

В.С. Круглик

ORCID id 0000-0002-5196-7241;

В.В. Осадчий

ORCID id 0000-0001-5659-4774

МІЖДИСЦИПЛІНАРНИЙ ПІДХІД У ПРОФЕСІЙНІЙ ПІДГОТОВЦІ МАЙБУТНІХ ПРОГРАМІСТІВ

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 у 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.

Ключові слова: міждисциплінарний підхід, професійна підготовка майбутніх програмістів, програмування.

© Круглик В.С., Осадчий В.В., 2017

Вступ. Сучасна галузь програмування потребує висококваліфікованих спеціалістів, які здатні розробити програмне забезпечення для будь-якої сфери діяльності, галузі виробництва чи особистих потреб людини. Спектр запитів суспільства та економіки на якісні та зручні програми для задоволення професійних і життєвих потреб достатньо широкий. Тому нині програмування є досить розвиненою і складною сферою діяльності, що вимагає пошуку нових підходів до професійної підготовки майбутніх програмістів.

Нині розвиток сучасної науки зміщується в бік полідисциплінарного, трансдисциплінарного і міждисциплінарного підходів. Як зазначає О.М. Князева, полідисциплінарність є характеристикою такого дослідження, коли який-небудь феномен або об'єкт вивчається одночасно і з різних сторін кількома науковими дисциплінами, а трансдисциплінарність характеризує такі дослідження, які йдуть крізь кордони багатьох дисциплін і виходять за межі конкретних дисциплін, тим самим утворюючи холістичне бачення предмета дослідження (Князева О.М., 2011, с. 193–194). Міждисциплінарний підхід спроможний утворювати зв'язки-«мости» між конкретними і часто досить різними галузями знання (Огнев'юк В.О., Сисоєва С.О., 2014, с. 6–7). Полідисциплінарність відрізняється від міждисциплінарності характером відносин, які встановлюються між різними дисциплінами. У середині полідисциплінарного комплексу знань кооперація може бути взаємною і кумулятивною, але вона не є інтерактивною.

Міждисциплінарність же поєднує різні теоретичні припущення, методології та практики, які приходять від дисциплін, що залучаються. Тобто міждисциплінарність означає насамперед кооперацію різних наукових областей, циркуляцію загальних понять для розуміння якогось явища.

Проведений у рамках дослідження аналіз наукової літератури показав високий науковий інтерес дослідників до підвищення якості професійної підготовки майбутніх програмістів (Л.В. Гришко, Л.А. Матвійчук, В.В. Осадчий, З.С. Сейдаметова, Д.Є. Щедролосьєв), проте міждисциплінарний підхід висвітлювався фрагментарно. Зокрема, у роботі С.І. Тищенко окреслено підходи до взаємопов'язаного вивчення математики та спеціальних дисциплін шляхом інтегрування змісту математичних і спеціальних дисциплін у професійній підготовці молодших спеціалістів з програмування (Тищенко С.І., 2009).

Мета статті — висвітлення можливості міждисциплінарного підходу в процесі професійної підготовки майбутніх програмістів.

Міждисциплінарний контекст професії програміста. Ще у минулому столітті А.П. Єршов зазначав, що програміст повинен володіти здібністю першокласного математика до абстракції і логічного мислення у поєднанні з едісонівським талантом споруджувати все, що завгодно, з нуля і одиниці, поєднувати акуратність бухгалтера з проникливістю розвідника, фантазію автора детективних романів з тверезою практичністю

економіста, мати смак до колективної роботи, розуміти інтереси користувачів і багато що інше (Єршов А.П., 1972, с. 95). Отже, професія вимагає широкого кола міждисциплінарних знань та метакомпетенцій.

Слушним, на нашу думку, є зауваження А. Грецова та Т. Бедаревої про те, що програміст є спеціалістом, який працює не стільки з комп'ютером як з технічним пристроєм, скільки з інформацією. Він розробляє алгоритми, що дають змогу максимально ефективно вирішити певну задачу, і записує їх за допомогою наборів спеціальних символів, «зрозумілих» комп'ютеру (Грецов А., Бедарева Т., 2008, с. 139).

З огляду на це ми вважаємо, що професія програміста відноситься до трьох типів з п'яти, виділених Є.А. Клімовим на основі об'єкта праці: «людина — техніка», «людина — знакова система» і «людина — людина» (Клімов Є.А., 1983). Тому за специфікою майбутньої професійної діяльності студенти, по-перше, мають бути ознайомлені із засобами проектування у сфері матеріального світу, зокрема техносфери й інформаційної сфери, що відбувається під час вивчення дисциплін, присвячених інженерії програмного забезпечення, технологіям створення та методам розробки. Крім того, у сфері пізнавальних процесів до професіоналів сфери «людина — техніка» висуваються високі вимоги щодо різних видів пам'яті, мислення. Особливо варто виділити технічний тип мислення, що є своєрідним синтезом логічного та образного, оскільки в діяльності інженера поєднуються ці полярні стилі мислення. У професії програміста цьому типу діяльності більше всього відповідає посада адміністратора систем або мереж, який займається усуненням неполадок на різних серверах та на комп'ютерах, конфігуруванням і налаштуванням програмних систем, мережевого обладнання: оптимізує роботу мережевих пристроїв, вирішує проблеми, пов'язані зі спільним використанням обладнання організацією та інтернет-провайдером тощо (Осадчий В.В., Осадча К.П., Сердюк І.М., 2011, с. 21–22).

По-друге, відповідно до типу «людина — знакова система» майбутні програмісти мають навчитися створювати і переробляти тексти, документи, таблиці, формули, переліки, каталоги будь-яких об'єктів, вміти розбиратися в умовних позначеннях та штучних мовних системах. Так, основним завданням програміста є створення програми з використанням будь-якої мови програмування і безлічі умовних позначень: для назви змінних, функцій, процедур. Для того щоб майбутні програмісти оволоділи цими вміннями, потрібне вивчення не лише різних парадигм програмування (структурне, функціональне, модульне, об'єктно орієнтоване, логічне та ін.), а й комп'ютерних технологій обробки інформації

та розвитку мовлення (зокрема, іншомовного), використання штучних мов програмування для складання програм і програмних сценаріїв. До цього типу належить більшість назв професії програміста, а саме: інженер-програміст, системний програміст, веб-розробник, мережевий адміністратор, системний аналітик та ін.

По-третє, професії типу «людина — людина» вимагають вміння керувати групами, колективами, спільнотами людей, слухати і чути, чітко і ясно викладати свої думки, здатності розуміти внутрішній світ людини, співпереживати сторонній людині. Отже, професіонал повинен мати життєвий досвід, широкий кругозір та знання з гуманітарних дисциплін (філософії, психології, менеджменту). Цьому типу найбільше відповідають дві посади програмістів: менеджер проектів і лідер групи розробників. Основою їхньої діяльності є керівництво колективом програмістів і спілкування із замовниками програмного забезпечення.

Аналіз вимог до професійних якостей програмістів (Л.В. Гришко, Е. Дейкстра, О.Я. Кучерук, С. Макконнелл, М.Л. Смутьсон, Д.Є. Щедрольсьєв, Б. Шнейдерман) свідчить про широкий спектр компетентностей, які неможливо сформувати за умов ізоляції предметних галузей. Отже, на нашу думку, доцільним буде застосування міждисциплінарного підходу у професійній підготовці майбутніх програмістів.

Застосування міждисциплінарного підходу у процесі професійної підготовки майбутніх програмістів розглядається нами як частина загальної проблеми формування їхньої професійної компетентності. Тож дослідимо, як організувати кооперацію різних наук для розуміння явищ у процесі професійної підготовки майбутніх програмістів.

Від самого початку свого розвитку галузь програмування мала тісні зв'язки із математикою, що на сьогодні стає менш очевидним, проте залишається важливим в майбутній професійній діяльності програміста. Організація взаємодії програмування і математики у процесі навчання майбутніх програмістів необхідна для вирішення проблеми підготовки конкурентоздатних на сучасному ринку праці та пристосованих до сучасних вимог ІТ-бізнесу спеціалістів.

Сьогодні стає актуальним питання ефективної обробки великої кількості даних, аналіз і наочне представлення яких використовують у сучасній економіці для збирання цінної інформації з метою задоволення потреб певної компанії в аналітиці у межах своєї галузі. Аналітика великих даних (Big Data) і програмні рішення відіграють важливу роль у цьому процесі, а інженери-програмісти, інженери інтелектуального аналізу даних, бізнес-аналітики, архітектори великих даних (Big Data Architect) стають ключем до вико-

ристання цих технологій для реалізації найбільш ефективних проектів з аналізу великих обсягів даних. Для виконання популярної на сьогодні функції аналізу великих даних (Big Data) програміст має володіти знаннями з математики (основи лінійної алгебри та матричного обчислення, теорія похідної, основні методи математичної статистики, методи оптимізації, методи аналізу і рішення систем лінійних рівнянь, основи математичного аналізу, теорія ймовірностей) та вміти використовувати математичні методи при вирішенні прикладних задач.

Крім того, майбутній програміст має бути ознайомлений із поняттями у галузі інженерії знань, керування знанням, людино-машинної взаємодії. І якщо вони не будуть внесені у програму навчання як окремі дисципліни, то їх елементи мають бути присутніми у процесі вивчення професійних дисциплін з програмування («Штучний інтелект», «Алгоритми та структури даних», «Теорія програмування» тощо).

Професійна діяльність IT-фахівців нині вимагає фундаментальної математичної підготовки, оскільки, як зазначає О.Я. Кучерук, для створення програмного продукту необхідно мати достовірну адекватну модель тієї предметної області, яка досліджується або автоматизується. У зв'язку з цим виникають нові вимоги до підготовки інженерів-програмістів, рівень професійної підготовки яких залежить і від того, наскільки вони знайомі з математичними моделями, які використовуються при дослідженні різноманітних об'єктів, та методами побудови цих моделей, володіють ефективними алгоритмами та процесом їх створення для розв'язання поставленого завдання тощо (Кучерук О.Я., 2014, с. 134).

Зважаючи на це доцільним у викладанні дисциплін з програмування у процесі професійної підготовки майбутніх програмістів буде побудова заняття на основі двох частин: математичної та програмування. Таким чином, при отриманні завдання на розробку програмного продукту студенти спочатку мають задіяти знання з математики у частині математичного аналізу, лінійної алгебри, чисельних методів, теорії множин для моделювання рішення задачі, а потім представити програмне рішення на комп'ютері за допомогою програмування на штучній мові. У ході такого заняття відбувається формування професійної компетентності майбутніх інженерів-програмістів у рамках двох типів професії «людина — техніка» та «людина — знакова система».

Майбутній програміст має вміти застосовувати знання з економіки, адже будь-яка програма може бути затребувана на споживчому ринку, тому студенти повинні бути ознайомлені з економікою програмної інженерії. Це передбачає ознайомлення з принципами і методами економіки розробки технічних систем, включа-

ючи питання планування, формалізації та стандартизації виробничих процесів, економічного контролю і супроводу процесу розробки програмного продукту, а також з основними сучасними методами прогнозування трудомісткості програм з урахуванням їх характеристик і обмежених ресурсів. В основному студент повинен застосовувати вміння аналізувати ефективність процесу розробки програмного забезпечення під час написання програм на практичних заняттях з програмування, роботи над індивідуальними завданнями чи написання кваліфікаційної роботи, щоб у майбутньому бути здатним використовувати ці знання і вміння у професійній діяльності.

Одним з ефективних інструментів формування професійної компетентності є професійна дискусія. Як зазначають В.О. Ліщина та Н.М. Ліщина, «абсолютно природним чином в дискусіях, присвячених розробці програмного забезпечення, студенти використовують знання, отримані при вивченні дисциплін непрограмістів» (Ліщина В.О., Ліщина Н.М., 2013). Науковці пропонують застосовувати метод дискусій під час вивчення програмування від першого до старших курсів, завдяки йому студенти навчаться стисло і ясно формулювати свою точку зору, зіставляти власний підхід з підходом однокурсників, слухати і чути. Це, у свою чергу, розвиває майбутніх програмістів у рамках професії типу «людина — людина», закладаючи основи професійної компетентності. На молодших курсах, коли головне завдання студентів-програмістів полягає в оволодінні мовними засобами і здобутті стійких навичок їх застосування, темою дискусії може бути доцільність застосування тієї або іншої синтаксичної конструкції для вирішення конкретного нескладного завдання. Під час вивчення об'єктно орієнтованого програмування студентам пропонується ціла низка завдань для визначення ключових абстракцій і механізмів їх взаємодії, завдяки чому вони вчаться «придумувати змінні», визначати поля і методи об'єктів, пропонувати свої підходи до вирішення задач, зіставляти отримані рішення. На матеріалі об'єктно орієнтованого програмування дискусії стають змістовнішими. На старших курсах завдання студентів ускладнюються: майбутні програмісти мають виконувати завдання пошукового характеру, моделювати реальні процеси, обговорюючи з викладачем адекватність моделі, якість зробленого програмного забезпечення, питання стилю програмування.

Специфіка майбутньої професійної діяльності, яка полягає у частому пошуку ефективних рішень у процесі розробки програмного продукту чи проектуванні систем, вимагає тісного зв'язку професійної підготовки майбутніх інженерів-програмістів з дослідницькою діяльністю. Так,

в рамках курсу «Основи наукових досліджень» відбувається оволодіння студентами методологією і методами наукового дослідження. Одним із головних вмінь, які отримують майбутні програмісти під час вивчення цієї дисципліни, є формування експериментальних гіпотез та організація їх перевірки. Формування наукового світогляду, прищеплення студентам навичок науково-дослідницької діяльності, залучення їх до розв'язання наукових проблем забезпечується як у процесі вивчення дисциплін фундаментальної, природничо-наукової, так і професійної та практичної підготовки. Вагоме значення у формуванні конкурентоздатності та «вартості» майбутнього інженера-програміста на ринку праці належить його співпраці з науковими лабораторіями та відділами університету, оскільки саме цей вид діяльності забезпечує можливість одночасного здобуття професійного стажу, участь у проектуванні та розробці проектів, можливість відкриття власних напрямів роботи та дослідження, безпосередню взаємодію та співпрацю з провідними фахівцями та науковцями (Осипова Н.В., Вінник М.О., Тарасіч Ю.Г., 2014, с. 153–155).

Зважаючи на те що професійна комунікація інженерів-програмістів має низку особливостей, доцільно під час навчання програмуванню застосовувати міждисциплінарні зв'язки з філологічними дисциплінами. Для цього під час написання документації з програмування (технічного завдання на розробку програмного засобу, керівництв користувача, довідки по розроблюваній програмі) варто звернути увагу на вдосконалення уміння висловлювати свої думки та оформляти документацію відповідно до чинних вимог.

У професійній підготовці програмістів, конкурентоспроможних на ринку праці, нині значно

зростає роль іноземної мови, грамотне використання якої стає невід'ємною частиною їхньої професійної компетентності (іншомовної). Адже всі мови програмування ґрунтуються на лінгвограматичних конструкціях англійської мови, від рівня володіння якими залежить швидкість запам'ятовування і тлумачення конструкцій мов програмування. Тому буде цілком доречним застосування міждисциплінарних зв'язків з вивчення іноземної мови та професійного циклу програмістів.

Оскільки сфера майбутньої професійної діяльності програмістів лежить у площині пошуку, збирання, зберігання, переробки, передачі інформації, застосування сучасних засобів і методів її захисту в інформаційних мережах, то можна сказати, що всі види такої діяльності мають комунікативне забарвлення через особливий канал комунікації — персональний комп'ютер. Тому доцільно під час викладання професійних дисциплін надати студентам уявлення про етикет мережевої комунікації, через що реалізуватиметься міждисциплінарний зв'язок не лише з філологічними дисциплінами, а й з етикою.

Отже, у процесі навчання майбутніх програмістів здійснюється кооперація різних наукових дисциплін (математичних, економічних, філологічних, соціально-гуманітарних та ін.), циркуляція загальних понять для розуміння явищ у галузі програмування (математичне моделювання, технічна документація, економіка програмної інженерії, наукові методи пізнання, мовні конструкції, етика спілкування, інформаційна культура, інформаційне суспільство, авторське право на програмне забезпечення (ПЗ) тощо), що становить основу міждисциплінарного підходу у професійній підготовці майбутніх програмістів (рис. 1).



Рис. 1. Приклад міждисциплінарних зв'язків у професійній підготовці майбутніх програмістів

Висновки та перспективи подальших досліджень. Професійна підготовка майбутніх програмістів передбачає вивчення різних дисциплін, що пов'язані між собою послідовністю вивчення, забезпечують засвоєння інших предметів і є відправною точкою для подальшої професійної діяльності. Вивчення дисциплін гуманітарної та соціально-економічної, математичної та природничо-наукової (фундаментальної) сфер є основою формування професійної компетентності, що здійснюється на основі міждисциплінарного підходу під час вирішення професійно спрямованих завдань на

заняттях з дисциплін професійної та практичної підготовки. Кожна дисципліна представляє певну науку, надаючи частину знань у вигляді об'єктів, явищ, принципів, підходів, що застосовуються у процесі розробки програмних продуктів програмістами (програмуванні). Тому подальші дослідження доцільно буде спрямувати на розробку методики застосування міждисциплінарних зв'язків з науками гуманітарної, соціально-економічної, математичної та природничо-наукової сфер у процесі професійної та практичної підготовки майбутніх програмістів у ВНЗ.

ДЖЕРЕЛА

1. *Osipova N.* The model of formation of research competence of future software engineers / N. Osipova, M. Vinnik, Y. Tarasich // Informational Technologies in Education. — 2014. — № 20. — P. 150–159. — DOI: 10.14308/ite000505.
2. *Грецов А.* 100 популярных профессий. Психология успешной карьеры для старшеклассников и студентов / А. Грецов, Т. Бедарева. — СПб. : Питер, 2008. — 272 с.
3. *Ершов А.П.* О человеческом и эстетическом факторах в программировании // Кибернетика. — 1972. — № 5. — С. 95–99.
4. *Климов Е.А.* Психолого-педагогические проблемы профессиональной консультации / Е.А. Климов. — М. : Знание, 1983. — 95 с.
5. *Князева Е.Н.* Трансдисциплинарные стратегии исследований / Е.Н. Князева // Вестник ТГПУ. — 2011. — № 10. — С. 193–201.
6. *Кучерук О.Я.* Компетентнісний підхід у підготовці майбутніх інженерів-програмістів [Електронний ресурс] / О.Я. Кучерук // Науковий огляд. — 2014. — Т. 3, № 2. — Режим доступу: <http://naukajournal.org/index.php/naukajournal/article/view/170/259>
7. *Ліщина В.О.* Особливості підготовки студентів-програмістів у вищих навчальних закладах та розвиток їх професійної культури / В.О. Ліщина // Наукові записки РДГУ. — 2013. — Вип. 7 (50). — С. 90–92.
8. *Осадчий В.В.* Вступ до спеціальності програміста : навч. посіб. / В.В. Осадчий, К.П. Осадча, І.М. Сердюк. — Мелітополь : РВЦ МДПУ, 2011. — 296 с.
9. Освітологія: фахова підготовка : навч.-метод. посіб. / за ред. В.О. Огнев'юка ; авт. кол.: В.О. Огнев'юк, С.О. Сисоєва, О.О. Драч, К.О. Линьов та ін. — К. : ВП «Едельвейс», 2014. — 612 с.
10. *Тищенко С.І.* Інтегрування змісту математичних і спеціальних дисциплін у професійній підготовці молодших спеціалістів з програмування : автореф. дис. ... канд. пед. наук : 13.00.04 / С.І. Тищенко. — К., 2009. — 20 с.

Круглик В.С., Осадчий В.В.

МЕЖДИСЦИПЛИНАРНЫЙ ПОДХОД В ПРОФЕССИОНАЛЬНОЙ ПОДГОТОВКЕ БУДУЩИХ ПРОГРАММИСТОВ

Авторами статьи рассматриваются проблемы профессиональной подготовки будущих программистов в контексте внедрения междисциплинарного подхода. Дано описание специфики будущей профессиональной деятельности программистов на примере трех типов профессий по объекту труда («человек — техника», «человек — знаковая система», «человек — человек»). Применение междисциплинарного подхода в профессиональной подготовке будущих программистов рассматривается как часть процесса формирования их профессиональной компетентности через реализацию связей с дисциплинами гуманитарной, социально-экономической, математической и естественно-научной подготовки.

Ключевые слова: *междисциплинарный подход, профессиональная подготовка будущих программистов, программирование.*

V. Kruglyk, V. Osadchy

**INTERDISCIPLINARY APPROACH IN FUTURE SOFTWARE ENGINEERS
PROFESSIONAL TRAINING**

The authors examine the problem of future software engineers training in the context of the multidisciplinary approach. The description of the specific activity of future professional software engineers according to the three types of occupations as to the object of labour ("man-technique", "man-sign system", "man-man") is presented.

The authors emphasize that the interaction of software engineering and mathematics in teaching future programmers is necessary to address training competitive specialists in the modern labour market and to adapt to modern requirements of IT business professionals. Today the issue becomes relevant, the efficient processing of "big data" analysis and visual presentation are used in today's economy for collecting valuable information to meet the needs of company's analytics within their industry.

According to domestic and foreign appropriate experience in future, software engineers training sessions have to include two parts: mathematics and software engineering.

The paper shows that the future software engineer should be able to apply knowledge of economics, because any software has to be in demand in the consumer market, so the student must be acquainted with the economy of software engineering.

The use of the multidisciplinary approach in future software engineers training is considered as a part of the development of their professional competence through the implementation of communication of humanitarian, socio-economic, mathematical disciplines and natural-science training. It is determined that each of these disciplines provides knowledge in the form of objects, phenomena, principles, approaches used in the development of software engineers.

Key words: *interdisciplinary approach, future software engineers training, software engineering.*

Стаття надійшла до редакції 22.10.2016

Прийнято до друку 25.10.2016