

Ірина Шаклеїна, кандидат фізико-математичних наук, доцент,
доцент кафедри інформаційних систем та мереж
Національного університету "Львівська політехніка",
доцент кафедри фізики та інформаційних систем
Дрогобицького державного педагогічного університету імені Івана Франка
ORCID: <https://orcid.org/0000-0003-0809-1480>

ХМАРНІ СЕРВІСИ ДЛЯ ПІДТРИМКИ КОМАНДНОЇ РОБОТИ У ВИВЧЕННІ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

У статті проаналізовано освітній потенціал хмарних сервісів для організації командної проєктної діяльності під час вивчення об'єктно-орієнтованого програмування. Описано методику реалізації проєкту з використанням GitHub, Replit, Gitpod, Notion та інших інструментів. Особливу увагу приділено системі оцінювання командної взаємодії, що враховує технічні, комунікаційні та організаційні показники.

Ключові слова: хмарні сервіси; об'єктно-орієнтоване програмування; командна проєктна діяльність; GitHub; Replit; Notion; система оцінювання; ІТ-освіта.

Рис. 1. Лім. 10.

Iryna Shakleina, Ph.D. (Physical and Mathematical Sciences), Associate Professor,
Associate Professor of the Information Systems and Networks Department,
Lviv Polytechnic National University,
Associate Professor of the Physics and Information Systems Department,
Drohobych Ivan Franko State Pedagogical University
ORCID: <https://orcid.org/0000-0003-0809-1480>

CLOUD SERVICES TO SUPPORT TEAMWORK IN LEARNING OBJECT-ORIENTED PROGRAMMING

The article considers the pedagogical feasibility and effectiveness of organizing team project activities when studying object-oriented programming (OOP) using modern cloud services. The emphasis is on using cloud digital tools (GitHub, Replit, Gitpod, Notion, etc.) as components of a learning environment that supports teamwork, planning, code storage, documentation creation, and presentation of results.

The proposed approach involves implementing a team project, during which students create a software product in compliance with clear requirements for implementing the basic principles of OOP. The key components of the project are maintaining team documentation, working in a cloud repository, adhering to coding standards, and interacting through digital services. Such an organization reproduces processes close to modern software development practices in the IT industry.

The paper describes a multi-component system for assessing team interaction, which includes technical, organizational, and communication indicators. It considers both the individual contribution of students and the collective result of the team and is based on a quantitative analysis of activity in repositories, code quality, and collaboration efficiency.

It is concluded that integrating cloud services into the team project learning process contributes to a deeper understanding of the principles of object-oriented programming and the formation of professional and social competencies. It is a promising direction in the training of future IT specialists.

Keywords: cloud services; object-oriented programming; team project work; GitHub; Replit; Notion; assessment system; IT education.

Постановка проблеми. Об'єктно-орієнтоване програмування (ООП) є ключовою парадигмою сучасної розробки програмного забезпечення, що ґрунтується на таких поняттях, як абстракція, інкапсуляція, спадкування та поліморфізм. Засвоєння цих понять має не лише теоретичне, а й виразно практичне значення, оскільки дозволяє майбутнім фахівцям ефективно проєктувати й реалізовувати складні програмні системи.

Водночас сучасний ІТ-ринок висуває підвищені вимоги не лише до індивідуальних, а й до команд-

них навичок програмістів, особливо в контексті роботи над спільними проєктами, використання систем контролю версій і командних стандартів розробки програмних продуктів. Традиційні форми організації навчання, орієнтовані переважно на індивідуальну роботу студентів, часто не дають можливості повною мірою сформувати такі компетентності.

Ситуація ускладнюється ще й сучасними умовами навчання: дистанційним або змішаним форматом, необхідністю забезпечення доступу до навчальних ресурсів незалежно від часу і місця пере-

бування студентів, а також потребою швидко організувати колективну роботу над навчальними проектами. У цьому контексті актуальним стає використання хмарних сервісів, які дозволяють організувати спільну роботу студентів, зберігання та контроль версій коду, а також спрощують доступ до середовищ розробки без складної локальної інсталяції.

Серед таких сервісів найбільшого поширення набули GitHub, Replit, Gitpod, Visual Studio Code Spaces та інші. Вони надають можливість працювати з кодом у браузері, створювати командні репозиторії, контролювати зміни та взаємодіяти з одногрупниками в режимі реального часу. Хоча GitLab у хмарній версії є функціональним аналогом GitHub, його інтерфейс та налаштування можуть бути складнішими для студентів-початківців, а використання розширених функцій у більшості випадків потребує платних тарифів або окремого хостингу.

Таким чином, актуальною є проблема обґрунтування та педагогічного моделювання використання доступних хмарних сервісів для організації ефективної командної роботи студентів у процесі вивчення ООП, з урахуванням сучасних умов навчання та вимог до фахових ІТ-компетентностей. Це дозволить не лише підвищити мотивацію та залученість, а й наблизити навчальний процес до реальних умов сучасної розробки програмного забезпечення.

Аналіз останніх досліджень і публікацій. Проблема формування практичних і командних навичок у студентів під час вивчення ООП широко досліджується у сучасній педагогічній та методичній літературі. У багатьох роботах відзначено, що студенти часто відчують труднощі із засвоєнням абстрактних понять ООП, таких як інкапсуляція, успадкування та поліморфізм, особливо на початкових етапах навчання [1, 2]. Автори підкреслюють, що ці труднощі обумовлені як складністю самої парадигми, проблемами у організації ефективної навчально-пізнавальної діяльності студентів та недостатньою кількістю практичних завдань, наближених до реальних умов розробки.

Одним із ефективних засобів подолання цих труднощів дослідники називають організацію командної роботи над навчальними проектами. У роботах [3–5] зазначено, що залучення студентів до колективної діяльності з розробки програмних продуктів сприяє глибшому розумінню принципів розроблення програмних продуктів і водночас розвиває важливі *soft skills* – відповідальність, вміння комунікувати та узгоджувати роботу в команді. Дослідники вказують, що поєднання індивідуальних і групових завдань дозволяє підвищити мотивацію студентів і покращити результати навчання. Зокрема, робота [5] описує досвід стимулювання співпраці та командної діяльності студентів у

контексті курсу програмної інженерії на третьому курсі бакалаврської програми з інформатики. Студентам було запропоновано розробити програмний проект у командах від 3 до 5 студентів для підсумкового іспиту курсу. В якості платформи для спільної розробки програмного забезпечення використовували GitHub а для аналізу якості коду проектів – інструмент SonarQube. Автори говорять про проявлений студентами інтерес до роботи в командах над розробкою своїх проектів та використанню інструментів, які використовуються при в ІТ-сфері.

Окрему увагу у сучасних публікаціях приділено використанню хмарних сервісів, що дають змогу організувати ефективну командну роботу над проектами. Так, у роботі [6] розглядається досвід застосування GitHub у викладанні програмування. Автори досліджують використання репозиторіїв GitHub, наданих викладачами, як інструменту для підтримки викладання основних аспектів навчальної програми з програмування, та аналізують роботу студентів над проектами. Серед ключових висновків було те, що інтеграція репозиторіїв GitHub відкриває кілька можливостей для співпраці, розширення знань, самостійного навчання та зниження стресу. Однак, такий підхід також призвів до поширення командних та технічних проблем, через які студенти знаходили власні шляхи подолання цих труднощів.

Ще один напрям досліджень стосується застосування хмарних IDE (Replit, Gitpod, Visual Studio Code Spaces), які дозволяють студентам працювати з кодом без складної локальної інсталяції, безпосередньо в браузері. Робота [7] присвячена дослідженням, проведеним вступному курсі програмування на бакалавраті з комп'ютерної інженерії. Для підтримки навчання студентів використовувалися такі хмарні сервіси, як GitHub Classroom, Visual Studio Code та Replit. У статті обговорюються переваги та недоліки використання згаданих інструментів для вступного курсу програмування. Результати досліджень показали позитивний аспект використання даного підходу як з точки зору викладачів, так і студентів.

У публікаціях [8, 9] зазначається, що такі інструменти спрощують входження в командну розробку, роблять процес більш доступним і гнучким, особливо за умов дистанційного чи змішаного навчання. Зокрема, у [8] за результатами досліджень зазначено, що групова робота із застосуванням онлайнового середовища Replit, дозволяє студентам працювати спільно над проектами, розвиваючи навички командної роботи та вирішення складних алгоритмічних задач.

Разом з тим дослідники звертають увагу на обмеження у впровадженні хмарних сервісів у навчальний процес: потребу в підготовці викладачів, необхідність навчання студентів основам ро-

боти з системами контролю версій, а також технічні обмеження безкоштовних тарифів.

Водночас складні корпоративні хмарні платформи, такі як Microsoft Azure, AWS чи Google Cloud Platform, хоч і надають потужні інструменти для командної розробки, зазвичай є надмірно складними для студентів перших курсів і потребують оплати чи окремих освітніх грантів [10], що робить їх менш доцільними для застосування у викладанні ООП.

Таким чином, аналіз сучасних досліджень свідчить, що використання доступних хмарних сервісів, зокрема GitHub і хмарних IDE, має значний потенціал для підвищення ефективності навчання ООП та розвитку командних навичок студентів. Однак комплексних методичних підходів до організації командної роботи із застосуванням цих інструментів на рівні навчальних програм поки що розроблено недостатньо, що й зумовлює необхідність подальшого розроблення цілісних методик, які поєднують технічні та педагогічні аспекти організації командної роботи на базі хмарних сервісів, адаптованих до потреб сучасної вищої освіти.

Мета статті. Метою пропонованої статті є дослідження доцільності та педагогічного потенціалу використання хмарних сервісів (зокрема GitHub, Replit, Gitpod та інших) для організації командної роботи студентів під час вивчення об'єктно-орієнтованого програмування; аналіз їхнього впливу на формування практичних навичок програмування, розвиток командної взаємодії та підвищення мотивації до навчання, а також визначення важливості використання цих інструментів для формування командних і професійних компетентностей майбутніх фахівців.

Виклад основного матеріалу. Сучасна підготовка фахівців у галузі інформаційних технологій вимагає переосмислення традиційних підходів до викладання дисциплін програмування, зокрема ООП. Теоретичні знання щодо об'єктної моделі мають бути доповнені практикою застосування концепцій моделювання в реальних або наближених до реальних умовах командної розробки програмного забезпечення. Особливої актуальності ця проблема набуває в умовах зростання попиту на soft skills, дистанційного та змішаного формату навчання, а також враховуючи виклики, пов'язані з освітнім процесом в умовах війни в Україні.

Загальноприйнята структура навчального курсу з ООП часто побудована на модульному принципі з розділенням матеріалу на теми, кожна з яких супроводжується лабораторними роботами або вправами. Такий підхід сприяє поетапному ознайомленню студентів із синтаксичними та структурними особливостями ООП, однак ускладнює формування цілісного уявлення про розробку програмної системи в об'єктно-орієнтованому стилі. Як наслідок,

часто студенти демонструють знання синтаксису, але мають труднощі з побудовою повноцінної архітектури програмного продукту, що поєднує практичну реалізацію принципів абстракції, наслідування, композиції класів, поліморфізму та інкапсуляції.

Окрім дидактичних труднощів, освітній процес останніх років характеризується впливом зовнішніх дестабілізуючих чинників, що ускладнюють реалізацію традиційних форм очного навчання й вимагають пошуку альтернативних, зокрема цифрових і хмароорієнтованих, рішень.

Зокрема, впровадження хмарних сервісів, дозволяє організовувати роботу над проектами незалежно від технічних можливостей конкретного користувача або фізичного місця його перебування. За умови ефективного використання такі ресурси виступають не лише як технічні інструменти, а як цілісне цифрове середовище навчання, що забезпечує доступ до спільної розробки, обміну знаннями, керування процесами та презентації результатів. Водночас, такий підхід вимагає відповідної методичної підготовки як студентів, так і викладачів.

Запропоноване у статті рішення полягає у впровадженні в курс ООП командного проектного завдання з обов'язковим використанням хмарних сервісів, таких як GitHub, Replit, Gitpod, Notion тощо. Умови завдання передбачали розробку програмного застосунку (наприклад, чат, планувальник задач, мінігра) мовою C# або Java, у межах якого вони мають застосувати принципи об'єктно-орієнтованого моделювання, одночасно працюючи в умовах розподіленої командної взаємодії. Змістове наповнення завдання та організація командної роботи орієнтовані на відтворення практик, наближених до реальних процесів розроблення програмного забезпечення. На відміну від традиційних методів, що орієнтовані на індивідуальні завдання, цей формат дозволяє студентам спільно створювати програмний продукт у хмарному середовищі, застосовуючи принципи ООП на практиці та розвиваючи командні компетентності.

Команди формувалися з 3–4 студентів. Усі етапи роботи від постановки задачі до захисту проекту реалізовувалися із залученням відповідних хмарних сервісів. Центральною платформою, що забезпечувала технічну координацію роботи, виступав GitHub, який виконував функції:

- системи контролю версій і колективного редагування коду;
- майданчика для інтеграції з хмарними IDE (Replit, Gitpod, Codespaces);
- централізованого сховища документації (README, Wiki, markdown-файли).

Робота в репозиторії здійснювалася за попередньо наданим шаблоном, що включав типову структуру каталогів, зразок ReadMe-файлу, правила

ХМАРНІ СЕРВІСИ ДЛЯ ПІДТРИМКИ КОМАНДНОЇ РОБОТИ У ВИВЧЕННІ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

використання гілок і механізми злиття змін через pull requests. Особлива увага приділялася розвитку навичок командної взаємодії через використання інструментів для коментування коду, фіксації змін і документування завдань.

Окрім GitHub студенти активно використовували інші хмарні сервіси напрямки застосування яких та можливі зв'язки між ними наведені на рис. 1.

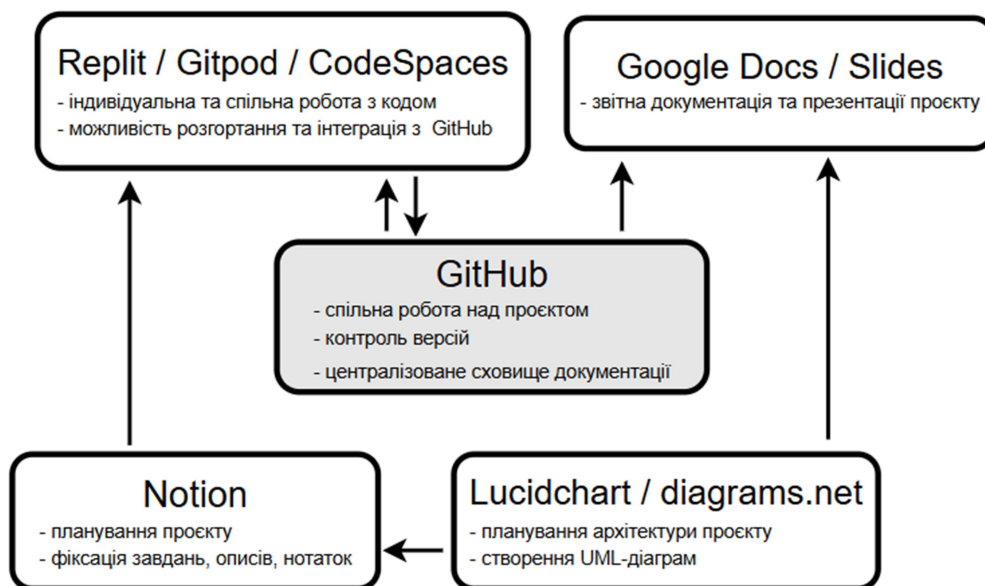


Рис. 1. Хмарні сервіси для проектної роботи з ООП

Хмарні сервіси, що використовуються в рамках проектної роботи, виконують не лише технічну функцію, а й створюють навчальне середовище нового типу. Вони дозволяють студентам працювати в умовах, наближених до реальних DevOps-процесів, формують у них розуміння сучасних практик командної розробки, сприяють формуванню відповідальності та самоорганізації, а також забезпечують наскрізну цифрову грамотність. Застосування зазначених вище інструментів забезпечує послідовну інтеграцію практики та теорії, формує культуру відкритої взаємодії та командної відповідальності.

Комплексне використання зазначених інструментів сприяло розвитку в студентів не лише технічної, а й організаційної та комунікаційної компетентностей. Особливо ефективним виявилось використання типової зв'язки Replit – GitHub – Notion, яка забезпечувала повноцінну підтримку життєвого циклу програмного проєкту: написання та запуск коду в Replit, контроль версій і колективна розробка через GitHub, планування та фіксація рішень у середовищі Notion. Ця конфігурація стала основою в більшості студентських команд і показала найкращі результати з точки зору організованості роботи та зручності взаємодії.

Окрім технічної реалізації, важливою складовою є організація команди: розподіл ролей, підтримка комунікації, дотримання дедлайнів, зві-

ність. Таким чином, формуються не лише предметні IT-компетентності, а й соціальні: відповідальність, здатність до співпраці, комунікація в команді.

З метою забезпечення об'єктивного оцінювання командної роботи було розроблено систему, що включає самооцінювання, взаємооцінювання та аналітику активності команди (кількість комітів, участь у pull requests, обговорення тощо). Такий підхід дозволяє збалансовано враховувати як індивідуальні, так і колективні результати діяльності.

Для кількісної оцінки ефективності командної взаємодії було запропоновано багатокомпонентну систему метрик, що охоплює технічні, комунікаційні та організаційні аспекти. Зокрема, було використано наступні критерії:

- *технічні показники*: відповідність проєкту вимогам об'єктно-орієнтованого підходу (реалізація принципів інкапсуляції, успадкування, композиції класів, поліморфізму) та дотримання стандартів кодування; кількість та якість commit-ів (регулярність, інформативність повідомлень); підготовка звітних матеріалів (опис проєкту, UML-діаграма класів, інструкції щодо запуску); покриття коду тестами.

- *організаційні показники*: рівномірність розподілу обов'язків між учасниками; здатність до самоорганізації та взаємопідтримки; адаптивність до змін у процесі розробки; рівень групової рефлексії та самооцінки.

- *комунікаційні показники*: активність участі в обговореннях; здатність до конструктивного вирішення конфліктів; ефективність планування та дотримання дедлайнів; якість презентації результатів.

Кожен показник оцінювався окремо за шкалою від 0 до 10. На основі цих даних обчислювався інтегральний індекс командної ефективності (ІКЕ). Така система дозволяє не лише зафіксувати підсумковий результат, а й виявити сильні та слабкі сторони командної взаємодії, зокрема ті, що потребують додаткової педагогічної уваги.

Методика має високу адаптивність до різних рівнів академічної підготовки. Студенти-початківці працюють з інструкціями, шаблонами проєктів, підказками щодо структури класів, тоді як більш досвідчені можуть обирати складніші теми, інтегрувати сторонні бібліотеки, автоматичне тестування тощо. Завдяки такій диференціації завдань забезпечується індивідуальна освітня траєкторія. Варто відмітити також, що ефективності проєктної діяльності з використанням зазначених цифрових інструментів сприяв той факт, що в попередніх навчальних курсах, таких як "Алгоритмізація та програмування", "Інформаційні технології" та інших, активно використовувались більшість з зазначених ресурсів.

У процесі впровадження методики було зафіксовано низку труднощів:

- початкові складнощі в роботі з Git/GitHub, особливо при злитті гілок і виникненні конфліктів;
- нерівномірний розподіл обов'язків у командах, що вимагало введення форм саморефлексії та взаємооцінювання;
- недостатній досвід у плануванні завдань і дедлайнів, який частково компенсувався використанням цифрових дошок задач (Notion, GitHub Projects);
- обмеження безкоштовних тарифів деяких хмарних IDE, що іноді ускладнювало роботу з проєктами (наприклад, обмеження кількості осіб для одночасної спільної роботи над проєктом в Replit).

Разом із тим результати проєктної діяльності свідчать про значний педагогічний ефект: студенти виявляли зростання мотивації, покращення розуміння взаємозв'язків між класами, активне включення в роботу, взаємну підтримку та здатність працювати в умовах обмеженого ресурсу. Захист проєктів, організований у форматі відкритих презентацій перед академічною групою, підсилив відчуття відповідальності за спільний результат і дозволив реалізувати рефлексивну складову навчального процесу.

У студентів, що працювали над командним проєктом, зафіксовано вищий рівень розуміння міжкласових зв'язків, кращу здатність до абстрагу-

вання й проєктування, а також вищу загальну успішність за результатами курсу. Було також зафіксовано покращення у мотивації студентів і якості фінальних проєктів порівняно з попередніми роками, коли використовувались лише індивідуальні завдання.

Таким чином, використання хмарних сервісів для підтримки командної проєктної роботи в курсі ООП виявилось ефективним педагогічним рішенням, яке дозволяє одночасно вирішувати кілька важливих дидактичних завдань – від формування предметних компетентностей до розвитку навичок командної взаємодії, самоорганізації, відповідальності й ділової комунікації. Такий підхід заслуговує на подальше поширення та адаптацію до інших курсів і освітніх контекстів.

Висновки. Розвиток цифрових технологій, зокрема хмарних сервісів, відкриває нові можливості для організації ефективного та гнучкого освітнього процесу в технічних дисциплінах. В умовах змішаного та дистанційного навчання, особливо в період воєнного стану в Україні, ці засоби стають критично важливими для забезпечення безперервності навчання та підтримки студентської взаємодії.

У статті було обґрунтовано доцільність застосування хмарних сервісів для підтримки командної проєктної діяльності під час вивчення курсу об'єктно-орієнтованого програмування. Запропонований підхід дозволяє поєднати засвоєння теоретичних основ ООП із практикою командної розробки програмного забезпечення, що сприяє формуванню у студентів не лише предметних, але й соціальних та професійних компетентностей. Досвід впровадження індивідуального завдання у форматі командного проєкту із застосуванням GitHub, Gitpod, Replit, Notion та інших хмарних сервісів підтвердив ефективність такого формату в навчальному процесі.

Проведене дослідження дозволяє зробити такі узагальнення:

- хмарні сервіси забезпечують технічну та організаційну підтримку командної роботи навіть за умов обмежених ресурсів;
- командна проєктна діяльність мотивує студентів, сприяє кращому засвоєнню принципів ООП та розвиває навички, затребувані в професійному середовищі;
- впровадження хмарних інструментів доцільно поєднувати із навчанням основ управління версіями, командної взаємодії та планування роботи;
- важливо враховувати технічні, психологічні та організаційні виклики, що виникають під час реалізації проєктів, і своєчасно реагувати на них через педагогічну підтримку, рефлексію та надання методичних рекомендацій.

Перспективи подальших досліджень вбачаються у вивченні впливу подібних підходів на рівень

сформованості професійних компетентностей випускників, розробці типових моделей впровадження проєктного навчання в ІТ-дисциплінах, а також у вдосконаленні цифрового інструментарію, адаптованого до потреб студентів та умов української освітньої реальності.

ЛІТЕРАТУРА

1. Бардус І. Методика фундаменталізованого навчання об'єктно-орієнтованому програмуванню майбутніх фахівців галузі інформаційних технологій. *Молодь і ринок*. Дрогобич, 2018. № 4/159. С. 64–70. DOI: <https://doi.org/10.24919/2308-4634.2018.131347>

2. Ковтонюк Г., Бак С., Крупський Я. Об'єктно-орієнтований підхід до створення графічного інтерфейсу користувача в Python із використанням модуля Tkinter. *Математика, інформатика, фізика: наука та освіта*. 2025. № 2(1), С. 169–175. DOI: <https://doi.org/10.31652/3041-1955-2025-02-01-05>

3. Петровський С.С., Пасічник О.А., Срипник Т.К. Особливості використання методу проєктів при викладанні освітніх компонент спеціальності комп'ютерні науки *Вісник Хмельницького національного університету. Технічні науки*. 2021. № 4. С. 30–34. URL: <https://elar.khmnu.edu.ua/handle/123456789/10723>

4. Пузікова А., Лупан І. Застосування навчальних проєктів у навчанні програмування. *Наукові записки. Серія: Проблеми природничо-математичної, технологічної та професійної освіти*. 2024. № 1 (3). С. 71–78. DOI: <https://doi.org/10.32782/cusu-pmtp-2024-1-9>

5. Raibulet, Claudia & Arcelli Fontana, Francesca. Collaborative and Teamwork Software Development in an Undergraduate Software Engineering Course. *Journal of Systems and Software*. 2018. 144 p. DOI: <https://doi.org/10.1016/j.jss.2018.07.010>

6. Ebrahim Adam. An Investigation Into the Perceived Effectiveness of GitHub Repositories to Teach Programming. *International Conference on Education Research* 2024. 1(1), PP. 1–9.

7. Bennedsen Jens, Böttjer Till, Tola Daniella Using Github Classroom in Teaching Programming. *Proceedings of the 18th International CDIO Conference*, 2022, Reykjavik Iceland, June 13–15, PP. 690–702.

8. Ткачук Г. Методика використання різних форм та методів навчання під час вивчення програмування. *Молодь і ринок*. Дрогобич, 2024. № 10/230. С. 45–52. DOI: <https://doi.org/10.24919/2308-4634.2024.314170>

9. Glassey, R. Use of GitHub in teaching programming. *Proceedings of the 15th International CDIO Conference*. 2019 URL: <https://www.cdio.org/sites/default/files/documents/172.pdf>

10. Luchkevych M., Shakleina I., Duda O. The impact of modern cloud technologies on the efficiency of DevOps processes. *Вісник Тернопільського національного технічного університету*. 2025. № 1 (117). С. 112–122. DOI: https://doi.org/10.33108/visnyk_tntu2025.01.112

butnikh fakhivtsivu haluzi informatsiynykh tekhnolohii [Methodology for using different forms and methods of learning when learning programming]. *Youth & market. Monthly Scientific and Pedagogical Journal*. Drohobych. № 4/159. pp. 64–70. DOI: <https://doi.org/10.24919/2308-4634.2018.131347> [in Ukrainian].

2. Kovtoniuk, H., Bak, S. & Krupskiy, Ya. (2025). Obiektno-orientovanyi pidkhid do stvorennia hrafichnoho interfeisu korystuvacha v Python iz vykorystanniam modulia Tkinter [An object-oriented approach to creating a graphical user interface in Python using the Tkinter module]. *Mathematics, computer science, physics: science and education*. No. 2(1), pp. 169–175. DOI: <https://doi.org/10.31652/3041-1955-2025-02-01-05> [in Ukrainian].

3. Petrovskiy, S.S., Pasichnyk, O.A. & Srypnik, T.K. (2021). Osoblyvosti vykorystannia metodu proektiv pry vykladanni osvitykh komponent spetsialnosti kompiuterni nauky [Features of using the project method when teaching educational components of the computer science specialty]. *Bulletin of Khmelnytskyi National University. Technical Sciences*. No. 4. pp. 30–34. Available at: <https://elar.khmnu.edu.ua/handle/123456789/10723> [in Ukrainian].

4. Puzikova, A. & Lupan, I. (2024). Zastosuvannia navchalnykh proektiv u navchanni prohramuvannia [The use of educational projects in teaching programming]. *Scientific notes. Series: Problems of natural science, mathematics, technology and vocational education*. No. 1 (3). pp. 71–78. DOI: <https://doi.org/10.32782/cusu-pmtp-2024-1-9> [in Ukrainian].

5. Raibulet, Claudia & Arcelli Fontana, Francesca. (2018) Collaborative and Teamwork Software Development in an Undergraduate Software Engineering Course. *Journal of Systems and Software*. 144 p. DOI: <https://doi.org/10.1016/j.jss.2018.07.010> [in English].

6. Ebrahim Adam (2024). An Investigation Into the Perceived Effectiveness of GitHub Repositories to Teach Programming. *International Conference on Education Research*. No. 1(1), pp. 1–9. [in English].

7. Bennedsen Jens, Böttjer Till, Tola Daniella (2022). Using Github Classroom in Teaching Programming. *Proceedings of the 18th International CDIO Conference*. Reykjavik Iceland, June 13–15, pp. 690–702. [in English].

8. Tkachuk, H. (2024). Metodyka vykorystannia riznykh form ta metodiv navchannia pid chas vyvchennia prohramuvannia [Methodology for using different forms and methods of learning when learning programming]. *Youth & market*. Drohobych. No. 10/230. pp. 45–52. DOI: <https://doi.org/10.24919/2308-4634.2024.314170> [in Ukrainian].

9. Glassey, R. (2019). Use of GitHub in teaching programming. *Proceedings of the 15th International CDIO Conference*. Available at: <https://www.cdio.org/sites/default/files/documents/172.pdf> [in English].

10. Luchkevych, M., Shakleina, I. & Duda, O. (2025). The impact of modern cloud technologies on the efficiency of DevOps processes. *Bulletin of Ternopil National Technical University*. No. 1 (117). pp. 112–122. DOI: https://doi.org/10.33108/visnyk_tntu2025.01.112 [in English].

REFERENCES

1. Bardus, I. (2018). Metodyka fundamentalizovanoho navchannia obiektno-orientovanomu prohramuvanniu mai-

Стаття надійшла до редакції 22.07.2025

