

Практичні аспекти впровадження Python у навчанні цифрових дисциплін

*Оксана Степанівна Башуцька¹, Шакуров Євген Олексійович²,
Шуста Володимир Семенович³*

Опубліковано	Секція	УДК
31.10.2025	Професійна освіта	004.43:37.02

DOI: <https://doi.org/10.5281/zenodo.17524067>

Анотація. Мета статті - дослідити питання практичного впровадження мови програмування Python у викладанні цифрових дисциплін та довести ефективність її застосування у формуванні цифрових компетенцій студентів.

Наукова новизна. Систематизовано педагогічні підходи до використання Python в освітньому процесі, розроблено класифікацію підходів до навчання на основі аналізу сучасних досліджень та запропоновано інтегровану модель впровадження Python у цифрових дисциплінах.

Результати. Визначено основні переваги використання Python у навчальному процесі, обґрунтовано вибір педагогічних методів, надано практичні рекомендації щодо організації навчальних дій на основі цієї мови програмування.

Висновки. Впровадження Python як навчального засобу для цифрових дисциплін сприяє розвитку алгоритмічного мислення та підвищує мотивацію студентів, а також забезпечує практичну спрямованість навчання.

Ключові слова: цифрова освіта; інноваційні технології; Python; педагогічні інновації, інформатика, викладання інформатики, цифрові компетентності.

Practical aspects of implementing Python in teaching digital disciplines

Annotation. Purpose of the article. To explore the practical details of the integration of the Python programming language in the teaching of digital disciplines and to support the effectiveness of such an application in the formation of digital competencies of the students. The research focuses on modern pedagogical approaches, discusses implementation strategies, and assesses the effectiveness of Python-based teaching in learning outcomes of computer science education.

Scientific novelty. The research systematizes pedagogical approaches to the use of Python in the educational process, develops a classification of teaching methods based on the analysis

¹ Оксана Степанівна Башуцька, кандидат економічних наук, доцент, доцент кафедри економічної кібернетики та інформатики, Західноукраїнський національний університет, ORCID: <https://orcid.org/0000-0002-2445-896X>.

² Шакуров Євген Олексійович, старший викладач кафедри інформатики, Харківський національний педагогічний університет імені Г. С. Сковороди, ORCID: <https://orcid.org/0000-0002-5381-3465>.

³ Шуста Володимир Семенович, кандидат фізико-математичних наук, доцент, кафедра оптики, ДВНЗ Ужгородський національний університет, ORCID: <https://orcid.org/0009-0004-7328-8268>

of existing research works, and proposes an integrated model for the implementation of Python in digital disciplines. The research forms part of the theoretical basis of programming language choice in educational settings and develops a framework for successful Python integration for various digital disciplines. The novelty is that the analysis of the practical implementation strategies is comprehensive, linking theoretical knowledge with practical programming skills.

Results. The study identifies key advantages of Python use in the educational process, substantiating the choice of pedagogical methods and developing practical recommendations on the organization of educational activities using this programming language.

The research shows that the versatility, readability, and wide range of libraries available in Python make it especially suitable for teaching computational thinking, data analysis, and solving algorithmic problems.

Conclusions. The use of Python in teaching digital disciplines helps to develop algorithmic thinking, raises students' motivation, and guarantees the practical turn of learning. Python is an effective tool for the development of computational competencies and the professional preparation of students for activities in areas of technology-intensive professions. Systematic implementation of Python in digitally related disciplines not only helps students to learn more efficiently but also increases student engagement and builds on the programming competence that students need to be effective in modern technology environments.

Keywords: digital education; innovative technologies; Python; pedagogical innovations, computer science, computer science teaching, digital competencies.

Вступ

Постановка проблеми. Сучасний стан розвитку системи освіти супроводжується цифровізацією освітнього процесу та зростає потреба у формуванні цифрової компетентності випускників вищих навчальних закладів. Цифровізація суспільства змушує систему освіти швидко адаптуватися до змін у технологіях та впроваджувати нові методи викладання технічних предметів. Особливої актуальності набуває питання вибору інструментальних засобів навчання програмування, які б забезпечували ефективне формування алгоритмічного мислення, аналітичних здібностей та практичних навичок розробки програмного забезпечення.

Актуальність дослідження зумовлена стрімким розвитком цифрових технологій та зростає потреба у фахівцях з високим рівнем цифрових компетентностей. Сучасна система освіти стикається з необхідністю кардинального оновлення методики викладання інформатики та програмування.

Традиційні методи викладання програмування часто не відповідають вимогам сучасного ринку праці, де особливо цінуються практичні навички роботи з актуальними технологіями. Бувши однією з найбільш використовуваних мов програмування, Python надає широкі можливості для модернізації освітнього процесу в цифрових напрямках, але проблема ефективного впровадження цієї технології в освітні програми недостатньо досліджена. Значення практичної підготовки студентів до використання сучасних програмних продуктів та алгоритмічних рішень робить необхідним розробку універсальних методів організації впровадження Python у навчальний процес цифрових дисциплін.

Дослідження практичних питань впровадження Python у викладання цифрових дисциплін буде спрямоване на систематизацію педагогічного досвіду, розробку методичних рекомендацій та обґрунтування ефективних методів використання цієї мови програмування в освітньому процесі.

Мова програмування Python займає провідне місце серед інструментів, що використовуються для навчання інформатиці, завдяки виразності свого синтаксису, великим бібліотекам та гнучкості використання.

Аналіз останніх досліджень і публікацій. Стаття N. A. Otahanov [1] присвячена класифікації модулів мови програмування Python, яка є основою для розуміння структурної організації навчального матеріалу. Автор систематизує різноманітні модулі Python за функціональним призначенням, що дозволяє оптимізувати процес вибору релевантних інструментів для конкретних навчальних завдань.

H. Chen [2] досліджує стратегії реформування викладання курсу програмування Python на основі технологій штучного інтелекту. Робота демонструє потенціал інтеграції AI-технологій у навчальний процес для персоналізації навчання та підвищення його ефективності.

Прогресивний підхід до викладання Python з орієнтацією на computational thinking представлено у дослідженні H. Ren, L. Yang, L. Jiang, Y. Bai, W. Lu, J. Chang [3]. Автори розробили методику поетапного формування обчислювального мислення через практичне програмування на Python.

S. Wang, C. Deng, J. Zhu [4] досліджують застосування моделі "перевернутого класу" у викладанні програмування Python. Їхнє дослідження показує переваги активного навчання та самостійної роботи студентів над традиційними лекційними форматами.

Реформи викладання Python у контексті нової інженерної освіти розглядаються у роботі Y. Zhu, Y. Guo, H. Lv, T. Zhang, Y. Yu, Y. Liu [5]. Дослідники наголошують на необхідності інтеграції теоретичних знань з практичними навичками інженерного проектування.

X. Zhang, W. Li, G. Wang [6] представляють проектно-орієнтований підхід до викладання статистичних експериментів з використанням Python. Їхня методика демонструє ефективність застосування реальних проектів для засвоєння статистичних концепцій.

Застосування Python у викладанні теорії ймовірностей та математичної статистики досліджено D. Zhang, C. Song [7]. Автори показують, як програмування може полегшити розуміння складних математичних концепцій через візуалізацію та моделювання.

Y. Zhang, M. Zhang, L. Wu, J. Li [8] розробили концептуальну рамку цифрової трансформації вищої освіти з використанням AI-технологій в інженерному навчанні. Їхнє дослідження підкреслює важливість системного підходу до впровадження цифрових технологій.

A. A. Abdulhameed, N. M. Hussien, Y. M. Mohialden, G. Herlem, I. Lajoie, R. Yahiaoui [9] досліджують покращення у сфері кібербезпеки через поєднання Python з цифровою криміналістикою, що розширює сферу застосування мови програмування у спеціалізованих дисциплінах.

Систематичний огляд ролі ІКТ (інформаційно-комунікаційних технологій) в обов'язковій освіті представлено T. Martínez-Soto, P. Prendes [10]. Їхнє дослідження надає широкий контекст для розуміння місця програмування у сучасній освітній системі.

M.-S. Chiu [11] вивчає думки викладачів щодо впровадження афективно-орієнтованого викладання математики у реальних та віртуальних класах, що є релевантним для розуміння психологічних аспектів навчання програмування.

P. S. Oh, G. Lee [12] аналізують виклики, пов'язані з гуманним епістемічним агентством у природничій освіті в контексті появи ChatGPT, що актуально для розуміння майбутнього програмування в освіті.

Всебічний огляд застосування цифрових технологій у викладанні професійної освіти надано X. Jiang, J. Xu, X. Xu [13]. Їхнє систематичне картування літератури виявляє основні тенденції та прогалини у цифровізації освіти.

В. Т.-М. Wong, К. С. Li, М. Liu [14] представляють огляд практик розумної освіти у різних дисциплінах, що допомагає зрозуміти міждисциплінарні підходи до впровадження цифрових технологій.

N. Gromik, D. Litz, B. Liu [15] досліджують модель ТРАСК (Technology, Pedagogy, and Content Knowledge) в австралійському контексті, що надає теоретичну основу для інтеграції технологій у навчальний процес.

Формулювання цілей статті. Метою даної статті є дослідити практичні аспекти інтеграції мови програмування Python у навчальний процес цифрових дисциплін та визначити ефективні педагогічні підходи до формування цифрових компетентностей здобувачів вищої освіти.

Результати

Успішне впровадження Python у цифрове навчання потребує врахування педагогічних та технічних аспектів навчального процесу. Дослідження сучасних методів програмування доводить необхідність систематизації способів застосування Python у різних предметних галузях.

Необхідність систематичного підходу до вивчення модулів Python, який дозволив би здобувачам поступово вивчати тонкощі програмування, підкреслюється N. A. Otahanov [1, с. 8], що, своєю чергою, є необхідністю для успішного подальшого навчання. Ця категоризація потім формує основу для створення навчальних програм, що застосовуються на практиці.

У таблиці 1 представлено різні методи викладання Python у рамках цифрових дисциплін та наведено порівняльний аналіз.

Таблиця 1.

Порівняльний аналіз методик викладання Python у цифрових дисциплінах

Методика	Основні принципи	Переваги	Обмеження	Сфера застосування
Computational Thinking	Поетапне формування алгоритмічного мислення	Розвиток логічного мислення	Потребує тривалого часу	Базові курси програмування
Flipped Classroom	Самостійне вивчення теорії, практика в аудиторії	Активне залучення здобувачів	Залежність від самодисципліни	Проміжні та advanced курси
Project-Based Learning	Навчання через реальні проекти	Практична спрямованість	Складність оцінювання	Спеціалізовані дисципліни
AI-Assisted Teaching	Використання ШІ для персоналізації	Індивідуальний підхід	Технічна складність	Адаптивне навчання

Джерело: розроблено авторами на основі [1-3]

У таблиці 1 показано, що кожна методологія має переваги та слабкі сторони, які слід враховувати під час вибору найкращої методології для певного навчального сценарію.

H. Chen [2, с. 12], показує, що інтеграція штучного інтелекту в освіту на Python відкриває перспективи для персоналізованого навчання, в якому складність завдань

може бути адаптована до потреб окремих учнів. Це особливо важливо для розвитку стійких навичок програмування.

Щоб побачити цілісне уявлення про використання Python в освітньому процесі, на рисунку 1 була створена блок-схема, яка показує співвідношення основних елементів.

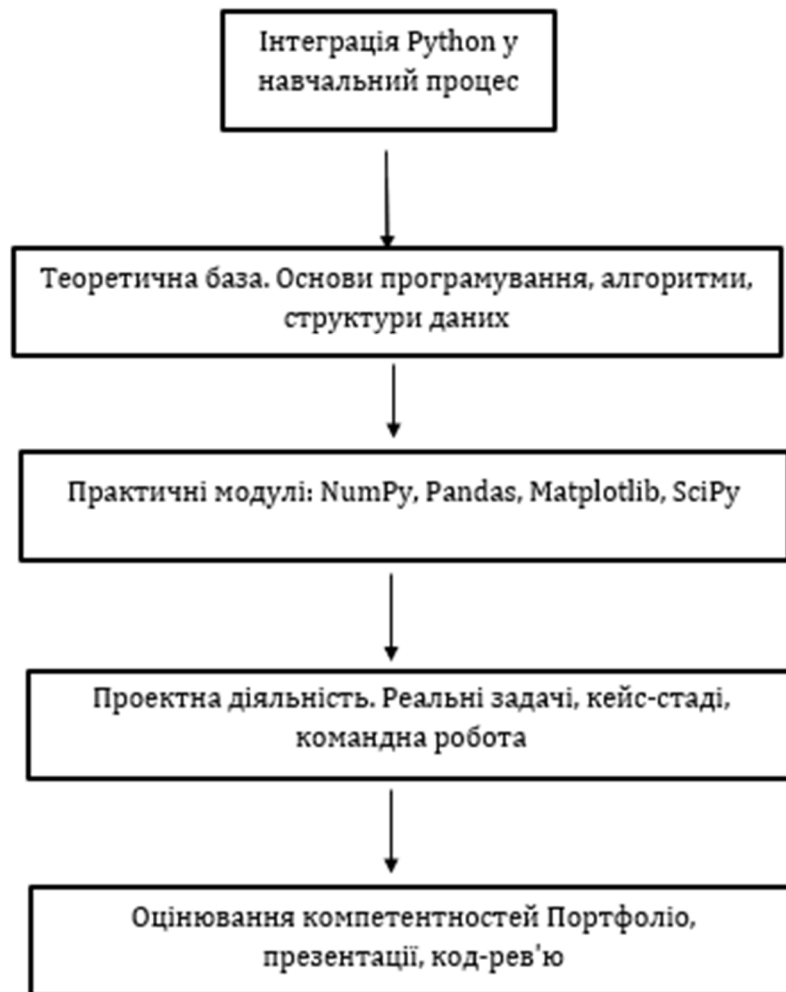


Рисунок 1. Структурна схема інтеграції Python у навчальний процес

Джерело: авторська розробка

Як видно зі структурної схеми на рисунку 1, модель прогресивного навчання демонструє, що вивчення цифрових компетентцій – це покроковий процес, заснований на вивченні Python, у якому кожен рівень доповнює попередній, роблячи навчання систематичним.

Прогресивна модель освіти, яку пропонують H. Ren, L. Yang, L. Jiang, Y. Bai, W. Lu, J. Chang [3, с. 445], підкреслює важливість розвитку computational thinking як основи для ефективного програмування. Цей підхід дозволяє здобувачам не просто засвоювати синтаксис мови, а й розвивати алгоритмічне мислення.

S. Wang, C. Deng, J. Zhu [4, с. 78] показують ефективність моделі "перевернутого класу" у навчанні програмування, де здобувачі самостійно опрацьовують теоретичний матеріал, а аудиторний час присвячується практичним завданням та обговоренню складних питань.

Систематизація модулів на Python продемонстровано на таблиці 2, за предметними областями сприяє оптимізації навчального процесу та дозволяє студентам зосередитися на релевантних інструментах.

Розподіл модулів Python за предметними областями навчання

Предметна область	Основні модулі	Практичне застосування	Рівень складності	Навчальні години
Математика та статистика	NumPy, SciPy, SymPy	Чисельні обчислення, статистичний аналіз	Середній	40
Візуалізація даних	Matplotlib, Seaborn, Plotly	Графіки, діаграми, інтерактивні візуалізації	Базовий	30
Машинне навчання	Scikit-learn, TensorFlow, Keras	Класифікація, регресія, нейронні мережі	Високий	60
Веб-розробка	Django, Flask, FastAPI	Веб-додатки, API, бази даних	Середній	50
Кібербезпека	Cryptography, Scapy, Hashlib	Шифрування, аналіз мережевого трафіку	Високий	45

Джерело: розроблено на основі [4-6]

Таблиця 2 демонструє розподіл навчального навантаження між різними предметними областями, що дозволяє планувати навчальну програму з урахуванням специфіки кожної дисципліни.

Y. Zhu, Y. Guo, H. Lv, T. Zhang, Y. Yu, Y. Liu [5, с. 556] наголошують на необхідності реформування викладання Python у контексті нової інженерної освіти, що передбачає інтеграцію міждисциплінарних проєктів та розвиток інноваційного мислення.

Процес формування цифрових компетентностей через вивчення Python включає декілька послідовних етапів, кожен з яких має специфічні цілі та методи реалізації зображено на рисунку 2.

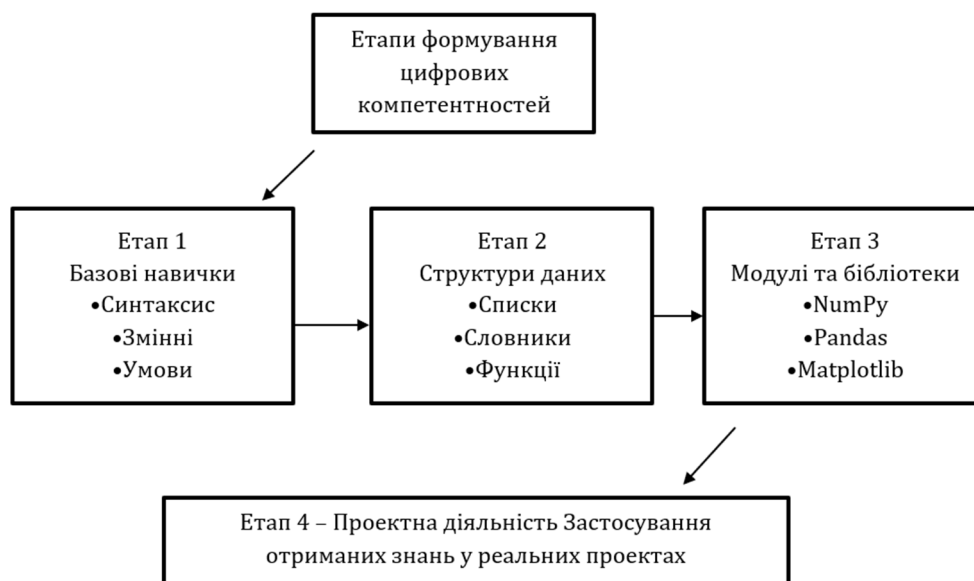


Рисунок 2 – Етапи формування цифрових компетентностей через Python
Джерело: авторська розробка

Рисунок 2 ілюструє поетапний підхід до формування цифрових компетентностей, де кожен етап забезпечує міцну основу для наступного рівня навчання.

X. Zhang, W. Li, G. Wang [6, с. 23] представляють проєктно-орієнтований підхід до викладання статистичних експериментів з використанням Python, що демонструє ефективність практичного застосування програмування для засвоєння складних концепцій.

D. Zhang, C. Song [7, с. 167] демонструють, як Python можна використовувати для візуалізації та моделювання в теорії ймовірностей та математичній статистиці, що робить абстрактні поняття доступнішими для розуміння.

Y. Zhang, M. Zhang, L. Wu, J. Li [8, с. 945] розробили концептуальну основу цифрової трансформації вищої освіти, яка містить системний підхід до впровадження технологій штучного інтелекту в освітній процес. Їхнє дослідження вказує на важливість узгодження технологічних можливостей з педагогічними потребами.)

Для об'єктивної оцінки різних педагогічних підходів проведено аналіз їхньої ефективності за ключовими показниками навчального процесу показано на таблиці 3.

Таблиця 3.

Оцінка ефективності педагогічних підходів до викладання Python

Критерій оцінки	Традиційний підхід	Flipped Classroom	Project-Based	AI-Assisted
Засвоєння матеріалу (%)	65	78	85	88
Мотивація студентів (1-10)	6.2	7.8	8.9	8.5
Практичні навички (1-10)	5.8	7.5	9.2	8.8
Часові витрати викладача	Низькі	Середні	Високі	Середні
Технічні вимоги	Мінімальні	Базові	Стандартні	Базові

Джерело: розроблено на основі [9-12, 15]

Результати таблиці 3 свідчать про значні переваги інноваційних підходів у порівнянні з традиційними методами, особливо у формуванні практичних навичок та підвищенні мотивації студентів.

A. A. Abdulhameed, N. M. Hussien, Y. M. Mohialden, G. Herlem, I. Lajoie, R. Yahiaoui [9, с. 15] розширюють сферу застосування Python у кібербезпеці, показуючи потенціал мови програмування для спеціалізованих дисциплін цифрової криміналістики.

Системний огляд T. Martínez-Soto, P. Prendes [10, с. 8] підтверджує важливість інтеграції ІКТ у навчальний процес, що створює контекст для розуміння місця Python у загальній системі цифрової освіти.

Ефективне впровадження Python у викладанні цифрових дисциплін потребує розуміння складної системи взаємозв'язків між різними компонентами навчального середовища, саме на рисунку 3 зображена система взаємозв'язків між компонентами.



Рисунок 3 – Система взаємозв'язків між компонентами навчального середовища
Джерело: авторська розробка

Рисунок 3 демонструє комплексну взаємодію всіх елементів навчального процесу, де технологічне середовище Python виступає центральним елементом, що поєднує викладачів, студентів, навчальний контент та систему оцінювання.

М.-S. Chiu [11, с. 24660] вивчає психологічні аспекти викладання у віртуальних середовищах, що є актуальним для розуміння особливостей дистанційного навчання програмування та формування позитивного ставлення до технологій.

P. S. Oh, G. Lee [12, с. 785] аналізують виклики, пов'язані з появою штучного інтелекту в освіті, що підкреслює необхідність адаптації методик викладання програмування до нових технологічних реалій.

Для ефективної інтеграції мови програмування Python у викладання цифрових дисциплін необхідно застосовувати системний підхід, що враховує послідовність формування компетентностей від базового рівня до професійного застосування. З цією метою розроблено п'яти етапну схему впровадження рисунку 4, яка охоплює весь цикл освітнього процесу від аналізу потреб та підготовки матеріалів до оцінки результатів й вдосконалення програми. Кожен етап структуровано за ключовими компонентами, що забезпечують логічну прогресію навичок, поступове ускладнення завдань та орієнтацію на практичне використання Python у реальних професійних контекстах. Модель відповідає сучасним вимогам підготовки фахівців у галузі інформаційних технологій та може бути адаптована до різних освітніх програм та рівнів підготовки.



Рисунок 4 – Поетапна модель впровадження Python у викладання цифрових дисциплін

Джерело: авторська розробка

Рисунок ілюструє структуровану модель впровадження мови програмування Python у викладанні цифрових дисциплін, представлену у вигляді п'яти послідовних етапів, оформлених як ієрархічна схема з кольоровим кодуванням. Кожен етап позначений номером та назвою, а також містить перелік ключових компонентів, реалізація яких забезпечує системний підхід до інтеграції Python в освітній процес. Модель охоплює повний цикл — від підготовки до оцінки ефективності, що дозволяє

формувати компетентності студентів у програмуванні, аналізі даних та автоматизації завдань.

Запропонована структура починається з Етапу 1: Підготовчий, який включає аналіз навчальних потреб, розробку програмного забезпечення та підготовку викладацьких матеріалів. Далі йде Етап 2: Впровадження, орієнтований на вивчення базових конструкцій, структур даних та функціонального програмування. Етап 3: Застосування передбачає практичну роботу з аналізом даних, візуалізацією та проектною діяльністю. Етап 4: Поглиблення акцентується на машинному навчанні, веброзробці та автоматизації процесів. Завершальний Етап 5: Оцінювання та вдосконалення забезпечує моніторинг результатів, аналіз ефективності та коригування програми. Така послідовність відповідає принципам поетапного формування професійних навичок і може бути адаптована до різних рівнів підготовки та спеціальностей у сфері цифрових технологій.

X. Jiang, J. Xu, X. Xu [13, с. 16875] надають всебічний огляд застосування цифрових технологій у професійній освіті, що дозволяє позиціонувати Python як ключовий інструмент цифрової трансформації навчального процесу.

B. T.-M. Wong, K. C. Li, M. Liu [14, с. 210] представляють міждисциплінарний підхід до впровадження розумних освітніх технологій, що розширює можливості використання Python у різних предметних областях.

N. Gromik, D. Litz, B. Liu [15, с. 42] досліджують модель TRACK у контексті інтеграції технологій у навчальний процес, що надає теоретичну основу для систематичного впровадження Python у цифрові дисципліни.

Висновки

Проведене дослідження дозволяє зробити висновок про високу ефективність впровадження Python у викладанні цифрових дисциплін за умови системного підходу до організації навчального процесу. Структурна модель інтеграції Python представляє необхідність поетапного формування компетенцій від базових програмних конструкцій до складних прикладних завдань.

Порівняльний аналіз педагогічних методів виявив переваги використання проектного навчання та методів за допомогою штучного інтелекту порівняно з традиційними педагогічними методами. Особливе значення має підвищення мотивації студентів та розвиток практичних навичок програмування із застосуванням реальних кейсів та інтерактивних технологій.

Систематизація модулів Python за предметними галузями забезпечує основу для розробки спеціалізованих навчальних програм, що враховують особливості різних цифрових дисциплін. Розподіл навчального навантаження на теоретичну та практичну частину дозволяє оптимізувати часові ресурси та ефективно засвоїти матеріал.

Система взаємозв'язків між компонентами навчального середовища підкреслює важливість координації між технологічними можливостями, педагогічними методами та індивідуальними потребами студентів. Успішне впровадження Python вимагає не лише технічної підготовки викладачів, але й психологічно стійких фахівців для навчання програмуванню.

Результати дослідження формують основу для подальшого вдосконалення методів викладання цифрових дисциплін та розробки інноваційних освітніх технологій. Перспективними напрямками подальших досліджень є адаптація запропонованих підходів до специфіки різних навчальних закладів, розробка автоматизованої системи оцінювання програмних проєктів, інтеграція технологій віртуальної реальності у навчання програмуванню.

Список використаних джерел

1. Otahanov N. A. Classification of Python Language Modules. *Indonesian Journal of Innovation Studies*. 2022. Vol. 19. DOI: <https://doi.org/10.21070/ijins.v19i.672> URL: <https://is.gd/UHwvAT>
2. Chen H. Research on Teaching Reform Strategies of Python Programming Course Based on Artificial Intelligence Technology. *Journal of Electronic Research and Application*. 2025. Vol. 9, No. 5. DOI: <https://doi.org/10.26689/jera.v9i5.12197> URL: <https://is.gd/212LQE>
3. Ren H., Yang L., Jiang L., Bai Y., Lu W., Chang J. A Computational-thinking-oriented Progressive Teaching Mode for Python Course. *2021 IEEE 3rd International Conference on Computer Science and Educational Informatization (CSEI)*. 2021. DOI: <https://doi.org/10.1109/CSEI51395.2021.9477642> URL: <https://ieeexplore.ieee.org/document/9477642>
4. Wang S., Deng C., Zhu J. Exploration of the Application of Flipped Classroom in Computer Programming Python Language. *Journal of Intelligence and Knowledge Engineering*. 2024. Vol. 2, No. 4. DOI: 10.62517/jike.202404404 URL: <https://is.gd/REtX9H>
5. Zhu Y., Guo Y., Lv H., Zhang T., Yu Y., Liu Y. The Reform and Practice of PYTHON Programming Teaching in the Context of Emerging Engineering Education. *Communications in Computer and Information Science*. 2023. Vol. 1813. P. 552–559. DOI: https://doi.org/10.1007/978-981-99-2449-3_48 URL: https://link.springer.com/chapter/10.1007/978-981-99-2449-3_48
6. Zhang X., Li W., Wang G. Construction and Application of the Project-based Teaching System of Statistical Experiment Course using Python Language. *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*. 2023. DOI: <https://doi.org/10.1109/CSTE59648.2023.00010> URL: <https://is.gd/TgFu6k>
7. Zhang D., Song C. The Application of Python in Teaching Probability Theory and Mathematical Statistics. *2025 International Conference on Intelligent Education and Computer Application (IECA)*. 2025. DOI: <https://doi.org/10.1109/IECA66054.2025.00040> URL: <https://is.gd/PlI49y>
8. Zhang Y., Zhang M., Wu L., Li J. Digital Transition Framework for Higher Education in AI-Assisted Engineering Teaching. *Science & Education*. 2025. Vol. 34. P. 933–954. DOI: <https://doi.org/10.1007/s11191-024-00575-3> URL: <https://link.springer.com/article/10.1007/s11191-024-00575-3>
9. Abdulhameed A. A., Hussien N. M., Mohialden Y. M., Herlem G., Lajoie I., Yahiaoui R. Improvements in Cybersecurity: Coupling Python for Digital Forensics (Version 1). Preprints.org. 2023. DOI: <https://doi.org/10.20944/preprints202307.1666.v1> URL: <https://www.preprints.org/manuscript/202307.1666/v1>
10. Martínez-Soto T., Prendes P. A Systematic Review on the Role of ICT and CLIL in Compulsory Education. *Education Sciences*. 2023. Vol. 13, No. 1. Article 73. DOI: <https://doi.org/10.3390/educsci13010073> URL: <https://www.mdpi.com/2227-7102/13/1/73>
11. Chiu M.-S. Teachers' opinions toward implementing affect-focused mathematics teaching in real and virtual classrooms. *Education and Information Technologies*. 2024. Vol. 29. P. 24653–24676. DOI: <https://doi.org/10.1007/s10639-024-12846-1> URL: <https://link.springer.com/article/10.1007/s10639-024-12846-1>
12. Oh P. S., Lee G. Confronting Imminent Challenges in Humane Epistemic Agency in Science Education: An Interview with ChatGPT. *Science & Education*. 2024. Vol. 34.

- P. 779–805. DOI: <https://doi.org/10.1007/s11191-024-00515-1> URL: <https://link.springer.com/article/10.1007/s11191-024-00515-1>
13. Jiang X., Xu J., Xu X. An overview of domestic and international applications of digital technology in teaching in vocational education: Systematic literature mapping. *Education and Information Technologies*. 2024. Vol. 29. P. 16867–16899. DOI: <https://doi.org/10.1007/s10639-024-12528-y> URL: <https://link.springer.com/article/10.1007/s10639-024-12528-y>
14. Wong B. T.-M., Li K. C., Liu M. A Review of Smart Education Practices Across Disciplines. *Communications in Computer and Information Science*. 2024. Vol. 1974. P. 208–217. DOI: https://doi.org/10.1007/978-981-99-8255-4_18 URL: https://link.springer.com/chapter/10.1007/978-981-99-8255-4_18
15. Gromik N., Litz D., Liu B. Technology, Pedagogy, and Content Knowledge: An Australian Case Study. *Education Sciences*. 2024. Vol. 14, No. 1. Article 37. DOI: <https://doi.org/10.3390/educsci14010037> URL: <https://www.mdpi.com/2227-7102/14/1/37>