УДК 37.091.3:[5+004.42]
DOI: https://doi.org/10.54662/veresen.3.2025.04

Ганна Погромська,
ORCID iD 0000-0002-6779-3995
кандидат педагогічних наук, доцент,
доцент кафедри теорії й методики
природничо-математичної освіти
та інформаційних технологій
Миколаївський обласний інститут
післядипломної педагогічної освіти
вул. Адміральська, 4-а, 54001, м. Миколаїв, Україна
hanna.pohromska@moippo.mk.ua

Наталя Махровська,
ORCID iD 0000-0001-9603-6902
кандидат фізико-математичних наук,
доцент кафедри теорії й методики
природничо-математичної освіти
та інформаційних технологій
Миколаївський обласний інститут
післядипломної педагогічної освіти
вул. Адміральська, 4-а, 54001, м. Миколаїв, Україна
natalya.makhrovska@moippo.mk.ua

СИНЕРГІЯ ПРОГРАМУВАННЯ ТА ПРИРОДНИЧО-МАТЕМАТИЧНИХ ДИСЦИПЛІН У КОНТЕКСТІ STEM-ОСВІТИ

У статті досліджено можливості синергії програмування та природничо-математичної освіти у школі. Показано роль мови Python як ефективного засобу міжпредметної інтеграції в контексті STEM-освіти. На основі аналізу наукових джерел і практичного досвіду авторів висвітлено потенціал Python для розвитку обчислювального мислення учнів, моделювання, аналізу даних та візуалізації природничих процесів. Окреслено виклики інтеграції програмування в шкільну освіту та запропоновано рішення. Набули розвитку методичні розв'язки інтеграції Python у STEM-предмети, зокрема в занятті «Кодимо розумно: алгоритмічні структури у Python. Структура повторення: цикл for», приклади завдань, інструментів і видів діяльності, що сприяють розвиткові обчислювального мислення учнів. Запропоновано низку практичних рішень: упровадження курсу для вчителів «Старт у світ кодування: Python для початківців», варіанти інтеграції програмування у STEM-дисципліни.

Ключові слова*: обчислювальне мислення; програмування; синергія; Python; STEM.*

Вступ. Сучасна освіта перебуває в активному процесі трансформації, де ключову роль відіграє інтеграція цифрових технологій, зокрема програмування. У цьому контексті особливої актуальності набуває STEM-освіта. Вона є ключовим елементом підготовки учнів до сучасного технологічного світу. STEM-освіта сприяє розвиткові критичного мислення, аналітичних навичок і здатності розв'язувати складні проблеми. Упродовж останніх років навчання програмування дедалі більше розглядають

не лише як технічну навичку, а як базовий елемент формування STEM-компетентностей – сукупності знань, умінь і ставлень, що лежать в основі науки, технологій, інженерії та математики. Зокрема його інтеграція в освітній процес дає змогу системно розвивати алгоритмічне, комп'ютингове мислення, які є складниками ключових компетентностей XXI століття.

За останні роки Python закріпився як універсальний інструмент для навчання програмування через свою простоту, синтаксис та широкі можливості. Дослідження, зокрема від MIT (Massachusetts Institute of Technology) (MIT OpenCourseWare) та UNESCO (Institute for Information Technologies in Education), зазначають ефективність використання Python у навчанні не лише інформатики, а й природничих наук, математики, екології та інженерії. Публікації у виданнях «Computers & Education», «International Journal of STEM Education» та українських педагогічних журналах підтверджують, що Python відкриває нові можливості для реалізації міждисциплінарних STEM-проєктів уже на рівні шкільної освіти. Актуальність теми зумовлена зростанням попиту на фахівців із навичками програмування та необхідністю підготовки учнів до цифрової економіки.

Аналіз останніх досліджень і публікацій. Огляд сучасних досліджень свідчить про зростання інтересу до теми синергії програмування та STEM-освіти. Низка досліджень обґрунтовують зв'язок між вивченням програмування і розвитком когнітивних навичок, таких, як: декомпозиція, абстрагування, моделювання та критичне оцінювання рішень (Grover S., Pea R., 2013; Liu Z., Gearty Z., Richard E., 2024). Уперше формулювання поняття «обчислювальне мислення» як стилю мислення, притаманного комп'ютерним наукам, але придатного для розв'язання задач у будь-якій сфері, розкрито у праці Джинетт Вінг (Wing J., 2006). У дослідженні (Cheng L., Wang X. Ritzhaupt A., 2023) зазначено, що інтеграція такого мислення у викладання предметів STEM у початковій та середній школі має значний позитивний уплив на успішність учнів у навчанні STEM.

Низка досліджень доводить значення програмування в STEM-освіті. Огляд (Tariq R., Aponte Babines B. M., Ramirez J., Alvarez-Icaza I., Naseer F., 2024) підтверджує, що програмування в контексті STEM сприяє розвитку не лише технічних, а й метапізнавальних здібностей учнів, а також підвищує зацікавленість у вивченні природничо-математичних дисциплін. Особливе місце в цих дослідженнях посідає мова Python, яку дослідники визначають як зручну для шкільного навчання завдяки її інтуїтивному синтаксису, гнучкості й міждисциплінарному потенціалу (Yang D., Baek Y., Ching Y., Swanson S., Chittoori B., Wang S., 2021).

Вивчення мови Python сприяє інтеграції обчислювального мислення в математику та природничі науки і дозволяє учням моделювати складні явища (Sengupta P., Dickes A., Farris A. V., 2021). Робота (Meechai L., Choochat S., 2024) показує, що використання Python у навчанні інших предметів сприяє розвитку обчислювального мислення та покращує розуміння природничих і математичних дисциплін. Водночас у звіті OECD (2023) (PISA 2022) зазначено, що країни, які інтегрують програмування в шкільні програми, показують кращі результати в міжнародних тестуваннях PISA з математики та природничих наук.

В українській освітній практиці Python поступово посідає чільне місце серед мов програмування, рекомендованих для шкільного навчання. Це зумовлено його доступністю, простим синтаксисом та відповідністю сучасним освітнім тенденціям. У багатьох закладах освіти Python уже впроваджується в межах базового та профільного курсів інформатики, а також факультативних занять і STEM-гуртків.

Водночас, як зазначають автори Т. П. Грищенко, М. С. Львов, О. В. Співаковський, І. В. Тарасов, інтеграція Python у зміст шкільної освіти досі залишається фрагментарною, переважно обмеженою рамками інформатики, без належної міжпредметної координації. Це свідчить про потребу в розробленні методичних мо-

делей та навчально-практичних кейсів, у яких ефективно поєднується програмування з предметами природничо-математичного циклу.

Постановка проблеми. Попри визнану роль цифрових технологій у сучасній освіті, у школах програмування та природничо-математичні дисципліни здебільшого залишаються ізольованими напрямами навчання. Брак інтеграції між ними обмежує формування в учнів цілісного наукового мислення, міждисциплінарної грамотності та навичок XXI століття, необхідних для розв'язання реальних STEM-проблем. Крім того, значна частина вчителів інформатики не має інструментів або методичної підтримки для впровадження міждисциплінарного підходу до викладання програмування.

Метою статті є дослідження можливостей програмування як інтегративного компонента STEM-освіти, зокрема на прикладі мови Python, та представлення методичних рішень щодо підготовки вчителів до такої інтеграції.

Відповідно до мети окреслено **завдання**:

1. Розкрити можливості синергії програмування з предметами природничо-математичного циклу.

2. Запропонувати методичні рішення щодо реалізації дидактичного потенціалу Python для викладання предметів природничо-математичного циклу в контексті розвитку обчислювального мислення.

3. Визначити види навчальної діяльності, що виникають у результаті синергії програмування та STEM-дисциплін.

Основна частина. У сучасній освіті обчислювальне мислення визнають як одну з найважливіших навичок, необхідних для технологічно інтегрованого майбутнього. Воно є провідним умінням для процесів розв'язання проблем, що постійно розвиваються разом з інноваційним прогресом. Зокрема, у STEM-освіті обчислювальне мислення має вагоме значення, оскільки його субкомпетенції охоплюють абстракцію, декомпозицію, розпізнавання шаблонів та алгоритмічний дизайн. Окрему роботу (Yang D., Baek Y., Ching Y., Swanson S., Chittoori B., Wang S., 2021) присвячено саме розробленню й упровадженню інтегрованої навчальної програми STEM і формуванню обчислювального мислення, основаного на проєктно-орієнтованому навчанні, що автори розвідки визначають як фундаментальну навичку XXI століття та «третій стовп» наукової практики. Підкреслено, що його інтеграція у початкову та середню освіту з акцентом на STEM-навчанні має потенціал для покращення науковості змісту та підвищення зацікавленості учнів у STEM. Однак попри визнання обчислювального мислення як визначальної наукової практики та підтримку його викладання в середній школі воно залишається майже не представленим у контексті STEM-освіти.

Наразі актуальним є питання не про те, «чому» варто інтегрувати обчислювальне мислення у шкільні предмети, а «як» це зробити. Адже ключові компетентності, що супроводжують обчислювальне мислення, є значущими для STEM-навчання, що зумовлено їх зв'язком із дисциплінарними процесами STEM, такими, як моделювання, міркування, критичне мислення та розв'язання проблем.

У контексті STEM-освіти програмування розглядають як інструмент міжпредметної інтеграції, оскільки воно сприяє об'єднанню знань із математики, фізики, хімії, біології та технологій у цілісне навчальне середовище. Мова Python завдяки простоті, інтуїтивно зрозумілому синтаксисові та наявності численних освітніх бібліотек є надзвичайно зручною для реалізації такої синергії вже на рівні загальної середньої освіти.

У сучасних українських школах мову Python поступово інтегрують у курс інформатики, особливо в 7–11 класах, як на рівні базового, так і профільного навчання (Кобильник Т. П., 2021). Згідно з дослідженням (Кобильник Т. П., 2022) викладання починають із 7-го класу, а у старшій школі нею послуговуються у вибіркових модулях «Креативне програмування» та в курсі «Мова програмування та структури даних» на профільному рівні. Неперервність навчання основ алгоритмізації та

програмування наразі не забезпечено, що є проблемою. Одним із можливих рішень, на нашу думку, є синергія програмування зі STEM-дисциплінами.

Інтеграція Python у навчальні програми є фрагментарною і здебільшого обмеженою інформатикою, без системного зв'язку з іншими природничими предметами. Наприклад, проєктом «Оновлена інформатика – ІТ-студії» (між Міністерством освіти та Мінцифри у рамках EU4DigitalUA) передбачено вивчення мови Python у навчальній програмі 8 класу, але лише в тих школах, які ще проходять адаптацію до поширення (Сабашина Ю., 2023). У позакласних і факультативних активностях учителі застосовують Python для інтеграції з математикою: під час вивчення функцій доречним є паралельний розгляд операторів (розгалуження, вибору, циклів) із використанням або без онлайн-платформ тощо (Ракута В. М., 2024).

Звернімо увагу, що ефективність упровадження програмування в STEM-курси значною мірою залежить від підготовки педагогів. Різнопланове застосування програмування у STEM-проєктах потребує наявність підготовлених фахівців. Останнім часом таку ситуацію ускладнено нестачею вчителів, які готові і можуть викладати в розрізі сучасних мов. Як наслідок, спостерігаємо випадки, коли педагоги працюють за старими методиками. Наприклад, Т. Херт, Е. Грінвальд, С. Аллан, (Hurt T., Greenwald E., Allan S., 2023) у рамках розробки CT-S framework для шкільної освіти акцентують на потребі цілеспрямованої роботи з учителями, що мають розуміти не лише алгоритми, але й педагогічну логіку їх упровадження в навчання.

Кейс інтеграції Python у викладання фізики, де ключовим бар'єром стала не технічна складність мови, а запит щодо зміни підходів учителів до організації навчання та їхнє розуміння програмування як інструменту для наукового дослідження, описано в розвідці (Lane D., Galanti T., Rozas X. L., 2023) авторів Д. Лейна, Т. Галанті, Х. Розаса. Систему підвищення кваліфікації вчителів для впровадження міждисциплінарного та трансдисциплінарного змісту і реалізації принципів STEM-освіти в навчальному процесі описує О. В. Ліскович (Ліскович О. В., 2023). Авторка наголошує на практичній спрямованості освітнього процесу та надання педагогам можливості вибору курсів залежно від їхніх освітніх потреб. STEM-підхід у пропонованій системі належить розглядати як засіб розвитку інформаційно-цифрової компетентності та комплексного професійного зростання педагогів.

Упровадження Python у шкільну освіту (не лише на уроках інформатики) потребує адаптації навчальних програм. В Україні пілотні проєкти з інтеграції програмування в школах (наприклад, у рамках ініціативи Code Club Ukraine) показують, що учні 7–9 класів здатні опанувати базові навички Python за 3–4 місяці. Однак брак кваліфікованих кадрів та обмежений доступ до технічних засобів гальмують процес. Окреслимо низку викликів, що постають перед учителями, та запропонуємо можливі рішення (див. таблицю 1).

Таблиця 1.

Основні виклики та пропоновані рішення інтеграції Python у шкільну STEM-освіту

	Проблема / виклик	*Пропоноване рішення*
1.	Брак підготовлених учителів	Проведення тренінгів і онлайн-курсів
2.	Недостатня адаптація програм	Формування системи прикладів інтеграції програмування у STEM-дисципліни. Створення відкритих освітніх ресурсів з інтегрованими кейсами

3.	Нестача ефективної міжпредметної інтеграції	Організація спільного планування між учителями різних предметів. Розроблення міждисциплінарних STEM-уроків / занять. Розроблення методичних матеріалів для вчителів
4.	Обмежений доступ до техніки	Залучення хмарних платформ (наприклад, Google Colab), мобільних комп'ютерних класів

Джерело: авторський варіант

Однак, як свідчать спостереження освітянської спільноти, багато вчителів природничо-математичних дисциплін стикаються з кількома типовими труднощами:

- обмеженість досвіду у сфері програмування;
- брак готових методичних рішень для інтеграції тем (наприклад, «функція» в математиці та графік у Python);
- нерозуміння, як реалізувати міжпредметну співпрацю;
- недостатність ресурсів або мотивації.

У відповідь на ці виклики доцільним є створення навчальних програм для вчителів, що містять:

- поетапне опанування основ Python;
- ознайомлення з освітніми бібліотеками (matplotlib, pandas, sympy, random);
- приклади інтегрованих завдань і сценаріїв уроків;
- прототипи розроблення власного навчального контенту.

Як варіант подолання викликів 1 і 2 (див. таблиця 1) автори розробили курс «Старт у світ кодування: Python для початківців». Метою курсу є вдосконалення предметно-методичної та інформаційно-цифрової компетентностей учителів інформатики, природничої та математичної освітніх галузей. Розроблений курс (покликання на програму: https://cutt.ly/4e8pSdoa) створено в рамках сертифікованого заходу Миколаївського ОІППО та розраховано на 30 акад. год (1,5 кредиту ЄКТS). (Махровська Н. А., Погромська Г. С., 2024).

У рамках рішення 1 (див. таблиця 1) автори провели STEM-заняття на тему «Кодимо розумно: алгоритмічні структури у Python. Структура повторення: цикл for» і присвячено «STEM-тиждень МОІППО–2025». Під час заняття акцентовано на ідеї розумного програмування, за основу якого взято концепцію ефективності, повторного використання коду, адаптивності та логічності (оптимізація, автоматизація, адаптація) (рис. 1). Зазначені положення розглянуто в розрізі узагальнення базових алгоритмічних конструкцій, зокрема докладного розбору конструкції for. Учасники упевнились, що цикл for під час виконання практичних завдань може перетворитись на інструмент для аналізу, дослідження та нових відкриттів (табл. 2).

Рис. 1. Розумне програмування

Джерело: авторський варіант

Таблиця 2.

Цикл for у STEM

Сценарій циклу	STEM-приклад	Коментар
Підрахунок кількості	Підрахунок кількості учнів, які виконали певну умову в олімпіаді	*Math* + *Programming*: count += 1, умовні перевірки, статистика
Підрахунок суми / добутку	Розрахунок загальної ваги деталей у конструкції або сумарного балу на тесті	*Math*: арифметичні операції. *Engineering*: аналіз ресурсів
Обмін значень змінних	Перевірка коректності даних у таблиці – переставити значення місцями	*Algorithmic thinking* + *Programming*: розуміння пам’яті
Сигнальні мітки	Переривання обчислень при досягненні умови: «як тільки знайшли мінімум – стоп»	*Engineering*: оптимізація процесів. *Logic*: early stopping
Пошук максимуму / мінімуму	Знайти найкращий результат на конкурсі або найвищу температуру	*Data Science* + *Math*: базова аналітика, алгоритм
Розширені оператори (range(step), enumerate, zip)	Створення звітів, парне оброблення значень, контроль індексу	*Technology*: чистий, ефективний код. *Engineering*: структурування

Джерело: авторський варіант

Цей досвід показав, що навіть базові знання з Python надають змогу вчителям природничо-математичного циклу розширити дидактичний інструментарій і переосмислити роль програмування як інструменту пізнання, а не лише технічної навички. Ефективна підготовка педагогів передбачає перехід від «навчання Python» до «використання Python у власному предметі». У цьому полягає сутність міждисциплінарного STEM-підходу, де програмування не самоціль, а засіб формування нової якості освіти.

Автори виокремлюють три можливі підходи до інтеграції програмування в освіту:

1. Вивчення програмування без суттєвого змісту предметів.
2. Інтеграція, що підтримує вивчення предметного змісту для опису та тестування систем.
3. Інтеграція, що відповідає практиці STEM.

У площині дослідження пропонуємо розглядати інтеграцію за третім підходом. Адже ідея STEM-освіти полягає у викладанні природничих наук, технологій, інженерії та математики через інтегровані практичні завдання, моделювання, експерименти та міжпредметні зв’язки. Одним

зі стрижневих інструментів реалізації такого підходу є програмування, зокрема мовою Python, яка забезпечує доступний і потужний функціонал для створення цифрових моделей, оброблення даних і візуалізації результатів. Є значний потенціал для міжпредметної інтеграції програмування та STEM-дисциплін у викладання предметів природничо-математичного циклу в середній школі. Розглянемо можливі варіанти такої синергії.

Зв'язок програмування і математичної освіти простежуємо саме на рівні розвитку в учнів обчислювального мислення, що формується через програмування і є основою для розв'язання математичних задач. Воно базується на принципах інформатики та містить здатність формулювати задачі таким чином, щоб їх можна було ефективно розв'язати за допомогою обчислювальних методів, алгоритмів та цифрових технологій (Wing J. M., 2006). Python дає змогу здобувачам освіти візуалізувати абстрактні поняття, такі, як функції, графіки чи статистичні дані. Наприклад, бібліотека Matplotlib дає змогу будувати графіки функцій, що полегшує розуміння понять з курсу алгебри. Учні можуть створювати програми для розв'язання систем рівнянь або моделювання геометричних фігур, що робить математику більш інтерактивною.

Дослідження Є Хуа, Лянг Б., Нг Ол. (Ye H., Liang B., Ng OL., 2023) показує, що використання програмування Python у математичній освіті сприяє розвитку обчислювального мислення, зокрема через створення симуляторів, наприклад моделі поширення епідемій. Учні можуть згенерувати код для обчислення коренів квадратного рівняння та побудови параболи, що закріплює знання з алгебри. За даними Д. Вейнтроп, Е. Бегешті, М. Хорн (Weintrop D., Beheshti E., Horn M., 2020), такі підходи підвищують мотивацію учнів до вивчення математики на 25 %. У природничих науках програмування застосовують для моделювання складних процесів.

Наприклад, на основі бібліотеки *Pygame* можна створювати симуляції руху тіл, що допомагає учням зрозуміти закони Ньютона. У хімії можливості Python застосовують для аналізу хімічних реакцій або обчислення молярних мас за допомогою бібліотеки ChemPy. Дослідження (Lohakan M., Seetao C., 2024) описує експеримент, у якому 149 старшокласників використовували Python у складі набору для штучного інтелекту, щоб вивчати комп'ютерний зір і програмування. Результати підтвердили значне покращення рівня знань (з 11.15 до 18.67 балів у тестах). За даними NGSS (Next Generation Science Standart, 2022) міждисциплінарні проєкти підвищують рівень розуміння природничих наук на 30 % порівняно з традиційними методами. Python дає змогу реалізовувати прикладні навчальні ситуації, що поєднують елементи математики, фізики, хімії та біології (див. таблиця 3).

Таблиця 3.

Приклади інтеграції програмування у STEM-дисципліни

Предмет	**Розділ**	**Приклад**	**Інструмент**	**Застосування**
Математика	Алгебра	Розв'язання систем лінійних рівнянь	Бібліотека *sympy*	Використання *sympy.solve* для знаходження розв'язків системи
	Геометрія	Обчислення відстані між точками у 3D	Бібліотека *numpy*	Використання *numpy.linalg.norm* для обчислення евклідової відстані між точками
	Статистика	Аналіз даних та побудова гістограм	Бібліотеки *pandas*, *matplotlib*	Оброблення набору даних (наприклад, оцінки учнів) та візуалізація розподілу

Фізика	Механіка	Моделювання руху тіла під дією сили	Бібліотеки *numpy*, *matplotlib*	Розраховування траєкторії снаряда з урахуванням гравітації та візуалізація руху
	Електрика	Аналіз електричних кіл	*Raspberry* Pi, *micro*:bit, *Arduino* датчики струму, бібліотека *RPi*. *GPIO*	Збирання даних з датчиків для аналізу закону Ома через Python
Хімія	Хімічні реакції	Балансування хімічних рівнянь	Бібліотека *chempy*	Використання *chempy*. *balance_stoichiometry* для автоматичного балансування реакцій
	Аналітична хімія	Оброблення даних титрування	Бібліотеки *pandas*, *scipy*	Аналіз даних експерименту та побудова кривих титрування для визначення концентрації
Біологія	Генетика	Аналіз послідовностей ДНК	Бібліотека *biopython*	Використання *Bio*.*Seq* для аналізу послідовностей ДНК: підрахунок нуклеотидів
	Екологія	Моделювання популяційної динаміки	Бібліотеки *numpy*, *matplotlib*	Побудова моделей зростання популяції (наприклад, логістична модель) та їх візуалізація
Астрономія	Орбітальна механіка	Моделювання орбіти планети	Бібліотеки *numpy*, *matplotlib*	Розрахунок еліптичної орбіти за законами Кеплера та візуалізація траєкторії
	Астрофотографія	Оброблення зображень із телескопа	Бібліотека *astropy*	Використання *astropy* для аналізу зоряних зображень та визначення яскравості зірок

Джерело: авторський варіант

Окрім того, окреслимо Інструменти для навчання STEM з елементами програмування:

- Scratch, Code.org, Tynker (програмування + математика + фізика).
- Arduino, micro:bit (програмування + фізика + електроніка).
- Python, JavaScript, Minecraft Education (для STEM-занять із програмуванням).
- Desmos API, GeoGebra Scripts (математика + інформатика).

Пропонуємо звернути увагу на застосування видів діяльності, притаманних синергії STEM-заняття і програмування:

1. Аналіз даних та моделювання у процесі роботи над природничо-математичними проєктами. Учні можуть використовувати програмування для збирання, оброблення й аналізування даних; створювати або використовувати обчислювальні моделі та симуляції для розуміння концепцій або явищ. Наприклад, програмування може сприяти розширенню можливостей використання статистики в аналізі даних або використання ітеративного чисельного інтегрування для дослідження проблем у

фізиці, які інакше потребували б аналітичного числення.

2. Виконання проєктів (переважна кількість завдань STEM). Такий підхід сприятиме забезпеченню контексту для інтеграції обчислювального мислення з кількома STEM-дисциплінами. Учні можуть залучати алгоритмізацію для розв'язання реальних проблем. Наприклад, проєктування та програмування роботів для дослідження навколишнього середовища або моделювання та тестування інженерних конструкцій під навантаженням.
3. Розроблення та оцінювання обчислювальних інструментів. Автори пропонують цей вид діяльності для поглибленого вивчення засад програмування. Але навіть без написання коду з нуля учні можуть розвивати обчислювальне мислення у природничих науках за допомогою рефлексивного використання, проєктування або оцінювання функціональності обчислювальних інструментів (наприклад, графічні калькулятори, симуляції або навіть уявні інструменти) для досягнення наукових цілей з прив'язкою до власних специфічних питань.
4. Інструменти та середовища. Для інтеграції можна використовувати різноманітні інструменти – від візуальних мов програмування, зокрема Scratch, до текстових, як Python. Робототехніка (наприклад, Arduino, micro:bit) та діяльність без комп'ютерів також є ефективними способами розвитку обчислювального мислення. Середовища, що поєднують текст, зображення та програмний код, є зручними для створення навчальних матеріалів щодо інтегрування обчислення в предметний контекст.

Зазначене демонструє, як програмування може стати сполучною ланкою між математикою, аналізом даних, моделюванням реальних природничих процесів тощо і слугує потужним прикладом для розроблення шкільних STEM-програм.

Висновки та перспективи подальших досліджень. Програмування, зокрема мовою Python, є провідним інтегративним компонентом STEM-освіти для розвитку критичного мислення, аналітичних навичок та здатності розв'язувати складні проблеми. Python, завдяки своїй простоті та широким можливостям, відкриває нові можливості для міждисциплінарних STEM-проєктів на шкільному рівні та надає змогу візуалізувати абстрактні математичні поняття і моделювати природничі процеси. Інтеграція програмування позитивно впливає на успішність учнів у STEM та підвищує їхню зацікавленість у вивченні природничо-математичних дисциплін, що підтверджується зокрема кращими результатами країн у PISA-тестуваннях.

Запропоновані методичні рішення щодо реалізації дидактичного потенціалу Python слугуватимуть ефективному впровадженню міждисциплінарного підходу до викладання програмування в умовах шкільної STEM-освіти. Вони сприятимуть не лише формуванню в учнів базових навичок кодування, а й розвиткові обчислювального мислення, аналітичних здібностей та здатності до розв'язання реальних проблем через інтеграцію знань із математики, фізики, хімії, біології та інформатики. Пропоновані підходи – від використання бібліотек Python для візуалізації математичних понять до створення моделей природних явищ – зумовлюватимуть осмислення складних концепцій через практичну діяльність, експериментування та проєктну роботу учнів. Такі рішення формують основу для побудови міждисциплінарного навчального середовища, де програмування виступає не метою, а засобом пізнання, аналізу і моделювання реальних процесів, що відповідає сучасним освітнім викликам і запитам STEM-підходу.

Проведений аналіз сучасного стану впровадження Python у шкільну освіту в Україні засвідчив наявність потенціалу для синергії програмування та природничо-математичних дисциплін, що має бути

реалізований шляхом підготовки вчителів, адаптації навчальних програм і впровадження інноваційних освітніх практик. Запропонований авторський досвід створення курсу для педагогів «Старт у світ кодування: Python для початківців» та розробка STEM-заняття підтверджують реалістичність і дієвість такого підходу.

Представлені види навчальної діяльності, що виникають у результаті синергії програмування та STEM-дисциплін, демонструють широкий дидактичний потенціал програмування як інструменту міжпредметної інтеграції. Аналіз даних, моделювання, проєктна діяльність, розроблення та оцінювання обчислювальних інструментів сприяють формуванню в учнів відповідного однойменного мислення, критичного аналізу та здатності працювати з практичними задачами. Використання Python та інших цифрових середовищ забезпечує створення гнучкого освітнього простору, у якому учні навчаються досліджувати, експериментувати й створювати власні рішення, що відповідає вимогам сучасної STEM-освіти.

Перспективи досліджень пов'язуємо зі створенням інтегрованих навчальних програм STEM та Python для різних вікових груп та вивченням упливу реалізації міждисциплінарних Python-проєктів на розвиток soft skills.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кобильник Т. П. та інші. Методичні аспекти вивчення основ алгоритмізації і програмування мовою Python у шкільному курсі інформатики у старших класах / Т. П. Кобильник, У. П. Когут, В. Б. Жидик // Фізико-математична освіта. – 2021. – № 5(31). – С. 36–44.

2. Кобильник Т. П. та інші Python як засіб навчання основ алгоритмізації у закладах загальної середньої освіти / Т. П. Кобильник, О. В. Сікора, В. Б. Жидик, О. В. Шаран // Інформаційні технології і засоби навчання. – 2022. – Том 89. – № 3 – С. 16–32. DOI: 10.33407/itlt.v89i3.4896

3. Ліскович О. В. Використання можливостей STEM-освіти у процесі підвищення кваліфікації вчителів фізики та астрономії / О. В. Ліскович // Вересень. – 2023. – Том 2. – № (97). – С. 40–49.

4. Махровська Н. А., Погромська Г. С. Практико-технологічні аспекти впровадження алгоритмізації та програмування в освітній процес / Н. А. Махровська, Г. С. Погромська // Вересень № 1 (104) 2025. – С. 13–24. DOI: https://doi.org/10.54662/veresen.1.2025.02

5. Ракута В. М. Python у шкільному курсі інформатики. Основи програмування: навчальний посібник / В. М. Ракута. – Чернігів, 2024. – 160 с.

6. Сабашина Ю. Python у восьмому класі. Коли оновлять шкільний курс інформатики та які результати пілотного проєкту / Ю. Сабашина. 2023. [Електронний ресурс]. – Режим доступу: https://dou.ua/lenta/news/computer-science-in-schools/?utm_source=chatgpt.com

7. Cheng L., Wang X. Ritzhaupt A. View all authors and affiliations // Journal of Educational Computing Research. – 2023. – № 61 (2). – P. 416–443. DOI: https://doi.org/10.1177/07356331221114183

8. Grover S., Pea R. Computational thinking in K–12: A review of the state of the field // Educational Researcher. – 2013. – № 42(1). – P. 38–43. DOI: https://doi.org/10.3102/0013189X12463051;

9. Hurt T., Greenwald E., Allan S. The computational thinking for science (CT-S) framework: operationalizing CT-S for K–12 science education researchers and educators // Journal of Science Education and Technology. – 2023. – 10. – P. 1–16. DOI: https://doi.org/10.1186/s40594-022-00391-7

10. Lane D., Galanti T., Rozas X.L. Teacher Re-novicing on the Path to Integrating Computational Thinking in High School Physics Instruction // Journal for STEM Education Research. – 2023. – № 6. – P. 302–325. DOI: https://doi.org/10.1007/s41979-023-00100-1

11. Liu Z., Gearty Z., Richard E. Bringing computational thinking into classrooms: a systematic review on supporting teachers in integrating computational thinking into K-12 classrooms // International Journal of STEM Education. – 2024. – № 11(1). DOI: https://doi.org/10.1186/s40594-024-00510-6

12. Lohakan M., Seetao C. Large-scale experiment in STEM education for high school students using artificial intelligence kit based on computer vision and Python // Heliyon . – 2024. – Volume 10. – Issue 10. DOI: https://doi.org/ 10.1016/j.heliyon.2024.e31366

13. Meechai L., Choochat S. Large-scale experiment in STEM education for high school students using artificial intelligence kit based on computer vision and Python // Heliyon. – 2024. – № 10. [Електронний ресурс]. – Режим доступу: https://pmc.ncbi.nlm.nih.gov/articles/PMC11129092/

14. MIT OpenCourseWare / Massachusetts Institute of Technology [Електронний ресурс]. – Режим доступу: https://ocw.mit.edu/

15. Next Generation Science Standart / Офіційний сайт [Електронний ресурс]. – Режим доступу: https://www.nextgenscience.org

16. PISA 2022 / Results (Volume I): The State of Education Post-COVID-19 // https://www.oecd.org/en/publications/pisa-2022-results-volume-i_53f23881-en.html

17. Sengupta, P., Dickes, A., Farris, A. V. Integrating computational thinking in K-12 science and math education // The MIT Press. – 2021. [Електронний ресурс]. – Режим доступу: https://direct.mit.edu/books/oa-monograph/5047/Voicing-Code-in-STEMA-Dialogical-Imagination. DOI: https://doi.org/10.7551/mitpress/11668.001.0001

18. Tariq R., Aponte Babines B. M., Ramirez J., Alvarez-Icaza I., Naseer F.. Computational thinking in STEM education: current state-of-the-art and future research directions // Frontiers in Computer Science. – 2025 [Електронний ресурс]. – Режим доступу: https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2024.1480404/full

19. Weintrop D., Beheshti E., Horn M. Defining Computational Thinking for Mathematics and Science Classrooms // Journal of Science Education and Technology. – 2020. – № 25. – С. 127–147. DOI: https://doi.org/10.1007/s10956-015-9581-5

20. Wing J. M. Computational thinking. Communications of the ACM. – 2006. –№ 49(3). – P. 33–35. DOI: https://doi.org/10.1145/1118178.1118215

21. Yang D., Baek Y., Ching Y., Swanson S., Chittoori B., Wang S. Infusing Computational Thinking in an Integrated STEM Curriculum: User Reactions and Lessons Learned // European Journal of STEM Education. – 2021. – № 6 (1). [Електронний ресурс]. – Режим доступу: DOI: https://doi.org/10.20897/ejsteme/9559

22. Ye H., Liang B., Ng OL. Integration of computational thinking in K-12 mathematics education: a systematic review on CT-based mathematics instruction and student learning // International Journal of STEM Education. – 2023. – 10 (3). – [Електронний ресурс]. – Режим доступу: DOI: https://doi.org/10.1186/s40594-023-00396-w

THE SYNERGY OF PROGRAMMING AND NATURAL-MATHEMATICAL DISCIPLINES IN THE CONTEXT OF STEM EDUCATION

Pohromska Hanna,
PhD, Associate Professor
of the Department of Theory
and Methods for Teaching
the Natural Sciences, Mathematics
and Information Technologies
Mykolaiv In-Service Teachers Training Institute
4-a, Admiralska Street, 54001, Mykolaiv, Ukraine
hanna.pohromska@moippo.mk.ua

Makhrovska Natalya,
PhD, Associate Professor
of the Department of Theory
and Methods for Teaching
the Natural Sciences, Mathematics
and Information Technologies
Mykolaiv In-Service Teachers Training Institute
4-a, Admiralska Street, 54001, Mykolaiv, Ukraine
natalya.makhrovska@moippo.mk.ua

The article explores the synergy between programming, particularly with Python, and the system of STEM-oriented secondary education. Programming is no longer perceived as a solely technical skill; instead, it is increasingly recognized as a tool for developing students' computational thinking and as a foundation for interdisciplinary integration across mathematics, physics, chemistry, biology, and technology. Python, due to its simplicity and versatility, is identified as an optimal programming language for educational purposes at the school level. The paper emphasizes the potential of programming to serve as a connecting thread between STEM disciplines, enabling students to work with real data, construct models, visualize scientific concepts, and solve complex interdisciplinary tasks. The article analyzes current practices in Ukrainian schools, where Python is gradually being introduced within the computer science curriculum. However, this integration often remains fragmented and isolated from other subjects. Drawing on national and international research, as well as the authors' own teaching experience, this article identifies the key challenges of integrating Python into STEM education – including insufficient teacher preparation, a shortage of appropriate teaching materials, and limited access to equipment – and proposes practical solutions to address them. One of the highlights is the development and implementation of a certified teacher training course, «Start into the World of Coding: Python for Beginners», which promotes the transition from teaching Python as an isolated subject to using it as a tool for inquiry-based learning across disciplines. In addition, the article presents an open STEM lesson titled «Smart Coding: Algorithmic Structures in Python,» showcasing practical tasks that connect programming with mathematics and natural sciences. Through this experience, the authors demonstrate how programming facilitates the development of computational models, supports interdisciplinary project work, and enhances students' ability to think critically, analyze, and solve real-world problems.

Keywords: *computational programming; Python; STEM; synergy; thinking.*

REFERENCES

1. Cheng, L., Wang, X., & Ritzhaupt, A. (2023). View all authors and affiliations. *Journal of Educational Computing Research*, 61 (2), 416–443. DOI: https://doi.org/10.1177/07356331221114183 (eng).

2. Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. DOI: https://doi.org/10.3102/0013189X12463051 (eng).

3. Hurt, T., Greenwald, E., & Allan, S. (2023). The computational thinking for science (CT-S) framework: Operationalizing CT-S for K–12 science education researchers and educators. *Journal of Science Education and Technology*, 10, 1–16. DOI: https://doi.org/10.1186/s40594-022-00391-7 (eng).

4. Kobylnyk, T. P., Kohut, U. P. & Zhydyk, V. B. (2021). Metodychni aspekty vyvchennia osnov alhorytmizatsii i prohramuvannia movoiu Python u shkilnomu kursi informatyky u starshykh klasakh [Methodical aspects of teaching the basics of algorithmization and programming in Python in high school informatics]. *Fizyko-Matematychna Osvita*, 5 (31), 36–44 (ukr).

5. Kobylnyk, T. P., Sikora, O. V., Zhydyk, V. B., & Sharan, O. V. (2022). Python yak zasib navchannia osnov alhorytmizatsii u zakladakh zahalnoi serednoi osvity [Python as a tool for teaching the basics of algorithmization in secondary schools]. *Informatsiini Tekhnolohii i Zasoby Navchannia, 89* (3), 16–32. DOI: https://doi.org/10.33407/itlt.v89i3.4896 (ukr).

6. Lane, D., Galanti, T., & Rozas, H. L. (2023). Teacher re-novicing on the path to integrating computational thinking in high school physics instruction. *Journal for STEM Education Research*, 6, 302–325. DOI: https://doi.org/10.1007/s41979-023-00100-1 (eng).

7. Liskovych, O. V. (2023). Vykorystannia mozhlyvostei STEM-osvity u protsesi pidvyshchennia kvalifikatsii vchyteliv fizyky ta astranomii [Using STEM opportunities in teacher training]. *Veresen, 2* (97), 40–49 (ukr).

8. Liu, Z., Gearty, Z., & Richard, E. (2024). Bringing computational thinking into classrooms: A systematic review on supporting teachers in integrating computational thinking into K–12 classrooms. *International Journal of STEM Education*, 11(1). DOI: https://doi.org/10.1186/s40594-024-00510-6 (eng).

9. Lohakan, M., & Seetao, C. (2024). Large-scale experiment in STEM education for high school students using artificial intelligence kit based on computer vision and Python. *Heliyon,* 10 (10). DOI: https://doi.org/10.1016/j.heliyon.2024.e31366 (eng).

10. Mahrovska, N. A., & Pohromska, H. S. (2025). Praktyko-tekhnolohichni aspekty vprovadzhennia alhorytmizatsii ta prohramuvannia v osvitnii protses [Practical and technological aspects of implementing algorithmization and programming in education]. *Veresen,* 1 (104), 13–24. DOI: https://doi.org/10.54662/veresen.1.2025.02 (ukr).

11. Meechai, L., & Choochat, S. (2024). Large-scale experiment in STEM education for high school students using artificial intelligence kit based on computer vision and Python. Heliyon, 10. Retrieved from: https://pmc.ncbi.nlm.nih.gov/articles/PMC11129092/ (eng).

12. MIT OpenCourseWare. (n.d.). Massachusetts Institute of Technology. Retrieved from: https://ocw.mit.edu/ (eng).

13. Next Generation Science Standards. (n.d.). NextGenScience. Retrieved from: https://www.nextgenscience.org (eng).

14. PISA 2022 results (Volume I): The state of education post-COVID-19. Retrieved from: https://www.oecd.org/en/publications/pisa-2022-results-volume-i_53f23881-en.html (eng).

15. Rakuta, V. M. (2024). *Python u shkilnomu kursi informatyky. Osnovy prohramuvannia*. [Python in school informatics course. Programming basics]. Chernihiv: [Self-published] (ukr).

16. Sabashyna, Y. (2023). *Python u vosmomu klasi. Koly onovliat shkilnyi kurs informatyky ta yaki rezultaty pilotnoho proiektu* [Python in 8th grade: When the school course will be updated]. Retrieved from: https://dou.ua/lenta/news/computer-science-in-schools/?utm_source=chatgpt.com (ukr).

17. Sengupta, P., Dickes, A., & Farris, A. V. (2021). Integrating computational thinking in K–12 science and math education. In *Voicing code in STEM: A dialogical imagination* (MIT Press). DOI: https://doi.org/10.7551/mitpress/11668.001.0001 (eng).

18. Tariq, R., Aponte Babines, B. M., Ramirez, J., Alvarez-Icaza, I., & Naseer, F. (2025). Computational thinking in STEM education: Current state-of-the-art and future research directions. *Frontiers in Computer Science.* Retrieved from: https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2024.1480404/full (eng).

19. Weintrop, D., Beheshti, E., & Horn, M. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*, 127–147. DOI: https://doi.org/10.1007/s10956-015-9581-5 (eng).

20. Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49* (3), 33–35. DOI: https://doi.org/10.1145/1118178.1118215 (eng).

21. Yang, D., Baek, Y., Ching, Y., Swanson, S., Chittoori, B., & Wang, S. (2021). Infusing computational thinking in an integrated STEM curriculum: User reactions and lessons learned. *European Journal of STEM Education, 6* (1). DOI: https://doi.org/10.20897/ejsteme/9559 (eng).

22. Ye, H., Liang, B., & Ng, O. L. (2023). Integration of computational thinking in K–12 mathematics education: A systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education, 10* (3). DOI: https://doi.org/10.1186/s40594-023-00396-w (eng).

Стаття надійшла до редакції: 03.07.2025
Прийнято до друку: 24.09.2025