследование, метод построения ментальных моделей.

THE ANALYSIS OF FOREIGN RESEARCHES OF TRAINING FUTURE SOFTWARE ENGINEERS METHODS FOR PROFESSIONAL ACTIVITIES

Kruglik V.S.

*Melitopol State Pedagogical University named after B. Khmelnytskyi,
Hetmanska str., 20, Melitopol, Zaporizhzhya region, Ukraine*

kryglikvlad@gmail.com

The article deals with the issues of future software engineers training at universities. Traditional methods of future software engineers training are widely used in domestic higher education. These ones are primarily aimed at acquiring knowledge and have no focus on practical skills development. In particular, most programming tasks are based on the use of mathematical apparatus and rarely operate with abstract concepts. However, these concepts form the basis for further study of modern software development paradigms, including object-oriented programming and others. On the other hand, the efficiency of formation of future software engineers' professional competence depends not only on the teacher. The rate of students' activity, their intellectual initiative, original and innovative approaches to solving problems affect on learning quality. Bridging the gap between training content, forms and methods, acquired knowledge, and practical skills forming is essential for future software engineers' professional learning at universities. In this regard, a reasonable selection of forms, methods and means of education is extremely important for the formation of their professional competence and readiness to respond adequately to changes and activate their creative potential. The basis for this choice is profound analysis of the domestic and foreign researches in the field of future software engineers training. The article contains the review of some foreign studies which deal with methodological basis of higher professional education in the field of information technologies. M. Caspersen (Denmark) offers the methodological system of future programmers' professional training which is based on cognitive science and educational psychology, primarily cognitive load theory, cognitive skill acquisition, and cognitive apprenticeship, as well as research in programming methodology. L. Ma (Great Britain) suggests using a constructivist-based teaching approach to improve students' mental models of fundamental programming concepts. This approach integrates a cognitive conflict strategy and program visualization method. Eckerdal A. (Sweden) investigates relationship and mutual dependencies between theoretical and practical learning in the field of programming. The basis of the research is empirical data gathered with the interviews with novice and senior students, and educators. The author suggests methods of qualitative research, phenomenography, and content analysis for study of educational issues. Sorva J. (Finland) presents a pedagogical technique based on the method of visual program simulation for introductory programming course. The aim of this technique is to help novices to understand how software works. Truong N. (Australia) investigates issues of novice programmers learning, such as understanding programming concepts, knowledge of programming language syntax, understanding of compilation errors, and others. The author offers to use a special web-based programming environment which contains exercises with gaps to help novice programmers to learn. Teague D. (Australia) investigates the using of collaborative programming and pair-programming in the introductory CS courses. She considers that these techniques could help to avoid learning failures and be high effective for novice programmers learning. However, it is necessary to follow some important principles to achieve the result. Persianov V. (Russia) discusses the technique of programmed learning. Dubenetskaya E. (Russia) presents the method of professional-oriented tasks. All of the reviewed techniques can be used for for teaching future programmers at Ukrainian universities.

Key words: teaching methods, software engineers training, pair programming method, phenomenographic research, constructing mental models method.

Постановка проблеми в загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями.

У вітчизняній практиці професійної підготовки майбутніх інженерів-програмістів широко застосовуються традиційні методи, які спрямовані передусім на засвоєння вже готових знань і є не роблять акцент на практико-орієнтованості. З нашого досвіду більшість задач з програмування базуються на використанні математичного апарату, а не абстрактних поняттях, які є підґрунтям для вивчення сучасних парадигм розробки програмного забезпечення, зокрема, об'єктно-орієнтованого програмування. Традиційні методи характеризуються роз'єднаністю і ізольованістю. Ефективність формування професійної компетентності майбутніх інженерів-програмістів залежить не тільки від викладача, але і від ступеня активності, інтелектуальної ініціативи самих студентів і від того, як вони знаходять оригінальні і нестандартні підходи при вирішенні поставлених завдань. Подолання розриву між змістом, формами і методами навчання, набутими знаннями й їх практичним застосуванням є найважливішою умовою підготовки майбутніх інженерів-програмістів. У зв'язку з цим, надзвичайно важливе значення має обґрунтований відбір форм, методів і засобів навчання для формування професійної компетентності майбутніх інженерів-програмістів, які готові адекватно реагувати на зміни й активізувати свій творчий потенціал. Основою такого вибору є досконалий аналіз досвіду освітньої діяльності вітчизняних й іноземних вищих навчальних закладів у галузі підготовки майбутніх інженерів-програмістів.

Аналіз останніх досліджень і публікацій, у яких започатковано вирішення цієї проблеми і на які спирається автор. Зарубіжні науковці виділяють різні методи, підходи, технології та засоби професійної підготовки майбутніх інженерів-програмістів. Зокрема, обґрунтуванню методів і засобів професійної підготовки майбутніх інженерів-програмістів присвячені роботи Дж. Адамса (J. Adams), Й. Беннедсена (J. Bennedsen), А. Даннера (A. Danner), Г. Ескердал (A. Eckerdal), М. Е. Касперсена (M. E. Caspersen), Л. Ма (L. Ma), Р. Пучера (R. Pucher), Ю Сорви (J. Sorva), Д. Тіг (D. Teague), Н. Труонг (N. Truong), Ф. Ханна (Ph. Hanna) та ін.

Виділення невирішених раніше частин загальної проблеми, яким присвячується стаття. Вітчизняні дослідники активно звертаються до досвіду іноземних колег у процесі пошуку шляхів удосконалення вищої професійної освіти в галузі інформаційних технологій. Разом із тим, зарубіжний науковий доробок у цьому напрямі є надзвичайно багатим і постійно зростає, тому доцільним є систематичний аналіз наукових праць окремих вчених з метою встановлення можливості впровадження отриманих ними результатів у практику українських вишів.

Формулювання мети статті (постановка завдання). Проаналізувати роботи іноземних дослідників з метою узагальнення зарубіжного досвіду вибору доцільних методів професійної підготовки майбутніх інженерів-програмістів у вищих навчальних закладах.

Виклад основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів. Процес інформатизації суспільства висуває високі вимоги до професійної підготовки майбутніх інженерів-програмістів. Перед системою професійної освіти стоїть завдання підготувати творчо мислячу людину, яка володіє дослідницькими вміннями і навичками, здатна орієнтуватися в потоці наукової інформації і сучасних інформаційних технологіях, готова до постійного підвищення загальної і професійної компетентності, до можливої перекваліфікації та перепідготовки.

На цьому наголошують вітчизняні дослідники, які займаються розробленням способів удосконалення професійної підготовки майбутніх інженерів-програмістів. Наприклад, у роботі [1] В. В. Осадчий і С. С. Шаров наводять перелік вимог до фахівців з адміністрування баз даних на сучасному ринку праці і обґрунтовують умови успішного працевлаштування випускників вишів за цією спеціальністю, серед яких на першому місці є якісна фахова освіта.

Для цього потрібен відповідний арсенал форм, методів і засобів професійної підготовки.

М. Е. Касперсен (Данія), визнаючи програмування одним з семи грандіозних завдань у галузі комп'ютерної освіти, обґрунтовує концептуальні засади формування у майбутніх програмістів навичок, необхідних для системної розробки програм. У запропонованій методичній системі автор об'єднує найкращі практики програмування, застосовуючи у вступному курсі програмування для першокурсників методологію навчання поетапному вдосконаленню програм, що включає тест-керовану розробку, рефакторинг і підхід на основі моделей об'єктно-орієнтованого програмування. Увесь процес ґрунтується на положеннях когнітивної науки та педагогічної психології, передусім на когнітивній теорії навантаження, когнітивному набутті навичок і когнітивному учнівстві, а також дослідженнях у галузі методології програмування. Когнітивна теорія навантаження – це особлива сфера когнітивної науки та педагогічної психології, що вивчає, як зменшити навантаження на робочу пам'ять для оптимізації навчання [2, с. 21].

Згідно з методологічною системою, запропованою М. Е. Касперсеном, навчання програмуванню здійснюється з використанням комплексу навчальних засобів (підручника, вправ і завдань та відеоматеріалів). Підручник створюється на основі проблемно-орієнтованого підходу і відповідно до педагогічних принципів навчання об'єктно-орієнтованого програмування на основі *методу учнівства*. Цей метод передбачає, що студенти спочатку спостерігають за тим, як викладач демонструє використання нових методів або конструкцій для створення програми, потім застосовують новий матеріал у процесі розробки проекту під керівництвом викладача і на останньому етапі пишуть програми самостійно. Вправи і завдання містять вказівки, які допомагають студентам на окремих етапах розробки програми. На початкових етапах навчання подаються докладні вказівки до виконання завдань, а на завершальних – лише короткі рекомендації. Використання навчальних відеоматеріалів, які демонструють принципи і прийоми програмування, а також процес розробки програми, дозволяє диференціювати процес навчання: сильні студенти можуть пропускати те, що їм зрозуміло, а інші можуть зупиняти і переглядати відео кілька разів.

Для нашого дослідження важливим є зроблений автором акцент на методі учнівства та застосуванні підручника, вправ, завдань і відеоматеріалів з метою професійної підготовки майбутніх інженерів-програмістів.

З метою удосконалення професійної підготовки майбутніх програмістів Л. Ма (Велика Британія) [3] пропонує впровадження конструктивістської моделі навчання, яка поєднує пізнавальну стратегію конфліктів і візуалізації програм для поліпшення ментальних моделей студентів. Ідея конструктивізму ґрунтується на тому, що студенти мають активно будувати знання шляхом об'єднання емпіричного світу з існуючими когнітивними структурами, а не пасивно поглинати знання з лекцій або підручників (М. Бен-Арі). Досвід буденного життя майбутніх програмістів не може допомогти їм будувати життєздатні ментальні моделі комп'ютерних артефактів, тому студенти часто зловживають попередніми знаннями або використовують інтуїтивні моделі, приречені на нежиттєздатність. У зв'язку з цим на ранній стадії необхідно чітко допомогти студентам навчитися створювати життєздатні ментальні моделі, щоб уникнути безсистемної їх побудови самими студентами. Ступінь і правильність запропонованої під час навчання моделі повинні бути зрозумілими для актуального рівня студентів, тому спочатку пропонується пояснювати базові моделі і згодом переходити до пояснення абстрактних концепцій об'єктно-орієнтованого програмування.

Конструктивізм підкреслює важливість ментальних моделей у галузі викладання комп'ютерних наук, де ментальні моделі є заснованими на попередньому досвіді ідеями, стратегіями, способами розуміння, що існують у розумі людини і спрямовують її дії. Однією з ключових стратегій навчання на основі конструктивізму є *стратегія когнітивного конфлікту*, який виникає у випадку несумісності представлень. Відповідно до цієї стратегії

майбутні програмісти, отримуючи знання, прагнуть до узгодженості внутрішньої системи знань, уявлень, переконань та відчують дискомфорт у випадку наявності протиріч і неузгодженості. У зв'язку з цим, спочатку слід допомогти студентам усвідомити, що існуючі уявлення є ненауковими, а потім побудувати наукові знання і уявлення з програмування. Використання технологій візуалізації дозволяє програмістам побудувати програму без написання коду, що відповідно дозволяє студентам побудувати потрібні ментальні моделі. Також цьому сприяє й візуалізація програм, що допомагає проілюструвати програми після того, як вони були створені у традиційній текстовій манері.

Результати дослідження показують, що запропонована Л. Ма модель навчання є ефективною для підвищення інтересу студентів і формування у них здатності до побудови життєздатних ментальних моделей. Виділені автором *метод стратегії когнітивного конфлікту* та *метод побудови ментальних моделей*, на нашу думку, доцільно додати до перспективних методів професійної підготовки майбутніх інженерів-програмістів.

Г. Ескердал (Швеція) у своїй праці [4] розглядала феноменологічний аналіз розуміння студентами ключових понять об'єктно-орієнтованого програмування. З погляду феноменології процес навчання сприяє баченню чогось у новому світлі, дозволяє звернути увагу на явища, що раніше сприймалися як звичайна річ, як на нові для студента. Основне завдання дослідника під час феноменологічного дослідження результатів опанування початкового курсу об'єктно-орієнтованого програмування полягає у з'ясуванні того, який зміст студенти вкладають в абстрактні поняття, зокрема «об'єкт» і «клас», а також у поняття «уміння програмувати».

Основним методом феноменологічного дослідження є *метод інтерв'ю*. Г. Ескердал наводить різноманітні питання, спрямовані на виявлення розуміння студентами сутності вивчення програмування, важливих моментів курсу, цілей навчання, а також визначення найскладніших положень курсу. У дисертації подані конкретні приклади того, як теорія варіації і закономірності змін можуть бути використані у професійній підготовці майбутніх програмістів з метою підвищення ефективності практичного і теоретичного навчання.

У своїй дисертаційній роботі Ю. Сорва [5] (Фінляндія) обґрунтовує і експериментально перевіряє педагогічну технологію, мета якої полягає в наданні студентам цілеспрямованої допомоги у процесі вивчення основ програмування. Вона ґрунтується на візуальному моделюванні програм (visual program simulation – VPS) і залучає студентів-програмістів до інтерактивної симуляції, протягом якої вони беруть на себе роль комп'ютера як виконавця програми. Студенти використовують візуалізацію так званої «уявної машини», абстрактного комп'ютера, щоб проілюструвати процеси, які відбуваються в пам'яті, коли комп'ютер обробляє програму. Таке моделювання допомагає програмістам-початківцям навчитися розмірковувати про виконання програми, а також розвивати навички, необхідні для вивчення наступних курсів з програмування. Автор доводить, що VPS сприяє ефективному навчанню, підвищуючи пізнавальну активність за допомогою візуалізації. Технологія може застосовуватися на практиці як програмне забезпечення візуалізації, автоматично оцінювати навички студентів з моделювання і надавати зворотний зв'язок, який особливо важливий у великих класах, типових для початкових курсів.

Цінність роботи Ю. Сорви у рамках нашого дослідження полягає в тому, що він пропонує розглядати *візуальне моделювання програм* як метод і подає його основи з погляду теорії навчання. Цей метод, на його переконання, допомагає багатьом студентам навчитися програмуванню на початкових етапах. Проте, на нашу думку, варто бути обережним при використанні візуалізацій, адже вони подають процес програмування у спрощеному вигляді. Використовуючи цей метод у навчальному процесі, слід акцентувати увагу студентів на тому, що візуалізація – це одна з частин розробки програм, але не основна.

У роботі Н. Труонг [6] (Австралія) наводиться методика навчання студентів програмуванню з використанням веб-орієнтованого середовища. Ця методика спрямована на подолання

труднощів, які виникають у студентів перших курсів під час роботи у стандартних середовищах програмування, а саме: 1) встановлення та налаштування середовища програмування; 2) використання редактора середовища програмування; 3) розуміння питань, пов'язаних з програмуванням, і знання синтаксису мови програмування; 4) розуміння помилок компіляції; 5) налагодження. З метою подолання цих труднощів дослідник вважає доцільним використовувати веб-орієнтовані середовища програмування, зокрема, ELP.

Методика навчання програмуванню ґрунтується на використанні саме цього середовища програмування, що у варіанті для студентів є вправами типу «заповнити пропуски». Студентам не потрібно витрачати час на навчання використанню редактора середовища програмування, тому вони можуть безпосередньо зосередитися на вивченні програмування. Вправи розташовані за збільшенням складності. Для студентів передбачена можливість працювати у трьох режимах: Hint (порада), Solution (рішення) і My Program (моя програма). Середовище ELP надає студентам миттєвий зворотний зв'язок з детальною інформацією про помилки у написаній програмі і можливістю їх виправити.

Викладач може створювати і видаляти пропуски у вправі, змінювати поради у режимі Hint, вносити зміни в текст програми в режимі My Program, має можливість отримувати детальну інформацію про виконання студентами вправ: кількість доробок у програмі, інтервал часу між спробами компіляції, успішність кожної спроби. Проаналізувавши цю інформацію, він може зробити висновок про уміння студентів програмувати і визначити, кому необхідно надати додаткову допомогу.

Д. Тіг [7] (Австралія) пропонує метод парного програмування, який полягає в тому, що два програмісти спільно працюють над вирішенням завдання за одним комп'ютером. Один з них виконує роль «ведучого», тобто вводить код і вирішує тактичні проблеми, інший мислить стратегічно, ставить питання і відстежує помилки в коді. Перевагами такого методу, на думку дослідниці, є те, що студенти роблять меншу кількість помилок, краще розуміють код, який стає більш якісним, вчать один в одного, оволодівають навичками креативності, вчать швидко вирішувати проблеми. Водночас, вона називає недоліки цього методу, пов'язані з такими факторами: 1) відмінності у здібностях між студентами, що працюють у парі; 2) розуміння студентами правил парного програмування і готовність дотримуватися їх; 3) відвідуваність занять студентами; 4) необхідність контролю з боку викладачів за дотриманням студентами правил парного програмування.

Для послаблення негативного впливу цих недоліків Д. Тіг пропонує керуватися такими принципами: 1) бажано, щоб пару складали студенти зі схожими здібностями; 2) проводити лабораторні заняття, на яких студенти працюють у парах; 3) студентам, що створюють пару, повинно бути зручно сидіти поруч, вони обидва повинні мати легкий доступ до монітора, миші і клавіатури; 4) склад пар необхідно протягом семестру змінювати; 5) учні повинні розуміти, що про проблеми з партнером потрібно негайно повідомити викладачеві, щоб дати йому шанс виправити ситуацію; 6) програмісти в парі повинні працювати для досягнення спільної мети; 7) краще пропонувати завдання, які можуть бути виконані на щотижневому лабораторному занятті, ніж очікувати, що студенти будуть зустрічатися в позанавчальний час; 8) встановити стандарт кодування, якого повинен дотримуватися кожен студент; 9) дотримуватися принципу співпраці, взаємної поваги і загальної відповідальності у відносинах між обома студентами в парі і викладачем; 10) контролювати відвідуваність і запізнення, щоб студенти не залишалися без партнера; 11) здійснювати контроль над балансом між індивідуальною і спільною роботою студентів; 12) викладачі повинні спонукати пари до самостійного пошуку відповідей на питання, а не надавати їх.

В. В. Персіанов у дисертації [8] зосереджує увагу на навчанні майбутніх інженерів-програмістів моделюванню інформаційних процесів і пропонує застосовувати *метод програмного навчання*. Автором розроблені елементи теорії діалогових інформаційних систем, що дозволяють пов'язати концептуальне уявлення користувача з проектованою

інформаційно-логічною моделлю. Науковець пропонує організувати вивчення комп'ютерного моделювання на принципах програмованого навчання у процесі вирішення завдань з прикладної сфери з подальшим узагальненням, поширенням, інтерпретацією результатів і отриманих даних. У процесі навчання розробка навчальних моделей базується на застосуванні сучасних інструментальних засобів редагування, програмування і моделювання, інтегрованих програмних середовищ, автоматизованих навчальних систем і комп'ютерних моделей соціально-економічних систем.

З метою розвитку у майбутніх інженерів-програмістів навичок застосування математичних методів у процесі розробки, супроводу і обслуговування програмного забезпечення О. Р. Дубенецька [9] пропонує застосовувати *метод розв'язання професійно-орієнтованих задач*, зокрема для обробки статичного і динамічного інформаційного контенту, обчислення характеристик якості програмного продукту галузевої спрямованості, розрахунку вимірювань, отриманих при впровадженні та адаптації програмного забезпечення галузевого спрямування, побудови інформаційно-логічних моделей програмних продуктів, здійснення логічних операцій при визначенні ризиків у проектній діяльності тощо. Дослідниця пропонує застосовувати як засоби навчання програмні продукти математичного призначення, наприклад, Mathematica, Maple, Mathcad, які дозволяють виконувати складні розрахунки з графічним представленням отриманих результатів, будувати схеми і графіки для створення електронних документів технічного характеру у стандартному стильовому оформленні, розробляти програмні модулі для вирішення невеликих прикладних завдань, аналізувати статистичні дані та подавати їх у вигляді таблиць, гістограм, графіків тощо.

Висновки за дослідженням і перспективи подальших розвідок у цьому напрямі.

У процесі дослідження сформульованої проблеми були вивчені метод учнівства (М. Е. Касперсен), стратегія когнітивного конфлікту і метод побудови ментальних моделей (Л. Ма), методи феноменографічного дослідження (Г. Ескердал), метод візуального моделювання програм (Ю. Сорва), методика навчання студентів програмуванню з використанням веб-орієнтованого середовища і метод виконання вправ із пропусками (Н. Труонг), метод парного програмування (Д. Тіг), метод програмного навчання (В. В. Персіанов), метод розв'язання професійно-орієнтованих задач з використанням програмних засобів математичного призначення (О. Р. Дубенецька).

З метою формування компетентності з програмування ми, ґрунтуючись на дослідженнях зазначених науковців, вважаємо за доцільне використовувати зазначені методи навчання у процесі фахової підготовки майбутніх інженерів-програмістів у вищих навчальних закладах. Подальші дослідження спрямовані на удосконалення зазначених методів навчання й розробку загальної концепції педагогічної системи підготовки майбутніх інженерів-програмістів до професійної діяльності, до складу якої мають бути включені ці методи.

ЛІТЕРАТУРА

1. Осадчий В. В., Шаров С. В. Сертифікація професійних знань і умінь студентів у сфері баз даних як необхідна умова подальшого успішного працевлаштування. *Проблеми інженерно-педагогічної освіти*. 2015. № 46. С. 49-56.
2. Caspersen M. E. Educating novices in the skills of programming: dissertation for the PhD degree. Aarhus, 2007. 311 p.
3. Ma L. Investigating and improving novice programmers' mental models of programming concepts: thesis for the degree of doctor of philosophy. Glasgow, 2007. 208 p.
4. Eckerdal A. Novice Programming Students' Learning of Concepts and Practise: dissertation for the degree of Doctor of Philosophy. Uppsala, 2009. 76 p. <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A173221&dswid=-6013>.

5. Sorva J. Visual Program Simulation in Introductory Programming Education: Doctoral dissertation for the degree of Doctor of Science in Technology. Aalto, 2012. 422 p. <http://lib.tkk.fi/Diss/2012/isbn9789526046266/isbn9789526046266.pdf>.
6. Truong N. A web-based programming environment for novice programmers: dissertation for the degree of doctor of philosophy. Queensland, 2007. 286 p. http://eprints.qut.edu.au/16471/1/Nghi_Truong_Thesis.pdf.
7. Teague D. Pedagogy of introductory computer programming: a people-first approach: thesis for the degree of master of information technology (research) Queensland, 2011. 129 p. http://eprints.qut.edu.au/46255/1/Donna_Teague_Thesis.pdf.
8. Персианов В. В. Теория и практика обучения прикладной информатике в педагогических вузах на моделях социально-экономических систем: дис. ... доктора пед. наук. Тула, 1997. 311 с.
9. Дубенецкая Е. Р. Теоретико-методические подходы к обучению техников-программистов применению математических методов в профессиональной деятельности: Автореф. дис. Москва, 2014. 19 с.

REFERENCES

1. Osadchyj, V., and Sharov S. (2015), «The certification of professional knowledge and skills of students in the field of database as a prerequisite for further successful employment», *Problemy inzh.-ped. osvity*, no 46, pp. 49-56. [in Ukrainian]
2. Caspersen, M. E. (2007), «Educating Novices in The Skills of Programming» (PhD thesis), Aarhus University, Department of Computer Science. [in English]
3. Ma, L. (2007), «Investigating and improving novice programmer' mental models of programming concepts» (PhD thesis), University of Strathclyde, Department of Computer & Information Sciences. [in English]
4. Eckerdal, A. (2009), «Novice Programming Students' Learning of Concepts and Practise» (PhD thesis), Uppsala, Sweden, Acta Universitatis Upsaliensis, available at: www.diva-portal.org/smash/record.jsf?pid=diva2%3A173221&dswid=-6013 (access May 02, 2017). [in English]
5. Sorva, J. (2012), «Visual Program Simulation in Introductory Programming Education» (Doctor of Science in Technology thesis), Espoo, Finland, Aalto University School of Science, available at: <https://aaltodoc.aalto.fi/handle/123456789/3534?locale-attribute=en> (access May 02, 2017). [in English]
6. Truong, N. (2007), «A web-based programming environment for novice programmers» (PhD thesis), Queensland University of Technology, available at: http://eprints.qut.edu.au/16471/1/Nghi_Truong_Thesis.pdf (access May 02, 2017). [in English]
7. Teague, D. (2011), «Pedagogy of Introductory Computer Programming: A People-First Approach» (Master of Information Technology thesis), Queensland University of Technology, available at: http://eprints.qut.edu.au/46255/1/Donna_Teague_Thesis.pdf (access May 02, 2017). [in English]
8. Persianov, V. V. (1997), «Theory and practice of teaching applied computer science in pedagogical universities on the models of socio-economic systems», Doctor's thesis (Pedagogy), Tula, Russia. [in Russian]
9. Dubenetskaia, E. R. (2014), «Theoretical and methodological approaches to the training of technicians-programmers in applying mathematical methods in professional activity», Thesis abstract for Cand. Sc. (Pedagogy), Moscow, Russia. [in Russian]