

ПРОФЕСІЙНА ПІДГОТОВКА МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ У ПРОЦЕСІ ВИВЧЕННЯ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ ЯК ПРОБЛЕМА СУЧАСНОЇ ПЕДАГОГІЧНОЇ НАУКИ

Сергій Конюхов

Мелітопольський державний педагогічний університет імені Богдана Хмельницького

Анотація:

У статті проаналізовано сучасний стан розробленості проблеми професійної підготовки майбутніх інженерів-програмістів у процесі вивчення об'єктно-орієнтованого програмування. Акцентовано на доробку вітчизняних науковців, які досліджували особливості організації навчання майбутніх інженерів-програмістів, а також на роботах фахівців з комп'ютерних наук, у яких висвітлено фундаментальні поняття ООП. Уточнено такі поняття, як «методологія ООП», «парадигма ООП», «технології ООП», «компетентність майбутнього інженера-програміста з ООП». Окреслено проблеми у сфері професійної підготовки майбутніх інженерів-програмістів, які потребують розв'язання.

Ключові слова:

заклади вищої освіти; майбутній інженер-програміст; об'єктно-орієнтоване програмування; компетентність з об'єктно-орієнтованого програмування.

Аннотация:

Конюхов Сергей. Профессиональная подготовка будущих инженеров-программистов в процессе изучения объектно-ориентированного программирования как проблема современной педагогической науки. В статье проанализировано современное состояние разработанности проблемы профессиональной подготовки будущих инженеров-программистов в процессе изучения объектно-ориентированного программирования. Особое внимание уделено наработкам отечественных ученых, которые исследовали специфику организации обучения будущих инженеров-программистов, а также работам специалистов в области компьютерных наук, в которых раскрыты фундаментальные понятия ООП. Уточнены такие понятия, как «методология ООП», «парадигма ООП», «технологии ООП», «компетентность будущего инженера-программиста в ООП». Выявлены проблемы в сфере профессиональной подготовки будущих инженеров-программистов, требующие решения.

Ключевые слова:

учреждения высшего образования; будущий инженер-программист; об'єктно-орієнтоване програмування; компетентність в об'єктно-орієнтованом програмуванні.

Resume:

Koniukhov Serhii. Professional training of future software engineers in the process of studying object-oriented programming as a problem of modern pedagogical science

In the article the current state of the development of the problem of professional training of future software engineers in the process of studying object-oriented programming is analyzed. Particular attention is paid to the research of domestic scientists who investigate the features of the organization of training of future software engineers, as well as the works of specialists in the field of computer science, which highlight the fundamental concepts of the OOP. The concepts "OOP methodology", "OOP paradigm", "OOP technology", "competence of a future software engineer in OOP" are clarified, and problems in the field of professional training of future software engineers that need to be addressed are outlined.

Key words:

higher education institutions; future software engineer; object-oriented programming; competence in object-oriented programming.

Постановка проблеми. Світова індустрія розробки програмного забезпечення нині характеризується високим темпом розвитку. Як наслідок, змінюються вимоги суспільства й роботодавців до інженерів-програмістів. Зокрема вони повинні мати високий рівень компетентності з обраної спеціалізації, бути здатними до професійної мобільності й адаптації до швидко змінюваних умов діяльності, а також до постійного підвищення кваліфікації.

Професійна підготовка майбутніх інженерів-програмістів у закладах вищої освіти (ЗВО) має враховувати ці виклики, а також відповідати процесам інтеграції національної системи вищої освіти зі світовим освітнім простором і вітчизняної ІТ-індустрії з загальносвітовою. У зв'язку з цим вона потребує модернізації на засадах концепції студентоцентрованого навчання, компетентнісного й практикоорієнтованого підходів, посилення зв'язку з роботодавцями.

Аналіз останніх досліджень і публікацій. У дослідженнях, присвячених проблемам професійної підготовки майбутніх інженерів-

програмістів та ІТ-фахівців у ЗВО, розкрито її теоретико-методологічні засади (Л. Гришко, В. Круглик, В. Осадчий, З. Сейдаметова, С. Семеріков та ін.), особливості впровадження компетентнісного підходу (М. Вінник, Л. Зубик, В. Сєдов, А. Стрюк та ін.), окремі аспекти вивчення об'єктно-орієнтованого програмування (О. Баранюк, І. Барков, Т. Вакалюк, В. Єремєєв, С. Жуковский, Л. Калінникова, Г. Рудакова та ін.).

Результати прикладних досліджень вітчизняних (І. Бернакевич, В. Бублик, П. Вагін, П. Кравець, М. Львов, Д. Манжос, Т. Рак, О. Співаковський та ін.), а також іноземних науковців (Т. Бенайа, М. Бен-Арі, Є. Бенедсен, М. Бергес, Е. Зур, М. Касперсен, М. Колінг, Н. Ліберман, Н. Рагоніс, П. Хабвейзер та ін.) свідчать про наявність низки проблем у процесі формування у студентів здатності до використання об'єктно-орієнтованого підходу під час написання програм.

Вітчизняні науковці досліджують також проблеми навчання об'єктно-орієнтованого програмування й об'єктно-орієнтованого

підходу до розробки програмного забезпечення. Так, наприклад, у роботах Т. Гончаренко, Л. Зубик, Ф. Ільєсової, В. Круглика, С. Лещук, С. Семерікова ООП розглядається як елемент професійної підготовки майбутніх інженерів-програмістів і фахівців з інформаційних технологій. О. Теплицький та І. Теплицький досліджують особливості навчання об'єктно-орієнтованого моделювання з використанням спеціалізованих програмних середовищ студентів природничих спеціальностей. Окремі аспекти методики навчання об'єктно-орієнтованого підходу в програмуванні й споріднених дисциплінах вивчають О. Азаров, О. Черняк і Л. Савицька, О. Баранюк, М. Цибулько. Місце об'єктно-орієнтованого підходу в сучасній природничо-математичній освіті є предметом розвідок І. Степури.

Разом з тим, у наявних дослідженнях об'єктно-орієнтований підхід здебільшого розглядається в межах окремих дисциплін, тому

актуальним є вивчення можливостей зазначеної парадигми як ланки, що з'єднує окремі курси з програмування.

Формулювання цілей статті. Мета статті – проаналізувати сучасний стан розробленості проблеми професійної підготовки майбутніх інженерів-програмістів у процесі вивчення об'єктно-орієнтованого програмування.

Виклад основного матеріалу дослідження. У сучасних дослідженнях, науковій і технічній літературі об'єктно-орієнтоване програмування часто позначається як парадигма, методологія або технологія програмування. Усі зазначені поняття є загальнонауковими й використовуються для опису дослідницької діяльності в окремих галузях науки. Поняття «методологія» і «парадигма» мають філософські витоки й використовуються для осмислення процесу пізнання й наукової діяльності загалом. У табл. 1 наведено деякі визначення зазначених понять.

Таблиця 1

Визначення понять «методологія», «парадигма» і «технологія»

Поняття	Визначення
Методологія	– «система принципів і способів організації і побудови теоретичної та практичної діяльності, а також вчення про цю систему» [5, с. 19]; – «1) вчення про науковий метод пізнання; 2) сукупність методів, які застосовуються в певній галузі діяльності людини» [6, с. 217]; – «сукупність підходів, способів, методів, прийомів і процедур, що застосовуються в процесі наукового пізнання та практичної діяльності для досягнення наперед визначеної мети» [15, с. 372].
Парадигма	– «сукупність методологічних, світоглядних, наукових, управлінських та інших настанов, що сформувались історично й прийняті у своїй спільноті як зразок, норма, стандарт розв'язання проблем» [13, с. 465]; – «означає прийняту певним науковим співтовариством модель постановки й розв'язання проблем, яка забезпечує існування наукової традиції» [15, с. 296].
Технологія	– «сукупність виробничих процесів у певній галузі виробництва, а також науковий опис способів виробництва, сукупність прийомів, які використовуються в якійсь справі, майстерності, мистецтві» [5, с. 19]; – «сукупність прийомів і способів отримання, обробки й переробки сировини, матеріалів, напівфабрикатів або виробів, а також наукова дисципліна, що розробляє й удосконалює такі прийоми та способи» [13, с. 65].

Проаналізувавши визначення, наведені в табл. 1, доходимо висновку, що кожне з цих понять може бути застосовано до комп'ютерних наук і програмування як до особливого виду діяльності.

Методологія програмування – це «об'єднана загальним філософським підходом сукупність методів, які використовуються в процесі розроблення програмного забезпечення» [5, с. 19; 10, с. 32]. Більш розгорнуте визначення наведено в тлумачному словнику з інформатики: «Методологія – сукупність поглядів на розробку ПЗ і його логічну структуру, втілених у формі набору взаємодіючих інструментальних і мовних засобів, підтримуваних прийнятою в організації нормативною базою, системою навчання та

сформованою корпоративною культурою» [14, с. 419].

Поняття «парадигма» в сучасній науковій обіг увів Т. Кун у роботі «Структура наукових революцій». Науковець розглядав її як спосіб виконання дій, зразок розв'язування дослідницьких задач, а не лише як теорію [6, с. 279].

Отже, поява понять «парадигма програмування» й «парадигма об'єктно-орієнтованого програмування» є закономірною, оскільки будь-яке програмування передбачає використання певних зразків і стандартів у процесі написання програм.

Термін «парадигма програмування» вперше використав проф. Р. Флойд у своїй лекції

лауреата премії Тюрінга в 1978 р., апелюючи саме до роботи Т. Куна. Науковець зазначав, що деякі його ідеї можна застосовувати в галузі комп'ютерних наук. Зокрема в ній, як і в будь-якій іншій науці, є спільноти однодумців, що використовують власні мови й парадигми [18, с. 456]. Р. Флойд підкреслював, що розвиток комп'ютерних наук і програмування вимагає створення нових парадигм, але розвиток майстерності окремого програміста вимагає від нього оволодіння різними парадигмами [18, с. 457] (а не відмови від старих на користь нових – *прим. автора*).

У тлумачному словнику з інформатики зазначено, що термін «парадигми програмування» пов'язаний з уявленнями про основні елементи програмування й зв'язки між ними. Такими базовими елементами є дані, механізми їх оброблення (оператори, функції, процедури, методи тощо мов програмування), способи організації взаємодії даних і засобів їх обробки [14, с. 461–462].

Як зауважує І. Одинцов, коли методологія використовується на стадії програмування, її часто називають парадигмою програмування – способом мислення й програмування, який не пов'язаний з конкретною мовою [10, с. 74].

Технологію програмування прийнято розглядати як наукову дисципліну і як сукупність процесів розробки програмного забезпечення [5; 10; 14]. Зокрема, В. Камаєв і В. Костерін визначають технологію програмування так: «наукова й практично апробована стратегія створення програм, що містить опис сукупності методів і засобів розробки програм, а також порядок застосування цих методів і засобів» [5, с. 20]. Подібне трактування технології програмування містить і «Тлумачний словник з інформатики»: «Набір методичних, організаційних та інструментальних засобів, що полегшують створення програм і допомагають підвищити їх споживчі характеристики» [14, с. 563].

Ми дотримуємось поглядів авторів наведених визначень і вважаємо, що вони не суперечать загальнонауковим дефініціям (табл. 1) і можуть використовуватись для визначення методології, парадигми й технології об'єктно-орієнтованого програмування (ООП).

Об'єктно-орієнтованому програмуванню присвячено чимало наукових, навчально-методичних, технічних праць. Дослідники наводять як термінологічні, так і описові (через опис функцій і властивостей) визначення ООП, які спираються на концептуальні особливості цього підходу. У зв'язку з цим зупинимось лише на окремих дефініціях.

Зокрема, термінологічне визначення наводить Г. Буч: «Об'єктно-орієнтоване програмування –

це метод програмування, що ґрунтується на поданні програми у вигляді сукупності об'єктів, що взаємодіють між собою, кожний з яких є екземпляром певного класу, а класи є членами певної ієрархії спадкування». Дослідник підкреслює, що програму можна вважати об'єктно-орієнтованою, коли виконуються три умови, зазначені в цьому визначенні [2, с. 69].

На нашу думку, це визначення не потребує розширення або уточнення й на нього можна спиратись у процесі навчання майбутніх інженерів-програмістів.

У роботі І. Одинцова зазначено: «Методологія об'єктно-орієнтованого програмування – це підхід, який використовує об'єктну декомпозицію, коли статична структура системи відображена в термінах об'єктів і зв'язків між ними, а поведінка системи – в термінах обміну повідомленнями між об'єктами» [10, с. 83].

Основними поняттями ООП є «клас» і «об'єкт». Г. Шилдт, автор фундаментального керівництва з програмування на C++, зазначає, що клас визначає новий тип даних, який задає формат об'єкта. Він містить дані й код для виконання над ними, тобто пов'язує дані з кодом. Об'єкти – це екземпляри класу [16, с. 30–31].

До основних концепцій ООП належать: абстрагування, інкапсуляція, модульність, ієрархія (Г. Буч [2]); інкапсуляція, спадкування, поліморфізм, композиція (М. Вайсфельд [3]); моделювання дій реального світу, наявність типів даних користувача, приховування деталей реалізації, можливість багаторазового використання коду, інтерпретація викликів функцій на етапі виконання (А. Пол [11]).

Загалом нині залишається актуальною думка, яку висловив Б. Страуструп (B. Stroustrup) ще в 1988 р.: об'єктно-орієнтоване програмування – це парадигма написання «гарних» програм для розв'язання безлічі проблем. Об'єктно-орієнтовані мови повинні добре підтримувати об'єктно-орієнтований стиль програмування, тобто надавати засоби для того, щоб програмування було зручним (легким, безпечним, ефективним) [21, с. 10].

Отже, як слушно зазначив С. Семеріков, «об'єктно-орієнтоване програмування є найбільш природною методологією програмування, яка, урахувавши особливості психічних процесів, дає змогу створювати чітко структуровані та осяжні складні програмні продукти» [12, с. 58].

Спираючись на наведені вище визначення, а також урахувавши особливості ООП, вважаємо за доцільне уточнити поняття «методологія об'єктно-орієнтованого програмування»,

«парадигма об'єктно-орієнтованого програмування» й «технологія об'єктно-орієнтованого програмування».

Ми розглядаємо *методологію ООП* як сукупність підходів, способів, методів, прийомів і процедур, які використовуються в процесі написання складних програм, що являють собою сукупності дискретних об'єктів – екземплярів певних класів, які, своєю чергою, утворюють ієрархію спадкування й об'єднують у єдине ціле дані й поведінку. До таких методів, зокрема, належать: аналіз предметної області й виявлення важливих елементів, декомпозиція та композиція об'єктів, побудова діаграм класів, вибір об'єктно-орієнтованої мови програмування тощо.

Ми розглядаємо *парадигму ООП* як прийняту за норму (стандарт, зразок) певною неформальною спільнотою програмістів модель написання складних програм, що передбачає ефективно використання таких механізмів, як інкапсуляція, приховування даних, інтерфейси, спадкування, множинне спадкування, повторне використання коду, поліморфізм, переваження функцій і операцій тощо.

Ми розглядаємо *технологію ООП* як практично апробовану й задокументовану (формально або неформально) стратегію написання об'єктно-орієнтованих програм, що полегшує створення таких програм і допомагає забезпечити належну якість продукту. Найчастіше такі стратегії представлені у вигляді шаблонів (патернів) проектування, що дає змогу багаторазово використовувати ефективні рішення.

Отже, поняття «методологія ООП», «парадигма ООП» і «технологія ООП» не є синонімічними, але всі вони використовуються для опису процесу створення складних об'єктно-орієнтованих програм. З огляду на це, їх можна використовувати як взаємозамінні.

Компетентність фахівців ІТ-галузі з ООП є предметом багатьох досліджень. Наприклад, М. Havenga, Е. Mentz, і R. De. Villiers виділили чотири категорії здібностей (знань, умінь і стратегій діяльності), необхідних для успішного застосування ООП: 1) когнітивні (знання, розуміння, застосування, аналіз, синтез, оцінювання); 2) метакогнітивні (знання про себе, задачі, стратегії діяльності й керування тощо); 3) розв'язання проблем (знання про типи проблем, шляхи їх розв'язання та ін.); 4) з ООП (розробка й обслуговування програмного забезпечення, умінь проектувати класи, створювати об'єкти, використовувати методи, контролювати доступ тощо) [19, с. 3–7].

Під *компетентністю з ООП майбутніх інженерів-програмістів* ми розуміємо: здатність розуміти фундаментальні основи об'єктно-орієнтованого програмування, використовувати

знання з царини методології та технології ООП й умінь виконувати декомпозицію та композицію предметної області, яка підлягає моделюванню, визначати властивості об'єктів і взаємодію між ними, розробляти алгоритми оброблення об'єктів, реалізовувати їх засобами об'єктно-орієнтованих мов програмування для виконання професійних завдань і створення ефективних, надійних, якісних програм.

У процесі формування в майбутніх інженерів-програмістів професійної компетентності під час навчання ООП виникають різні проблеми. Зокрема, однією з них, за визначенням Д. Саянемі, М. Куїтнінен і Т. Тінкасало є наявність у студентів-початківців помилкових уявлень про такі концептуальні поняття ООП, як «клас» і «об'єкт», які є основою для виокремлення даних і методів у програмі. Дослідники також підкреслюють важливість розуміння студентами ще одного аспекту об'єктно-орієнтованих програм – їх виконання (послідовність використання методів та їхній вплив на об'єкти) [20].

Ще дві суттєві проблеми, які постають у процесі навчання ООП, називає Л. Манжос: труднощі з розумінням базових понять ООП і вибором мови програмування. Щодо останньої, то автор наголошує на тому, що під час вибору слід урахувувати такі характеристики, як кросплатформність, наявність різноманітних вільно поширюваних і комерційних середовищ розроблення, можливість використання для розв'язання широкого кола задач [8, с. 48]. На нашу думку, важливими є також інші фактори, зокрема повнота реалізації в мові програмування концепцій ООП, попередній рівень професійної підготовки та ін.

Цікаві думки на проблеми навчання ООП висловлює М. Касперсен. На його переконання, складність вивчення об'єктно-орієнтованого підходу зумовлена двома чинниками: внутрішньою складністю самої парадигми й випадковою складністю, яка пов'язана з зовнішніми факторами (використання неадекватних мов програмування, інструментів, стратегій навчання, брак практичного досвіду ООП у викладачів тощо). Дослідник зауважує, що багато проблем пов'язано з випадковою складністю, тому їх можна й потрібно долати [17, с. 77].

Спираючись на роботи цих та інших дослідників, а також на власний досвід, виділимо такі основні проблеми, що негативно впливають на ефективність формування в майбутніх інженерів-програмістів професійної компетентності й вимагають подолання в процесі навчання об'єктно-орієнтованого програмування:

1. Когнітивні проблеми, що пов'язані зі складністю об'єктно-орієнтованої парадигми.

2. Недостатній рівень сформованості інтра- та інтерперсональних здібностей (особистих, соціальних, управлінських).

3. Проблема вибору мови програмування для навчання ООП.

4. Проблеми програмного забезпечення, яке використовується в навчанні ООП.

Отже, залишається актуальним завдання щодо вдосконалення наявних і впровадження нових форм, методів і засобів формування в майбутніх інженерів-програмістів розуміння фундаментальних положень об'єктно-орієнтованої парадигми, а також здатності до їх практичного застосування. Одним зі шляхів виконання цього завдання є аналіз прогресивного досвіду навчання ООП у закладах вищої освіти.

Так, В. Круглик у межах запропонованої ним системи підготовки майбутніх інженерів-програмістів до професійної діяльності особливу увагу приділив вивченню парадигм програмування, зокрема об'єктно-орієнтованого. Науковець підкреслює, що в процесі вивчення ООП необхідно приділяти особливу увагу активізації пізнавальної діяльності студентів шляхом організації їхньої самостійної роботи й залучення до науково-дослідної роботи. Дослідник розробив струнку схему навчання майбутніх інженерів-програмістів програмування, якою передбачено вивчення об'єктно-орієнтованої парадигми на другому році навчання в межах дисципліни «Програмування» на прикладі мов програмування C++ або C#. Корисним для практики викладання ООП є розкритий ним методичний прийом, коли для демонстрації відмінностей між об'єктно-орієнтованим і структурним підходами використовується приклад гри в теніс [7, с. 289–290]. Уважаємо, що цей приклад викладач може легко змінити на будь-який інший або запропонувати студентам самостійно виконати добір подібних прикладів. Закріплювати й поглиблювати знання й навички з ООП В. Круглик пропонує під час вивчення таких професійно-орієнтованих дисциплін: «Операційні системи та системне програмування», «Веб-програмування», «Візуальне програмування». Так, у курсі «Веб-програмування» окремо розглядається реалізація ООП у мові програмування PHP [7, с. 293]; у курсі «Візуальне програмування» студенти навчаються створювати програми з графічним інтерфейсом користувача засобами середовищ візуального об'єктно-орієнтованого програмування [7, с. 294].

С. Семеріков розробив методичну систему активізації пізнавальної діяльності студентів у процесі вивчення чисельних методів, що ґрунтується на використанні об'єктного підходу,

об'єктно-орієнтованої технології програмування й спеціалізованої математичної об'єктно-орієнтованої бібліотеки, призначеної для програмування чисельних методів [12]. Науковець подав докладну характеристику технології об'єктно-орієнтованого програмування, доводячи її найбільшу придатність і зручність для застосування в курсі чисельних методів. Він підкреслив, що «об'єктно-орієнтоване програмування є найбільш природною методологією програмування, яка, урахувавши особливості психічних процесів, дає змогу створювати чітко структуровані й осяжні складні програмні продукти» [12, с. 59]. Особлива цінність розглянутого дослідження, на нашу думку, полягає в тому, що запропонована автором методика продемонструвала можливості розширення сфери застосування ООП на дисципліни математичного циклу, які часто вивчалися окремо від курсів програмування.

Л. Зубик у дисертації зосередила увагу на формуванні професійних компетентностей майбутніх бакалаврів з інформаційних технологій у процесі вивчення фахових дисциплін. Зокрема на основі аналізу виробничих функцій, які формуються в процесі опанування таких дисциплін, і послідовності їх вивчення дослідниця виявила наявність змістового зв'язку курсів «Об'єктно-орієнтоване програмування» (вивчається в 3-4 семестрах) і «Web-технології та web-дизайн» (вивчається в 6 семестрі) [4, с. 34–35]. Л. Зубик показала, що «Об'єктно-орієнтоване програмування» є одним з тих фундаментальних курсів, вивчення яких повинно передувати опануванню фахових дисциплін (зокрема «Web-технології та web-дизайн», «Організація баз даних і знань», «Технології комп'ютерного проектування») і мати з їх змістом прямий теоретичний зв'язок [4, с. 148–149].

Необхідність застосування прийомів критичного підходу в процесі навчання ООП підкреслюють О. Азаров, О. Черняк і Л. Савицька. У своїй роботі вони зазначають, що студентам обов'язково слід показати й переваги, і недоліки ООП, сформувати в них уміння визначати, які об'єктно-орієнтовані засоби слід використовувати для ефективного виконання поставлених завдань. Автори зосереджують увагу на організації вивчення поняття «поліморфізм» і механізмах його реалізації (перевантаження функцій, перевизначення функцій, віртуальні функції, інтерфейси, параметризовані типи) в мовах програмування C++ і C# з застосуванням критичного підходу [1].

С. Лещук розкриває досвід виконання студентами програмного проекту в межах

спецкурсу для майбутніх фахівців з інформатики. Дослідниця звертає увагу на те, що одним з завдань цього спецкурсу є закріплення розуміння студентами принципів об'єктно-орієнтованого програмування. Для цього, крім власне створення проекту, студентам були запропоновані дидактичні вправи, обговорення, дискусії [8, 84–85]. Запропонований С. Лещук спецкурс був зорієнтований на студентів освітнього ступеня «магістр» і освітньо-кваліфікаційного рівня «спеціаліст». На нашу думку, здійснювати формування розуміння концепцій ООП на цьому етапі навчання вже запізно, але систематизація й закріплення сформованих раніше знань і навичок є важливим завданням, що й продемонструвала у своїй роботі дослідниця.

На основі аналізу зазначених досліджень доходимо висновку, що наявні проблеми у сфері професійної підготовки майбутніх інженерів-програмістів з об'єктно-орієнтованого програмування можна подолати шляхом створення комплексу умов для формування у студентів відповідних знань, умінь і навичок у процесі вивчення фахових дисциплін.

Висновки та перспективи подальших досліджень. У процесі дослідження ми проаналізували сучасний стан розробленості проблеми професійної підготовки майбутніх інженерів-програмістів у процесі вивчення об'єктно-орієнтованого програмування. Акцентовано на доробку вітчизняних науковців, які досліджували особливості організації навчання майбутніх інженерів-програмістів, а також на роботах визнаних фахівців з комп'ютерних наук, у яких висвітлені фундаментальні поняття ООП. Це дало змогу уточнити такі важливі поняття, як «методологія ООП», «парадигма ООП», «технології ООП», «компетентність майбутнього інженера-програміста з ООП», а також окреслити ряд проблем у сфері професійної підготовки майбутніх інженерів-програмістів, які потребують розв'язання.

Подальші дослідження спрямовуватимуться на розробку організаційно-методичних умов формування професійної компетентності майбутніх інженерів-програмістів у процесі вивчення об'єктно-орієнтованого програмування в закладах вищої освіти.

Список використаних джерел

1. Азаров О. Д., Черняк О. І., Савицька Л. А. Аспекти критичного підходу до викладання поняття поліморфізму в об'єктно-орієнтованому програмуванні. *Інформаційні технології та комп'ютерна інженерія*. 2017. Т. 39. № 2. С. 31–34.
2. Буч Г., Максимчук Р. А., Энгл М. У., Янг Б. Дж., Коналлен Д., Хьюстон К. А. Объектно-ориентированный анализ и проектирование с примерами приложений. Москва: ООО «И. Д. Вильямс», 2008. 720 с.
3. Вайсфельд М. Объектно-ориентированное мышление. Санкт-Петербург: Питер, 2014. 304 с.
4. Зубик Л. В. Формування професійних компетентностей майбутніх бакалаврів з інформаційних технологій у процесі вивчення фахових дисциплін: дис. ... канд. пед. наук : 13.00.04 / Національний університет водного господарства та природокористування. Рівне, 2016. 341 с.
5. Камаев В. А., Костерин В. В. Технологии программирования: Учебник. Москва: Высш. шк., 2006. 454 с.
6. Краткий философский словарь / А. П. Алексеев, Г. Г. Васильев и др.; под ред. А. П. Алексеева. Москва: ТК Велби, Изд-во «Проспект», 2004. 496 с.
7. Крулик В. С. Система підготовки майбутніх інженерів-програмістів до професійної діяльності у вищих навчальних закладах: дис. ... д-ра пед. наук : 13.00.04 / Запорізький національний університет. Запоріжжя, 2018. 682 с.
8. Лещук С. О. Окремі методичні аспекти підготовки ІТ-фахівців. *Інформаційні технології в освіті*. 2017. № 1(30). С. 81–96.
9. Манжос Л. А. Объектные технологии и объектно-ориентированное программирование. *Инженерия программного обеспечения*. 2010. № 3. С. 47–49.
10. Одинцов И. О. Профессиональное программирование. Системный подход. Санкт-Петербург: БХВ-Петербург, 2004. 624 с.

References

1. Azarov, O.D., Cherniak, O.I., Savytska, L.A. (2017). Aspects of a critical approach to teaching the concept of polymorphism in object-oriented programming. *Informatsiini tekhnologii ta kompiuterna inzheneriia*. Vol. 39, No. 2. 31–34. [in Ukrainian]
2. Buch, G., Maksimchuk, R.A., Engl, M.U., Yang, B.Dzh. et al (2008). *Object-oriented analysis and design with examples of applications*. Moscow: ООО «I.D. Vilyams». [in Russian]
3. Vaisfeld, M. (2014). *Object-Oriented Thinking*. SPb.: Piter. [in Russian]
4. Zubyk, L.V. (2016). *Formation of professional competences of future bachelors on information technologies in the process of professional disciplines: thesis*. Rivne. [in Ukrainian]
5. Kamaev, V.A., Kosterin, V.V. (2006). *Programming Technologies: Textbook*. Moscow: Vyssh. shk. [in Russian]
6. Alekseev, A.P., Vasilev, G.G. (et al.) (2004). *Brief philosophical dictionary*. Moscow: TK Velbi, Izd-vo Prospekt. [in Russian]
7. Krukhlyk, V.S. (2018). *System of training future programmers-engineers for professional activity in higher educational institutions: Doctoral dissertation*. Zaporizhia. [in Ukrainian]
8. Leshchuk, S.O. (2017). Some methodological aspects of IT specialists training. *Informatsiini tekhnologii v osviti*, 1 (30), 81–96. [in Ukrainian]
9. Manzhos, L.A. (2010). Object Technologies and Object-Oriented Programming. *Inzheneriya prohramnoho zabezpechennya*, 3, 47–49. [in Ukrainian]
10. Odintsov, I.O. (2004). *Professional programming. Systems approach*. Spb.: BHV-Peterburg. [in Russian]
11. Pol, A. (2001). *Object-oriented programming in C ++*. Moscow: Binom. [in Russian]
12. Semerikov, S.O. (2000). *Activation of cognitive activity of students in the study of numerical methods in object-oriented programming technology: thesis*. Kryvyi Rih. [in Ukrainian]

11. Пол А. Объектно-ориентированное программирование на С++. Москва: Бином, 2001. 464 с.
12. Семеріков С. О. Активізація пізнавальної діяльності студентів при вивченні чисельних методів у об'єктно-орієнтованій технології програмування: дис. ...канд. пед. наук : 13.00.02 / Криворізький держ. педагогічний ун-т. Кривий Ріг, 2000. 255 с.
13. Сурмін Ю. П. Майстерня вченого: Підручник для науковця. Київ: Навчально-методичний центр «Консорціум з удосконалення менеджмент-освіти в Україні», 2006. 302 с.
14. Глумачний словник з інформатики / Півняк Г. Г., Бусигін Б. С., Дівізніок М. М. та ін. Дніпропетровськ: Нац. гірнич. ун-т, 2010. 600 с.
15. Філософський енциклопедичний словник / НАН України, Ін-т філософії ім. Г. С. Сковороди; голов. ред. В. І. Шинкарук. Київ: Абрис, 2002. 742 с.
16. Шилдт Г. С++: руководство для начинающих. Москва: Издательский дом «Вильямс», 2005. 672 с.
17. Bennedsen J., Caspersen M. E., Kölling M. (Eds.). Reflections on the Teaching of Programming: Methods and Implementations. LNCS 4821. Berlin Heidelberg: Springer-Verlag, 2008. 258 p.
18. Floyd R. W. The paradigms of programming. *Communications of the ACM*. 1979. Vol. 22. Issue 8. P. 455–460.
19. Havenga M., Mentz E., Villiers R. De. Knowledge, skills and strategies for successful object programming: a proposed learning repertoire. *South African Computer Journal*. 2008. Vol. 12. № 1. P. 1–8. URL: <https://journals.co.za/content/comp/12/1/EJC28081> (дата звернення: 23. 05. 2018).
20. Sajaniemi J., Kuittinen M., Tikansalo T. A Study of the Development of Students' Visualizations of Program State during an Elementary Object-Oriented Programming Course. *Journal on Educational Resources in Computing (JERIC)*. 2008. Vol. 7. Issue 4. Article № 3.
21. Stroustrup B. What Is Object-Oriented Programming? *IEEE Software*. 1988. Vol. 5. Issue 3. P. 10–20.
13. Surmin, Yu.P. (2006). *Scientist's workshop: A textbook for a scientist*. Kyiv: Navchalno-metodychnyi tsentr «Konsortsium z udoskonalennia menedzhment-osvity v Ukraini». [in Ukrainian]
14. Pivniak, H.H., Busyhin, B.S., Diviziniuk, M.M. (et al.). *Dictionary of Informatics*. Dnipropetrovsk: Nats. hirnych. un-t. [in Ukrainian]
15. Shynkaruk, V.I. (ed.) (2002). *Philosophical Encyclopedic Dictionary*. Kyiv: Abrys.
16. Shildt, G. (2005). *C ++: A Beginner's Guide*. Moscow: Izdatelskiy dom "Vilyams". [in Russian]
17. Bennedsen, J., Caspersen, M.E., Kölling, M. (Eds.) (2008). *Reflections on the Teaching of Programming: Methods and Implementations*. LNCS 4821. Berlin Heidelberg: Springer-Verlag. [in English]
18. Floyd, R.W. (1979). The paradigms of programming. *Communications of the ACM*. Vol. 22. Issue 8. 455–460. [in English]
19. Havenga, M., Mentz, E., Villiers, R. De. (2008). Knowledge, skills and strategies for successful object programming: a proposed learning repertoire. *South African Computer Journal*. Vol. 12. No 1. 1–8. Retrieved from: <https://journals.co.za/content/comp/12/1/EJC28081>. [in English]
20. Sajaniemi, J., Kuittinen, M., Tikansalo, T. (2008). A Study of the Development of Students' Visualizations of Program State during an Elementary Object-Oriented Programming Course. *Journal on Educational Resources in Computing (JERIC)*. Vol. 7, Issue 4, Article No. 3. [in English]
21. Stroustrup, B. (1988). What Is Object-Oriented Programming? *IEEE Software*. Vol. 5, Issue 3. 10–20. [in English]

Рецензент: д.пед.н., професор Лук'янова Л.Б.

**Відомості про автора:
Конюхов Сергій Леонідович**

konukhov@mdpu.org.ua

Мелітопольський державний педагогічний університет
імені Богдана Хмельницького
вул. Гетьманська, 20, м. Мелітополь,
Запорізька обл., 72300, Україна

doi: 10.7905/nvmdpu.v0i20.2465

Матеріал надійшов до редакції 29. 05. 2018 р.
Прийнято до друку 13. 06. 2018 р.

**Information about the author:
Koniukhov Serhii Leonidovych**

konukhov@mdpu.org.ua

Bohdan Khmelnytsky Melitopol
State Pedagogical University
20 Hetmanska Str., Melitopol,
Zaporizhia region, 72300, Ukraine

doi: 10.7905/nvmdpu.v0i20.2465

Received at the editorial office 29. 05. 2018.
Accepted for publishing 13. 06. 2018.