

РОЗДІЛ 9. ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ В ОСВІТІ

ТЕОРЕТИЧНІ АСПЕКТИ ФОРМУВАННЯ ІНВАРІАНТНОЇ ЧАСТИНИ ЗМІСТУ НАВЧАННЯ МОВ І ТЕХНОЛОГІЙ ПРОГРАМУВАННЯ

THE THEORETICAL ASPECTS OF THE FORMATION OF THE INVARIATIVE PART OF THE CONTENT OF TEACHING LANGUAGES AND TEACHING PROGRAMMING'S TECHNOLOGIES

Стаття присвячена одній із актуальних проблем фахової підготовки майбутніх інженерів-педагогів педагогічних вишів з урахуванням специфіки динамічної галузі ІКТ. Проблема зацікавлення знань може бути вирішена шляхом кількісних і якісних змін змісту дисциплін. У цих умовах основним завданням викладача є пошук незмінних елементів дисципліни «Мови та технології програмування» – інваріантів, які дадуть змогу перейти від кількісних змін у методиці навчання програмування до якісних, що допоможе вдосконалити методику навчання мов і технологій навчання. Розкрито протиріччя між необхідністю вдосконалення процесу формування змісту навчальних дисциплін з використанням принципу інваріантності й недостатньо сформованими теоретичними підходами до формування змісту; потребою в сучасних засобах навчання мов і технологій програмування та невідповідністю наявних засобів навчання принципу інваріантності. Розглянуто процес формування змісту навчальних дисциплін засобами моделювання адаптивних систем, які дає змогу синтезувати системи, які мають можливість змінювати параметри керування та структуру системи. Для досягнення необхідного рівня якості професійної підготовки майбутніх інженерів-педагогів комп'ютерного профілю повинні бути вирішені такі завдання: урахування процесів швидкої еволюції технологій, продуктів і послуг; забезпечення можливості швидкої зміни змісту дисциплін професійної підготовки; розроблення нових технологій формування змісту підготовки, враховуючи специфіку динамічної галузі ІКТ. Основна увага зосереджується на процесі навчання, де регулювання змінює не тільки керуючу інформацію, а й зміст самої дисципліни. Доведено доцільність використання інваріантного принципу. На прикладі навчання мов і технологій програмування розкрито пошук інваріантів дисциплін, що є першочерговим завданням для формування змісту, й описано механізми адаптації варіативної частини змісту дисципліни.

Ключові слова: фахова підготовка, майбутні інженери-педагоги, інваріантний принцип, зміст дисциплін, ІКТ, мови й технології програмування.

The article is devoted to the one of the most burning problems of the professional training of the future engineers-pedagogues at high educational institutions. Considering the specifics of the dynamical scientific field of the ICT. Within

such conditions the main task of the pedagogue is the search of the unchangeable elements of the discipline "language and Technologies of the programming" – invariants which allow to transfer from the quantitative changes within the methodic of teaching programming to the qualitative changes which will allow to improve the methodic of teaching languages and technologies of programming. There was revealed the contradiction which exists between the necessity of improvement of the process of formation of the educational disciplines' content and the insufficiently formed theoretical approaches to the content's formation; between the demand in modern ways of teaching languages and programming languages and inappropriateness of the existing methods of teaching programming to the invariance. The problem of the backwardness of the reliable and relevant educational information of the disciplines within the field of the computer technologies could be solved and accomplished with the help of the qualitative and quantitative transformations of the disciplines' content. There is revealed the process of the formation of the educational disciplines by means of modeling with the help of the adaptive systems which gives the opportunity to synthesize the systems which have the opportunity to change the parameters of the control options and the system's structure. For the achievement of such necessary level of the professional training of the future engineers-pedagogues of the computer profile, there should be solved certain tasks: consideration of the processes of the fast evolution of technologies, products and services; providing of the opportunity of the fast change of disciplines (of the profile training); development of new technologies of formation of the training's content, considering the specifics of the dynamical field of the informational computer technologies. The main attention is concentrated on the educational process within which the regulation changes not only the controlling information but also it changes the structure of the system. It is proved the advisability of the implementation of the invariant principle. On an example of teaching language and technologies of programming there were revealed the research of the unchangeable elements – the disciplines' invariants, which are the most prior task for the content's formation; also there was revealed and described the mechanism of adaptation of the invariant part of the discipline's content.

Key words: professional training, future engineers-pedagogues, invariant principle, the disciplines' content, ICT, programming languages and technologies of programming.

УДК 378.09.011.3–051-057.21:004
DOI <https://doi.org/10.32843/2663-6085-2019-13-2-42>

Алексєєва Г.М.,
канд. пед. наук, доцент кафедри комп'ютерних технологій в управлінні та навчанні й інформатики Бердянського державного педагогічного університету

Кравченко Н.В.,
канд. фіз.-мат. наук, доцент кафедри комп'ютерних технологій в управлінні та навчанні й інформатики Бердянського державного педагогічного університету

Горбатюк Л.В.,
канд. пед. наук, доцент кафедри комп'ютерних технологій в управлінні та навчанні й інформатики Бердянського державного педагогічного університету

Антоненко О.В.,
канд. техн. наук, доцент кафедри комп'ютерних технологій в управлінні та навчанні й інформатики Бердянського державного педагогічного університету

Овсянніков О.С.,
канд. пед. наук, доцент кафедри комп'ютерних технологій в управлінні та навчанні й інформатики Бердянського державного педагогічного університету

Постановка проблеми в загальному вигляді. Стрімкий розвиток інформаційних технологій спровокував процеси, глибоко суперечливі за своїм

характером, які ще більше ускладнили процес навчання саме інформаційно-комунікаційний технологій. Із кожним роком поглиблюється проблема

запізнювання вірогідності навчальної інформації дисциплін у галузі комп'ютерних технологій.

Проблема запізнення знань може бути вирішена шляхом кількісних і якісних змін змісту дисциплін. Найчастіший спосіб вирішення цієї проблеми – це постійне оновлення змісту дисциплін. І цим шляхом іде більшість вишів України. На підтвердження цього факту в мережі Інтернет можна зустріти велику кількість методичних розробок для кожної предметної галузі комп'ютерних технологій, для кожної технології, для кожного програмного продукту, а інколи для кожної його версії. Але шалені зусилля викладачів не розв'язують проблему, а лише її загострюють. Настала необхідність переходу від кількісних до якісних змін.

Аналіз останніх досліджень і публікацій.

Проблема професійної підготовки майбутніх інженерів-педагогів досліджувалася багатьма вченими, зокрема С. Артюхом, А. Ашеровим, О. Бєловою, Н. Брюхановою, Е. Зеєром, О. Коваленко, М. Лазарєвим [4] та іншими. Аналіз праць показав відсутність прийнятних технологій формування змісту дисциплін професійної підготовки майбутніх інженерів-педагогів комп'ютерного профілю, які б ураховували специфіку динамічної галузі ІКТ.

Виділення не вирішених раніше частин загальної проблеми. Процес формування змісту навчальних дисциплін повинен моделюватися адаптивними системами. Особливість цих систем полягає в наявності адаптивного керування, що дає змогу синтезувати системи, які мають можливість змінювати параметри керування та структуру системи. Тобто в процесі навчання регулювання змінює не тільки керуючу інформацію, а й зміст самої дисципліни. Доцільно використовувати також інваріантний принцип.

Отже, актуальність теми статті зумовлена протиріччями між необхідністю вдосконалення процесу формування змісту навчальних дисциплін із використанням принципу інваріантності й недостатньо сформованими теоретичними підходами до формування змісту; потребою в сучасних засобах навчання мов і технологій програмування та невідповідністю наявних засобів навчання принципу інваріантності.

Мета статті полягає в аналізі процесу формування змісту навчальних дисциплін із використанням принципу інваріантності й теоретичному обґрунтуванні структури і змісту навчання мов і технологій програмування, побудованому на принципі інваріантності.

Виклад основного матеріалу. Розглянемо принцип інваріантності формування змісту професійної підготовки майбутніх інженерів-педагогів комп'ютерного профілю. Розвиток сфери інформаційно-комунікаційних технологій вимагає принципово нових підходів до формування структури і змісту системи професійної підготовки майбут-

ніх інженерів-педагогів комп'ютерного профілю, оскільки вимоги до змісту підготовки фахівців істотно змінюються в процесі самої підготовки. Традиційні методи розроблення структури та змісту підготовки фахівців, характерні для усталених галузей науки, техніки й економіки, малоефективні для високо динамічної сфери ІКТ [2].

Для досягнення необхідного рівня якості професійної підготовки майбутніх інженерів-педагогів комп'ютерного профілю повинні бути вирішені такі завдання: урахування процесів швидкої еволюції технологій, продуктів і послуг; забезпечення можливості швидкої зміни змісту дисциплін професійної підготовки; розроблення нових технологій формування змісту підготовки, враховуючи специфіку динамічної галузі ІКТ.

Процес формування змісту навчальних дисциплін повинен моделюватися адаптивними системами. Особливість цих систем полягає в наявності адаптивного керування, що дає змогу синтезувати системи, які мають можливість змінювати параметри керування та структуру системи. Тобто в процесі навчання регулювання змінює не тільки керуючу інформацію, а й зміст самої дисципліни.

Наприклад, у роботі П. Федорука розглядається модель системи дистанційного навчання, яка базується на визначених засадах функціонування адаптивного викладача, що дає змогу також автоматизувати реалізацію адаптивного навчального процесу в навчальній системі. Крім цього, доцільно використовувати інваріантний принцип.

Термін «інваріант» широко використовується в науці й позначає щось незмінне. Конкретне значення терміна залежить від предметної галузі, в якій він використовується. Інваріанти можуть становити множину об'єктів і/або певну структуру. Так, інваріанти як незмінні наукові вимоги, на основі яких реалізується особистісно орієнтований підхід до виховання студента, розглядаються в працях І. Бєха [1]. Інваріанти професійної компетентності в статті М. Ільязова розглядаються як теоретичний конструкт на основі компетенцій [3]. На думку М. Левшина, під час проектування певної педагогічної системи обов'язково необхідно визначити її підґрунтям інваріанти, що надає інваріантності статусу принципу проектування педагогічних систем [5].

Пошук незмінних елементів – інваріантів – у кожній дисципліні є першочерговим завданням для формування змісту дисциплін професійної підготовки майбутніх інженерів-педагогів комп'ютерного профілю. І лише після виокремлення інваріантів можна будувати механізми адаптації варіативної частини змісту дисципліни.

Розглянемо це на прикладі. Освоєння програмування можна характеризувати трьома історичними етапами, що відрізняються рівнем складності одержання професійної кваліфікації [6]. Ранньому етапу характерний дуже високим рівнем

складності. Програмування здійснюється мовою кодів комп'ютера, пізніше мовою асемблера. Від програміста потрібне глибоке знання архітектури комп'ютера: системи команд, методів керування пам'яттю й керування обчислювальним процесом тощо (професія, що з'явилася в ті часи, – «програміст» – належить до елітних, загадкових).

З появою мов програмування високого рівня почався другий етап, що характеризується різким зниженням рівня початкової професійної підготовки. Ідеї мови програмування стали доступні широкому колу різних фахівців, як наслідок, програмістська діяльність стала складником багатьох професій. У цей період створено величезну кількість досить простих програм. Потреба в автоматизації нескладних (за сучасними мірками) завдань була задоволена. На рубежі 1980-х років з'явилося поняття «криза програмування». За прогнозами тих років, до 2005–2010 років для задоволення потреби в автоматизації праці все людство мало перемкнутися на програмування. У відповідь на виниклі проблеми наука програмування висунула безліч нових ідей і переглянула старі: вивчена й розвинена концепція типів цих мов програмування; з'явилося поняття «інкапсуляція», за допомогою якої вдалося впорядкувати обчислювальні структури; використання поліморфізму й правил іменування ідентифікаторів додало програмам властивість модифікованості. Перехід до технології персональних комп'ютерів – інструмент, що зробив їх масовими, сприяв створенню спеціальних засобів керування обчислювальним процесом, які одержали загальну назву «дружній інтерфейс». У практику програмування введені концепції подібного програмування й захищених від помилок програмних кодів. Нарешті, сформувалася ідея об'єктно-орієнтованого програмування.

З появою об'єктно-орієнтованого програмування розмови про «кризу програмування» поступово припинилися. Почався новий, третій, етап у складності одержання базових знань для зайняття програмістською діяльністю. Тепер досить поверхове освоєння приймань програмування дає змогу створювати нескладні програми. Однак професіонал, як правило, знайде в таких програмах безліч недоліків. Найчастіше якість таких програм не задовольняє сучасним вимогам надійності, модифікованості, як наслідок, час життя програми буде невеликим. Легкість одержання працюючої програми створює в програмістів ілюзію достатньої підготовленості. Однак для створення, яке відповідає всім сучасним вимогам складного програмного виробу, необхідно глибоке розуміння ідей, принципів об'єктно-орієнтованого програмування, реалізації їх у системі програмування, пристрою об'єктно-орієнтованої програми, об'єктно-орієнтованої технології створення програмних систем [2].

Отже, програмування стало водночас і простим (відносно), і складним, а навчальний процес з вивчення програмування нагромадив низку протиріччя об'єктивного й суб'єктивного характеру. Протиріччя пов'язано з кількістю мов програмування, що поповнюється щороку. Деякими мовами вміє користуватися тільки невелика кількість їх розробників, інші стають відомі мільйонам людей. Із кожним роком поглиблюється проблема запізнення та інколи зовсім втрати вірогідності навчальної інформації дисциплін у галузі програмування. Аналіз літературних джерел дав змогу зробити висновки щодо актуальності проблеми навчання програмування. Так, у роботі О. Спіріна й Т. Вакалюка на основі аналізу наукової літератури показано, що використання web-орієнтованих технологій під час навчання основ програмування майбутніх учителів інформатики є провідним напрямом діяльності. Проаналізовано основні можливості використання різних видів web-орієнтованих технологій під час навчання основ програмування, зокрема систем із проведення спортивних змагань із програмування, та хмарних технологій. В. Кругликом та іншими проаналізовано міждисциплінарний підхід у професійній підготовці. Авторами статті розглядаються проблеми професійної підготовки майбутніх програмістів у контексті впровадження міждисциплінарного підходу. Наведено опис специфіки майбутньої професійної діяльності програмістів на прикладі трьох типів професій за об'єктом праці («людина – техніка», «людина – знакова система», «людина – людина»). Застосування міждисциплінарного підходу в професійній підготовці майбутніх програмістів розглядається як частина процесу формування їхньої професійної компетентності через реалізацію зв'язків із дисциплінами гуманітарної, соціально-економічної, математичної та природничо-наукової підготовки. О. Мельник у статті розглянув основні теоретичні й методологічні положення методичної системи навчання мов програмування під час підготовки фахівців у вищих навчальних закладах I–II рівнів акредитації. Наведено основні переваги та недоліки використання деяких мов програмування під час розв'язання певних завдань. А. Макаренко, А. Бондарчук, Є. Хотинський, В. Лимар у статті висвітлили результати досліджень сучасних мов програмування, які використовуються з метою розроблення програм для окремих вузлів сучасних систем надання телекомунікаційних послуг. Визначено, що одним із найбільш перспективних підходів до створення сучасних систем надання телекомунікаційних послуг є технологія OpenFlow Softswitch. Показано, що на основі мов програмування C та Java розроблено спеціальні мови програмування мереж – Frenetic, NetCore й Nettle. Найбільш популярною мовою програмування серед програмістів, у тому числі й сучасних

систем надання телекомунікаційних послуг, є мова Java. П. Мерзликін на основі досвіду провідних світових вишів обґрунтував добір мови програмування для вступних курсів програмування, описав досвід упровадження Python як першої мови програмування на першому курсі спеціальності «Інформатика». Н. Праворська розглянула завдання оцінювання переваг і недоліків різних мов програмування й обґрунтування вибору необхідної мови для виконання конкретної поставленої математичної задачі.

Однак, незважаючи на велику кількість робіт, є низка проблем, які ще досить не вирішені. Перелічимо основні проблеми, що виникають у навчальному процесі під час освоєння програмування.

По-перше, об'єктно-орієнтоване програмування орієнтоване передусім на створення складних програм, а студент, виконуючи лабораторні роботи, вирішує прості завдання. У курсових роботах студент найчастіше вирішує локальні завдання. Керівником проекту є викладач, на ньому лежить відповідальність за проект загалом. У результаті більша частина студентів не одержує досвід створення складних програм і, тим більше, керівництва процесом їх розроблення.

По-друге, об'єктно-орієнтоване програмування створює гарні можливості модифікації програм, але ці можливості закладаються ще на етапі проектування програми. У студента відсутня мотивація забезпечення модифікованості програми, тому що стиль його роботи – «здав і забув». На порядку денному постає проблема виконання завдань з інших предметів.

По-третє, наука програмування швидко розвивається, тому отримані зазвичай на початкових курсах знання приймань програмування до моменту закінчення ВНЗ значною мірою застарівають. Виникає парадокс: першокурсник професійно більш сучасний, ніж випускник.

По-четверте, складність завдань автоматизації неухильно зростає. Молодий фахівець, одержавши уявлення про одні рівні складності програмних систем, потрапляє в умови наступного рівня складності. Тому неминучий етап адаптації до нових умов. Від випускника знову потрібні якості оперативного реагування на умови роботи, які змінилися.

По-п'яте, якісна програма повинна не тільки вирішувати прикладне завдання, а й легко читатися та розумітися. У програмістській практиці нерідка ситуація, коли вже розроблена програма передається іншому програмістові для її супроводу й розвитку, робота цього програміста починається з вивчення програми. Використання гарних правил іменування ідентифікаторів та аналогічних методичних приймань суттєво підвищує спадковість. Тому доступність розуміння програми є насамперед практичним принципом: як показують

наукові дослідження, під час налагодження програм до 90% часу витрачається на читання вхідного тексту. У програміста, який «пише» програму, створюється ілюзія її ясності. З метою економії часу студенти використовують закладені в систему програмування правила іменування ідентифікаторів, по закінченні двох тижнів доводиться відновлювати в пам'яті те, що написано раніше. У результаті на запитання «Яке призначення кнопки Button8?» студент відповідає тільки після уважного перегляду й вивчення своєї програми. Більшу допомогу в читанні програми виявляє застосування правил іменування ідентифікаторів, однак часто студенти зневажають цими правилами.

По-шосте, особливість організації навчального процесу в ВНЗ полягає в тому, що студент повинен вивчити велику кількість загальноосвітніх, загальнотехнічних і спеціальних дисциплін. Диференціація наук досягла великого розмаху й не пройшла повз програмування, яке опирається на теорію алгоритмів, теорію абстрактних типів даних, теорію проектування програм тощо. Як правило, освоєння різних дисциплін здійснюється під керівництвом різних викладачів, у результаті чого в студентів виникають утруднення в усвідомленні цілісності розподіленого по різних дисциплінах навчального матеріалу.

По-сьоме, перед викладачем постає проблема оцінювання знань і навичок студентів у галузі програмування, яка значною мірою зачіпає інтереси студента. З огляду на те що обговорювана технологія орієнтована на створення складних програм, викладачеві доводиться долати серйозні труднощі під час вибору адекватних засобів контролю або спрощувати картину. Контроль знань на рівні синтаксису конструкцій мови програмування важко назвати задовільним для програмування.

По-восьме, творці програмних засобів, як правило, вважають реалізацію програм комерційною таємницею, докладно розкривається тільки зовнішня логіка програми. Тому гарні приклади реалізації об'єктно-орієнтованих програм із коментарями на вибір розв'язків фактично відсутні. Із цієї причини, пропонуючи приклади програм студентам, викладач діє суб'єктивно, а студенти залежні від професійних пристрастей конкретного викладача.

Отже, в умовах навчального процесу студент не завжди має можливість повною мірою відчувати потребу у використанні рекомендацій об'єктно-орієнтованого програмування й оцінити їх гідно. Випускник вишу, що одержав освіту в галузі інформатики й обчислювальної техніки на прикладі простих програм, під керівництвом «структурно-орієнтованих» викладачів і на прикладі успішних «структурно-орієнтованих» програм, змушений під час одержання освіти й на початковому етапі

професійної діяльності самостійно відфільтрувати ідеї структурного проектування програм, несумісні з ідеями об'єктно-орієнтованої технології. Не можна, звичайно, стверджувати, що ці ідеї були поганими, але вони вже зіграли свою роль і на зміну їм прийшли нові. Крім того, сучасні системи програмування вже не містять засобу підтримки структурного проектування програм.

Початкове навчання програмування в школі й на першому курсі вишу здійснюється без використання об'єктно-орієнтованих засобів, неминуче базується на структурнім програмуванні, на функціональній декомпозиції розв'язуваних завдань. У школяра й студента виробляється стереотип процедурного мислення. Перехід до об'єктно-орієнтованого програмування містить у собі багато несподіванок. Використання об'єктної декомпозиції розв'язуваних завдань припускає повну перебудову системного мислення. Тому роз'яснення метафори об'єктного програмування має займати істотну частину лекційного курсу. У цьому разі неминуча й часом хвороблива перебудова професійного мислення не буде перешкодою навчальному процесу.

Висновки. Стрімкий розвиток сфери інформаційно-комунікаційних технологій вимагає принципово нових підходів до формування структури й змісту системи професійної підготовки майбутніх інженерів-педагогів комп'ютерного профілю, оскільки вимоги до змісту підготовки фахівців істотно змінюються в процесі самої підготовки. Традиційні методи розроблення структури та змісту підготовки фахівців, характерні для усталених галузей науки, техніки й економіки, малоефек-

тивні для високодинамічної сфери інформаційно-комунікаційних технологій.

Проблема запізнення знань може бути вирішена шляхом кількісних і якісних змін змісту дисциплін. У цих умовах основним завданням викладача є пошук незмінних елементів дисципліни «Мови й технології програмування» – інваріантів, які дадуть змогу перейти від кількісних змін у методиці навчання програмуванню до якісних, що допоможе вдосконалити методику навчання мов і технологій навчання.

БІБЛІОГРАФІЧНИЙ СПИСОК:

1. Бех І.Д. Виховання особистості: у 2 кн. Київ: Либідь, 2003. Кн. 1: Особистісно-орієнтований підхід: теоретико-технологічні засади: навч.-метод. вид. 280 с.
2. Биков В.Ю., Задорожна Н.Т. Сучасні підходи та принципи побудови порталів. *Засоби і технології єдиного інформаційного освітнього простору*: 36 наук. праць. Київ, 2004. С. 17–44.
3. Ильязова М.Д. Инварианты профессиональной компетентности: сущность и структура. *Педагогика и психология образования*. 2011. № 1. С. 46–53.
4. Лазарєв М.І. Полісистемне моделювання змісту технологій навчання загальноінженерних дисциплін: монографія. Харків: Вид-во НФаУ, 2003. 356 с.
5. Левшин М.М. Інваріантність як принцип проектування педагогічних систем. *Вища освіта України. Тематичний випуск «Педагогіка вищої школи: методологія, теорія, технології»*. 2014. № 3. Дод. 2. Т. 1. С. 97–104.
6. Соколова А.Н. Технологические аспекты преподавания дисциплины «Параллельное программирование». Москва, 2017. С. 303.