

УДК 378.147

Методи і засоби навчання об'єктно-орієнтованого програмування у вищих навчальних закладах

Конюхов Сергій Леонідович

*Мелітопольський державний педагогічний університет
імені Богдана Хмельницького,
м. Мелітополь, Україна*

*старший викладач кафедри інформатики і кібернетики
<http://orcid.org/0000-0002-1925-3425>, konukhov@mdpu.org.ua*

Анотація. У статті проаналізовані методи і засоби навчання, які використовуються у практиці навчання майбутніх інженерів-програмістів, визначені способи їх використання у процесі формування у студентів компетентностей у сфері об'єктно-орієнтованої розробки. Наприклад, застосування електронних засобів навчального призначення з урахуванням індивідуальних навчальних потреб студентів (середовища, які реалізують метод візуалізації понять об'єктно-орієнтованого програмування, електронні тренажери й ін.). Ще один підхід полягає у послідовному підвищенні ролі інтерактивних методів навчання, які спонукають студентів до самостійного формулювання висновків, застосування вже наявних знань і пошуку нових. Визначені напрями подальших досліджень, які полягають у розробці системи методів і засобів навчання, адаптованих до особливостей об'єктно-орієнтованого програмування як предмета вивчення у закладах вищої освіти.

Ключові слова: заклади вищої освіти; майбутній інженер-програміст; професійні компетентності; об'єктно-орієнтована парадигма; методи навчання; засоби навчання.

Methods and Means of Training Object-Oriented Programming in Higher Education Institutions

Koniukhov Serhii Leonidovych

*² Bogdan Khmelnytsky Melitopol State Pedagogical University,
Melitopol, Ukraine*

*Senior Lecturer of the Department of Computer Science and Cybernetics
<http://orcid.org/0000-0002-1925-3425>, konukhov@mdpu.org.ua*

Abstract. The article analyzes the methods and means of teaching that are used in the practice of training future engineer-programmers, the methods of their use in the process of formation of students competence in the field of object-oriented development. For example, the use of electronic teaching aids, taking into account the individual educational needs of students. These can be environments that implement the method of visualizing the concepts of OOP, simulators, with which you can organize an independent training with automatic checking of results, etc. Another approach is to consistently enhance the role of interactive learning methods that encourage students to independently formulate conclusions, apply existing knowledge and find new ones. The directions of further research, which consist in the development of a system of methods and teaching methods, adapted

to the features of object-oriented programming as a subject of study in institutions of higher education, are determined.

Keywords: higher education institutions; future software-engineer; professional competence; object-oriented paradigm; teaching methods; learning tools.

ВСТУП

Постановка проблеми. Серед багатьох факторів, які нині обумовлюють професійну успішність програміста, ми виділяємо сформованість професійних компетентностей у сфері об'єктно-орієнтованої розробки програмного забезпечення. Власний досвід практичного програмування дозволяє стверджувати, що знання, розуміння і навички застосування об'єктно-орієнтованого підходу утворюють міцне підґрунтя для написання якісного коду, оволодіння новими технологіями розробки, а також допомагають визначати суттєві недоліки цих технологій. Разом із тим, досвід викладання відповідних курсів у закладах вищої освіти свідчить, що вивчення фундаментальних теоретичних основ об'єктно-орієнтованої парадигми (ООП), її методів й інструментарію є достатньо складним для студентів. У зв'язку з цим постає низка завдань щодо застосування методів і засобів навчання, які допомогатимуть студентам у процесі опанування ООП.

Важливе місце у системі цих методів і засобів належить інформаційно-комунікаційним технологіям (ІКТ): у процесі навчання майбутніх інженерів-програмістів доцільно використовувати електронні засоби навчального призначення (ЕЗНП), різноманітні онлайн-ресурси (платформи для розробки застосувань, онлайн-компілятори, масові відкриті онлайн-курси, системи керування навчанням тощо), середовища розробки програм, засоби моделювання тощо. ІКТ можуть виконувати різноманітні функції, зокрема формування і закріплення у студентів практичних навичок об'єктно-орієнтованої розробки.

Аналіз останніх досліджень та публікацій. Пошук шляхів подолання проблем професійної підготовки майбутніх програмістів у закладах вищої освіти нині є актуальним напрямом наукових досліджень. Результати цих розвідок представлені у роботах Р. Аткинсона, О. Гладких, Л. Гришко, Ю. Жураковського, Л. Зубик, В. Круглика, В. Осадчого, К. Осадчої, В. Седова, З. Сейдаметової, А. Стрюка, Д. Шефер й ін. Окремі аспекти організації процесу вивчення ООП досліджують П. Алексеєвський, І. Барков, Й. Беннедсен, А. Еккердал, М. Касперсен, Й. Кнудсен, О. Мадсен, А. Петров, Г. Рудакова, О. Широкова й ін. Теоретичні й методологічні основи створення ЕЗНП досліджують В. Биков, А. Гуржій, Ю. Дорошенко, М. Жалдак, Л. Зайнутдінова, В. Лапінський, С. Раков, І. Роберт, О. Співаковський, С. Христочевський й ін. Особливості розробки і застосування у навчальному процесі комп'ютерних тренажерів висвітлені у роботах Д. Алюшевої, В. Грибової, Ю. Жарких, Я. Крупського, С. Лисоченко, В. Михалевич, М. Роговець, Г. Осипенкова, Л. Салехової, В. Самойлова й інших науковців.

Отже, проблеми впровадження і педагогічно обґрунтованого застосування ЕЗНП, а також професійної підготовки майбутніх інженерів-програмістів у закладах вищої освіти досліджені достатньо добре. Разом із тим, враховуючи зміни, які постійно відбуваються у ІТ-гаузі, методи і засоби навчання студентів програмної розробки з використанням ООП вимагають подальшого удосконалення.

Формулювання цілей статті. Метою статті є огляд методів і засобів навчання, які можуть бути використані з метою формування у майбутніх інженерів-програмістів професійних компетентностей, необхідних для здійснення об'єктно-орієнтованої розробки.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Методи навчання об'єктно-орієнтованого програмування майбутніх інженерів-програмістів у закладах вищої освіти

Вивчення теоретичних основ об'єктно-орієнтованого підходу, а також конкретних реалізацій й інструментарію ООП у різних мовах програмування, як було зазначено вище, спрямоване на формування у студентів компетентностей, необхідних для успішної професійної діяльності. Забезпечити це можливо лише з використанням різноманітних методів і засобів навчання.

Беззаперечно ефективним є комплексне застосування класичних (задачний підхід, диференційований підхід, проблемний метод, інтерактивні методи) і спеціалізованих (паралельне вивчення двох мов програмування; колективне створення навчального програмного продукту) методів ([Гришко, 2009, с. 47](#)).

На основі проведеного дослідження О. Кривонос пропонує використовувати методи, які наближають навчальний процес до реального програмування: парне програмування (над розробкою програми працюють одночасно два студенти), метод обмеження часу виконання програми і розміру файлу (вимоги до розміру програми і часу її роботи визначаються у завданні, успішність його виконання оцінюється з урахуванням ступеня дотримання цих вимог) ([Кривонос, 2014, с. 124-129](#)).

Комплекс специфічних методів навчання, спрямованих на формування компетентності з програмування, наводить В. Круглик: метод моделювання; проектний; дослідницький; індукція, дедукція та абдукція як методи навчання математики; евристичний метод; метод виконання вправ із пропусками; візуальне моделювання програм; стратегія когнітивного конфлікту; метод побудови ментальних моделей; методу учнівства; парне програмування; програмоване навчання; ітераційний метод ([Круглик, 2018, с. 247](#)). Вважаємо доцільним застосовувати ці методи у процесі навчання майбутніх інженерів-програмістів ООП з урахуванням особливостей зазначеної парадигми.

Пошуки ефективних методів навчання ООП проводять також закордонні дослідники. Вони, зокрема, підкреслюють, що ООП – це перш за все методологія розробки програм, методи якої схожі на природний стиль мислення людини. Для того, щоб продемонструвати і довести студентам потужність цього підходу, доцільно дотримуватись таких етапів введення фундаментальних понять ООП: 1) обговорення основних принципів об'єктно-орієнтованого мислення; 2) введення поняття об'єкта на основі спостереження реального світу; 3) формування поняття класу шляхом абстрагування і узагальнення властивостей безлічі схожих об'єктів; 4) введення поняття спадкування на основі вивченого поняття класу; 5) ілюстрування підкласів шляхом додавання нових властивостей і методів до існуючого класу ([Zhu, & Zhou, 2003](#)).

Деякі підходи до вивчення ООП аналізують Дж. Удварос і М. Губан ([Udvaros, & Gubán, 2016, с. 28-33](#)):

1. Гейміфікація. ООП доцільно викладати з використанням ігрових елементів. Автори зазначають, що цей метод є ефективним під час навчання мови програмування, але може застосовуватися також для навчання концепцій і методів об'єктно-орієнтованого програмування.

2. Змішане електронне навчання. Оскільки для вивчення ООП недостатньо лише аудиторних занять, студенти повинні займатися самостійно. Завданням викладача є надання їм необхідних матеріалів і допомоги, забезпечення контролю і оцінювання з використанням систем керування навчанням.

3. Застосування допоміжних програмних засобів. Автори пропонують створювати і розміщати у LMS інтерактивні додатки для демонстрації фундаментальних концепцій ООП. Вони розробили програму, яка візуалізує поняття «клас», «об'єкт», «властивості класу», «методи класу», «спадкування», полегшуючи їх розуміння для студентів, які мають мінімальні навички програмування.

Метод візуалізації з ігровими елементами використовують для навчання студентів об'єктно-орієнтованому програмуванню французькі дослідники. Вони пропонують навчальний програмний засіб PrOgO (Object-Oriented PrOgramming), ідея якого ґрунтується на ігрових метафорах. Студентам пропонуються готові блоки, маніпулюючи якими вони отримують складні об'єкти. Кожний блок є окремим класом. При розміщенні блоку у сцені автоматично генерується код мовою C++. Після цього можна переглянути методи класу і викликати їх. Програмний засіб не передбачає написання коду: створюючи складні тривимірні об'єкти з готових блоків, студенти аналізують стан, поведінку і взаємодію об'єктів. Отже, гра допомагає продемонструвати такі концепції ООП, як клас, об'єкт, створення об'єкта, виклик метода, передача параметрів, знищення об'єктів. Окрім того, хоча студенти і не пишуть програми власноруч, вони мають можливість вивчати згенерований код, який містить прості інструкції, тому є достатньо легким для розуміння, відтворення і удосконалення ([Djelil, Albouy-Kissi, Albouy-Kissi, Sanchez, & Lavest, 2015, с. 290-291](#)).

У закордонних вишах для вивчення ООП також використовується метод розробки комп'ютерних ігор, розглянутий вище. Наприклад, пропонується застосовувати фреймворк Game, який надає набір готових до використання об'єктів ([Chen, & Cheng, 2007, с. 198](#)).

Розробка комп'ютерної гри у процесі вивчення курсу ООП складається з таких етапів ([Chen, & Cheng, 2007, с. 200](#)):

1. Створення команд у складі двох студентів з метою формування навичок взаємодії у процесі розробки програм, а також полегшення розробки проекту.

2. Вибір теми гри. Студентам пропонують для аналізу ігри, створені раніше іншими групами, на основі чого вони можуть сформулювати власну тему.

3. Остаточне узгодження теми гри.

4. Розробка гри з індивідуальними консультаціями викладача і підтримкою спеціальної групи помічників.

5. Моніторинг виконання проекту. Студенти відображають свій прогрес на спеціально створеному сайті. За результатами контролю, який відбувається щотижня, командам надаються необхідні рекомендації і допомога.

6. Проміжний контроль: команди представляють релізи, за якими можна оцінити ступінь готовності проекту.

7. Підсумкове оцінювання і обчислення рейтингу з урахуванням якості фінальної версії гри і звіту, результатів поточного моніторингу і голосування студентів.

Автори дослідження зазначають, що використання цього підходу допомагає подолати проблеми, які виникають у процесі вивчення ООП, за рахунок посилення мотивації, забезпечення глибокого вивчення ООП, індивідуального підходу ([Chen, & Cheng, 2007, с. 203](#)).

Таким чином, на основі аналізу методів навчання, які нині використовуються у практиці професійної підготовки майбутніх інженерів-програмістів, можна зробити висновок, що усі вони можуть бути використані у процесі формування компетентностей у сфері об'єктно-орієнтованої розробки. Перевагу слід надавати методам, які апелюють до абстрактно-логічного (метод побудови ментальних моделей, стратегія когнітивного конфлікту й ін.) і практичного (парне програмування, метод обмежень, ітераційний метод й ін.) мислення. Окрім того, доцільно послідовно збільшувати застосування інтерактивних методів навчання.

Наприклад, вивчення дисципліни «Програмування» розраховане на вісім семестрів. Вже у другому семестрі студенти розпочинають ознайомлення з окремими поняттями ООП. Далі протягом 3-8 семестрів вони вивчають теоретичні основи об'єктно-орієнтованого програмування, аналізу і моделювання, а також їх практичну реалізацію у мовах програмування. На першому курсі велике значення мають методи викладення нового матеріалу, спрямовані на представлення студентам основ нової для багатьох з них науки, але далі у міру розширення обсягу їхніх знань і умінь слід надавати їм більшу самостійність і збільшувати відповідальність за отримані результати. В ідеалі, на четвертому курсі це дозволить практично повністю замінити класичні лекційні заняття на заняття у форматі змішаного навчання.

Засоби навчання об'єктно-орієнтованого програмування майбутніх інженерів-програмістів у закладах вищої освіти

Досліджуючи використання інформаційних технологій і програмного забезпечення у навчальному процесі, вчені (О. Балалаєва ([Балалаєва, 2013](#); [Балалаєва, 2016](#)), В. Вембер ([Вембер, 2006](#)), Ю. Лобода ([Лобода, 2012](#)) й ін.) зазначають, що на сьогодні існує багато підходів до визначення основних понять і теоретичних положень, а також класифікації цих програмних продуктів.

Наприклад, декілька існуючих класифікацій наводить В. Вембер: за переважним видом навчальної діяльності учня, за способом використання на заняттях, за застосуванням на етапах педагогічного ланцюжка «теорія – практика – контроль» тощо. Дослідниця наголошує, що «не існує не тільки єдиного підходу до класифікації електронних засобів навчального призначення, а й визначеності з термінологією в цій сфері», що виявляється у одночасному функціонуванні синонімічних понять, наприклад: електронні засоби навчального призначення, педагогічні програмні засоби, комп'ютерні системи навчального призначення тощо. Вчена розробила систему

класифікації електронних освітніх ресурсів, виділяючи шість основних груп: електронна навчальна книга, електронний навчальний посібник, електронний підручник, автоматизовані навчальні системи, програмно-методичні комплекси, програмні засоби контролю за навчально-виховним процесом ([Вембер, 2006](#)).

Колектив авторів В. Дем'яненко, Г. Лаврентьєва, М. Шишкіна наводить класифікацію комп'ютерних програм навчального призначення, поділяючи їх на три групи за типом навчальної діяльності: 1) закріплення нового матеріалу: довідкові, додаткові, моделюючі, практикуми; 2) пояснення нового матеріалу: навчальні, демонстраційні; 3) контроль і оцінювання знань: оцінювальні ([Дем'яненко, Лаврентьєва, & Шишкіна, 2013, с. 45](#)).

Автори підручника «Комп'ютерні технології в освіті» називають три групи засобів комп'ютерних технологій, призначених для організації електронного навчання, а саме: електронні навчальні матеріали, комп'ютерні тестові системи, системи керування навчанням ([Жарких, Лисоченко, Сусь, & Третяк, 2012, с. 14](#)).

Класифікувати електронні засоби навчального призначення з використанням фасетного методу пропонує О. Балалаєва. Ієрархічний метод полягає у створенні ієрархічної деревовидної структури шляхом послідовного поділу множини вищого рівня на підмножини. Одним з суттєвих недоліків такого підходу є вимога щодо належності одного об'єкту лише одному класу. Фасетний метод передбачає можливість паралельного поділу загальної множини об'єктів на незалежні підмножини за різними ознаками. У такому випадку немає жорсткої класифікаційної структури і підмножини можуть перетинатися між собою ([Балалаєва, 2013, с. 42-43](#)). Дослідниця наводить різноманітні класифікації електронних засобів навчального призначення і на основі проведеного аналізу пропонує власну класифікацію, яка базується на 14 ознаках, зокрема: за дидактичним призначенням; за реалізацією інтелектуальних функцій; за ступенем інтерактивності тощо ([Балалаєва, 2016, с. 243](#)).

Аналізуючи наведені вище класифікації, можна зробити висновок, що їхні автори відносять комп'ютерні тренажери до різних типів електронних засобів навчального призначення: електронні навчальні посібники (В. Вембер); засоби практичної підготовки (Ю. Лобода); засоби закріплення нового матеріалу – практикуми (В. Дем'яненко, Г. Лаврентьєва, М. Шишкіна); електронні навчальні матеріали – засоби для набуття практичних навичок (Ю. Жарких, С. Лисоченко, Б. Сусь, О. Третяк). Спільним для більшості з наведених підходів є підкреслення ролі електронного тренажера як засобу практичної підготовки.

Наведемо визначення, запропоноване Ю. Лободою: «Комп'ютерний тренажер – пристрій, у якому як модель об'єкта управління, так і робоче місце студентів й інструктора реалізовані завдяки комп'ютерним засобам.» ([Лобода, 2012](#)).

Колектив авторів В. Грибова, Г. Осипенков, С. Сова називає три типи тренажерів за задачами, які вони виконують: тренажери для відпрацювання знань і практичних навичок; тренажери для відпрацювання моторно-рефлекторних реакцій і навичок (до їх складу мають входити спеціалізовані апаратні компоненти); змішаний тип тренажерів. Вони зазначають, що застосування тренажерів полегшує професійну підготовку фахівців, допомагає підвищити її ефективність і якість, знижує її вартість, надає можливість сформувати практичні навички, зокрема, шляхом моделювання різноманітних сценаріїв ([Грибова, Осипенков, & Сова, 2009, с. 27-28](#)).

Електронні тренажери є елементом освітнього середовища, яке ґрунтується на застосуванні методів моделювання реальних процесів. Таке моделювання надає низку переваг: моделі мають нижчу вартість, ніж їхні реальні прототипи; модель легше створити; моделювання дозволяє уникати реальних небезпечних ситуацій; моделювання можна призупинити і проаналізувати ситуацію, що у реальному житті не завжди можливо; студенти є активними учасниками моделювання, а не просто слухачами або спостерігачами: вони беруть на себе відповідальність, шукають шляхи для досягнення успіху, організують взаємодію тощо; моделювання мотивує до навчання, оскільки активна участь у діяльності спонукає студентів більш глибоко дізнаватися про її предмет ([Educational Simulations, 2011](#)).

У практиці закордонних закладів вищої освіти електронні тренажери також використовуються для формування у майбутніх інженерів-програмістів розуміння і навичок застосування методів ООП. Особливістю цих програмних засобів є широке використання візуалізації. Розглянемо деякі приклади.

Досвід впровадження програмного тренажеру «Traffic Simulation» для ознайомлення студентів з базовими поняттями ООП і формування навичок використання класів подає В. Проулкс. Тренажер представляє собою спрощену модель дорожнього руху, у складі якої виділені такі об'єкти: автомобільний шлях, автомобіль, група автомобілів, світлофор. У програмі реалізовані певні шаблони їхньої поведінки. Завдання студента полягає у тому, щоб змоделювати реальну ситуацію дорожнього руху, використовуючи наявні класи і методи або створюючи нові у межах своїх повноважень, наприклад: розробити механізм визначення вільного місця на дорозі; розробити для автомобіля метод визначення можливості руху уперед; розробити для групи автомобілів методи для додавання нових екземплярів автомобілів і видалення тих, що поїхали далі і т.ін. Авторка пропонує декілька способів застосування цього тренажеру у навчальному процесі, що залежить від рівня готовності студентів до засвоєння нового матеріалу ([Proulx, 1998, c. 49](#)).

Метод візуалізації для вивчення основ об'єктно-орієнтованого підходу також застосовують М. Естевез і А. Мендес. Дослідники розробили освітнє середовище «ООР-Anim», яке відображує процес виконання програми, зокрема, взаємодію між об'єктами. Студентам пропонуються класи, на прикладі яких часто пояснюються базові концепції ООП (зображення, коло, прямокутник, трикутник) ([Esteves, & Mendes, 2003](#); [Esteves, & Mendes, 2004](#)). Вікно програмного засобу складається з чотирьох областей: в одній з них студент пише об'єктно-орієнтований код для виконання завдання; в іншій відбувається візуалізація виконання цього коду. Окрім того, передбачена область для виведення повідомлень і область, де розміщені елементи управління процесом анімації ([Esteves, & Mendes, 2003](#)).

Середовище «ООР-Anim» допомагає зрозуміти базові концепції ООП, а також дозволяє візуалізувати внутрішні процеси, які відбуваються під час виконання програми, зрозуміти, як впливають на її перебіг зміни у коді, зроблені студентом, що полегшує пошук помилок.

Дослідники пропонують різні способи застосування цього програмного продукту у навчальному процесі. Наприклад, для організації самостійної і групової роботи студентів на лабораторних заняттях, зокрема у якості засобу заохочення слабких студентів до наполегливого пошуку способу розв'язання поставленої задачі. «ООР-

Anim» також можна застосовувати на лекціях для демонстрації або для створення проблемних ситуацій ([Esteves, & Mendes, 2004, с. 9-11](#)).

Перевагою проаналізованих тренажерів є адаптація до потреб навчального процесу у певному виші. Окрім того, у таких середовищах студентам надаються шаблони, спираючись на які вони можуть розробляти власні класи і самостійно робити висновки, що полегшує вивчення понять ООП.

Таким чином, основним засобом, який використовується у процесі навчання об'єктно-орієнтованого програмування, є середовища розробки програм, за допомогою яких студенти набувають навичок практичної реалізації концепцій ООП. Ефективним є застосування додаткових ЕЗНП, які дозволяють абстрагуватися від тонкощів конкретної мови програмування і зосередитися на концептуальних засадах даної парадигми програмування. Окрім того, доцільним є впровадження електронних тренажерів, які спрямовані на самостійне відпрацювання практичних навичок.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Удосконалення системи методів і засобів професійної підготовки майбутніх інженерів-програмістів у закладах вищої освіти до практичної діяльності за фахом є одним з важливих завдань вищої школи. У статті були проаналізовані методи і засоби навчання, які нині використовуються для навчання майбутніх інженерів-програмістів, на основі чого виділені способи виконання зазначеного завдання. Зокрема, шляхом впровадження нових підходів до використання електронних засобів навчального призначення, які ґрунтуються на урахуванні індивідуальних навчальних потреб студентів. Це можуть бути середовища, які реалізують метод візуалізації понять ООП, тренажери, за допомогою яких можна організувати самостійне тренування з автоматичною перевіркою результатів й ін. Окрім того, необхідно послідовно підвищувати роль інтерактивних методів навчання, які спонукають студентів до самостійного формулювання висновків, застосування вже наявних знань і пошуку нових. Подальші дослідження спрямовані на розробку системи методів і засобів навчання, адаптованих до особливостей ООП як предмета вивчення у закладах вищої освіти і спрямованих на ефективне формування відповідних професійних компетентностей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- Балалаєва, О. Ю. (2013). Фасетні класифікації електронних засобів навчального призначення. *Інформаційні технології і засоби навчання*, 6 (38), 41-52. Взято з <http://journal.iitta.gov.ua/index.php/itlt/article/view/926>.
- Балалаєва, О. Ю. (2016). *Проектування електронних посібників з латинської мови для вищих аграрних навчальних закладів*. (Дис. канд. пед. наук). Національний університет біоресурсів і природокористування України, Київ.
- Вембер, В. П. (2006). Навчально-методичні вимоги до електронного підручника. *Науковий часопис НПУ імені М.П. Драгоманова Серія № 2. Комп'ютерно-орієнтовані системи навчання*, 4 (11), 50-56. Взято з <http://enpuir.npu.edu.ua/handle/123456789/865?mode=simple>.

- Грибова, В., Осипенков, Г., & Сова, С. (2009). Концепция разработки диагностических компьютерных тренажеров на основе знаний. В Markov, K., Velychko, V., Ivanova, K., & Mitov, I. (Eds.). *Human Aspects of Artificial Intelligence. International Book Series «Information Science & Computing», (12). Supplement to the International Journal «Information Technologies & Knowledge», 3*, pp. 27-33. Sofia, Bulgaria: Institute of Information Theories and Applications FOI ITHEA.
- Гришко, Л. В. (2009). *Методична система навчання основ програмування майбутніх інженерів-програмістів*. (Автореф. дис. канд. пед. наук). Національний педагогічний університет імені М. П. Драгоманова, Київ.
- Дем'яненко, В. М., Лаврентьєва, Г. П., & Шишкіна, М. П. (2013). Методичні рекомендації щодо добору і застосування електронних засобів та ресурсів навчального призначення. *Комп'ютер у школі та сім'ї*, (1), 44-48.
- Жарких, Ю. С., Лисоченко, С. В., Сусь, Б. Б., & Третьяк, О. В. (2012). *Комп'ютерні технології в освіті: навч. посібн.* Київ: Видавничо-поліграфічний центр «Київський університет».
- Кривонос, О. М. (2014). *Формування інформаційно-комунікаційних компетентностей майбутніх учителів інформатики в процесі навчання програмування*. (Дис. канд. пед. наук). Інститут інформаційних технологій і засобів навчання НАПН України, Київ.
- Круглик, В. С. (2018). *Система підготовки майбутніх інженерів-програмістів до професійної діяльності у вищих навчальних закладах*. (Дис. доктора пед. наук). Запорізький національний університет, Запоріжжя.
- Лобода, Ю. Г. (2012). Електронні засоби навчання: структура, зміст, класифікація. *Інформаційні технології і засоби навчання*, 2 (28). Взято з <http://journal.iitta.gov.ua/index.php/itlt/article/view/649/492>.
- Chen, W. K., & Cheng, Y. C. (2007). Teaching Object-Oriented Programming Laboratory With Computer Game Programming. *IEEE Transactions on Education*, 50 (3), 197-203.
- Djelil, F., Albouy-Kissi, B., Albouy-Kissi, A., Sanchez, E., & Lavest, J. (2015). Towards a 3D Virtual Game for Learning Object-Oriented Programming Fundamentals and C++ Language - Theoretical Considerations and Empirical Results. In *Proceedings of the 7th International Conference on Computer Supported Education*, 289-294.
- Educational Simulations*. (2011). *Creativeteachingsite.com*. Retrieved from <http://www.creativeteachingsite.com/edusims.html>.
- Esteves, M., & Mendes, A. J. (2003). OOP-Anim, a system to support learning of basic object oriented programming concepts. In *Proceedings of the 4th International Conference on Computer Systems and Technologies (CompSysTech 2003)*. Retrieved from <http://ecet.ecs.uniruse.bg/cst/Docs/proceedings/S4/IV-6.pdf>.
- Esteves, M., & Mendes, A. J. (2004). A simulation tool to help learning of object oriented programming basics. In *Proceedings of the 34th Annual Frontiers in Education (FIE 2004)*, F4C-7-12.
- Proulx, V. K. (1998). Traffic simulation: A case study for teaching object oriented design. In *Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education*, 48-52.
- Udvaros, J., & Gubán, M. (2016). Demonstration the Class, Object and Inheritance Concepts by Software. *Acta Didactica Napocensia*, 9 (1), 23-34.
- Zhu, H., & Zhou, M. (2003). Methodology First and Language Second: a Way to Teach Object-Oriented Programming. In *Proceedings of the OOPSLA'03*. Retrieved from <https://faculty.nipissingu.ca/haibinz/research/oopsla03.pdf>.

REFERENCES (TRANSLATED AND TRANSLITERATED)

- Balalaieva, O. Yu. (2013). Facet classifications of e-learning tools. *Information Technologies and Learning Tools*, 6 (38), 41-52. Retrieved from <http://journal.iitta.gov.ua/index.php/itlt/article/view/926>. (in Ukrainian)
- Balalaieva, O. Yu. (2016). *Designing of electronic manuals from the Latin language for higher agricultural educational institutions*. (Candidate of Pedagogic Sciences Thesis). Natsionalnyi universytet bioresursiv i pryrodokorystuvannia Ukrainy, Kyiv. (in Ukrainian)
- Vember, V. P. (2006). Educational and methodical requirements for an electronic textbook. *Naukovyi chasopys NPU imeni M.P. Drahomanova Seriiia No 2. Kompiuterno-oriientovani systemy navchannia*, 4 (11), 50-56. Retrieved from <http://enpuir.npu.edu.ua/handle/123456789/865?mode=simple>. (in Ukrainian)
- Hrybova, V., Osypenkov, H., & Sova, S. (2009). The concept of developing diagnostic computer simulators on the basis of knowledge. In Markov, K., Velychko, V., Ivanova, K., & Mitov, I. (Eds.). *Human Aspects of Artificial Intelligence. International Book Series «Information Science & Computing», (12). Supplement to the International Journal «Information Technologies & Knowledge», 3*, pp. 27-33. Sofia, Bulgaria: Institute of Information Theories and Applications FOI ITHEA. (in Russian)
- Hryshko, L. V. (2009). *Methodical system of studying the basics of programming of future engineers-programmers*. (Author's abstract of Candidate of Pedagogic Sciences Thesis). Natsionalnyi pedahohichnyi universytet imeni M. P. Drahomanova, Kyiv. (in Ukrainian)
- Demianenko, V. M., Lavrentieva, H. P., & Shyshkina, M. P. (2013). Methodical recommendations on the selection and use of electronic resources and training resources. *Kompiuter u shkoli ta simi*, (1), 44-48. (in Ukrainian)
- Zharkyykh, Yu. S., Lysochenko, S. V., Sus, B. B., & Tretiak, O. V. (2012). *Computer technology in education: textbook*. Kyiv: Vydavnycho-polihrafichnyi tsentr "Kyivskiy universytet". (in Ukrainian)
- Kryvonos, O. M. (2014). *Formation of information and communication competences of future teachers of informatics in the process of programming of programming*. (Candidate of Pedagogic Sciences Thesis). Instytut informatsiinykh tekhnolohii i zasobiv navchannia NAPN Ukrainy, Kyiv. (in Ukrainian)
- Kruhlyk, V. S. (2018). *System of training of future engineers-programmers for professional activity in higher educational establishments*. (Candidate of Pedagogic Sciences Thesis). Zaporizkyi natsionalnyi universytet, Zaporizhzhia. (in Ukrainian)
- Loboda, Yu. H. (2012). E-learning tools: structure, content, classification. *Information Technologies and Learning Tools*, 2 (28). Retrieved from <http://journal.iitta.gov.ua/index.php/itlt/article/view/649/492>. (in Ukrainian)
- Chen, W. K., & Cheng, Y. C. (2007). Teaching Object-Oriented Programming Laboratory With Computer Game Programming. *IEEE Transactions on Education*, 50 (3), 197-203.
- Djelil, F., Albouy-Kissi, B., Albouy-Kissi, A., Sanchez, E., & Lavest, J. (2015). Towards a 3D Virtual Game for Learning Object-Oriented Programming Fundamentals and C++ Language - Theoretical Considerations and Empirical Results. In *Proceedings of the 7th International Conference on Computer Supported Education*, 289-294. (in English)
- Educational Simulations*. (2011). *Creativeteachingsite.com*. Retrieved from <http://www.creativeteachingsite.com/edusims.html>. (in English)
- Esteves, M., & Mendes, A. J. (2003). OOP-Anim, a system to support learning of basic object oriented programming concepts. In *Proceedings of the 4th International Conference on Computer Systems and Technologies (CompSysTech 2003)*. Retrieved from <http://ecet.ecs.uni-ruse.bg/cst/Docs/proceedings/S4/IV-6.pdf>. (in English)

- Esteves, M., & Mendes, A. J. (2004). A simulation tool to help learning of object oriented programming basics. In *Proceedings of the 34th Annual Frontiers in Education (FIE 2004)*, F4C-7-12. (in English)
- Proulx, V. K. (1998). Traffic simulation: A case study for teaching object oriented design. In *Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education*, 48-52.
- Udvaros, J., & Gubán, M. (2016). Demonstration the Class, Object and Inheritance Concepts by Software. *Acta Didactica Napocensia*, 9 (1), 23-34. (in English)
- Zhu, H., & Zhou, M. (2003). Methodology First and Language Second: a Way to Teach Object-Oriented Programming. In *Proceedings of the OOPSLA'03*. Retrieved from <https://faculty.nipissingu.ca/haibinz/research/oopsla03.pdf>. (in English)

Матеріал надійшов до редакції 10.03.2018