

Камінський О.Є.,

д. е. н., професор кафедри системного аналізу та кібербезпеки
КНЕУ імені Вадима Гетьмана

Лютій О.О.,

здобувач освіти

Kaminsky O.E,

Doctor of Economics, Professor of the Department of System Analysis
and Cybersecurity,

KNEU named after V. Hetman

Lyuty O.O.,

student

РОЛЬ ПРОГРАМУВАННЯ У РОЗВИТКУ КОМПЕТЕНТНОСТЕЙ З АНАЛІТИКИ ТА БЕЗПЕКИ В УМОВАХ ШІ

THE ROLE OF PROGRAMMING IN DEVELOPING ANALYTICS AND SECURITY COMPETENCES IN AI

Анотація. У статті розглядаються сучасні підходи до програмування як основу підготовки фахівців із смарт-аналітики та кібербезпеки в умовах стрімкого розвитку штучного інтелекту. Визначено ключові наукові аспекти, що супроводжують перетворення ролі розробника: від класичного кодування «з нуля» до взаємодії з генеративними мовними моделями та інструментами автоматизації. Наведено аналіз трьох основних освітніх парадигм — традиційної, адаптивної та генеративно-орієнтованої — із метою виявлення їхніх переваг і обмежень. Адаптивне навчання описане як технологія, що використовує алгоритми машинного навчання для персоналізації навчальних маршрутів, миттєвого зворотного зв'язку та автоматичної корекції складності завдань відповідно до індивідуальних потреб студента. Продemonстровано приклади впровадження адаптивних платформ (CybExer, AI-Cyber Range, Knewton Alta, Georgia Tech OMS), а також підбито підсумки мета-аналізу, що засвідчили зростання рівня успішності залученості студентів та зниження показника відмови. Окрему увагу приділено інтеграції генеративного штучного інтелекту у навчальні програми, який забезпечує автоматичну генерацію кейсів для виявлення вразливостей. Висвітлено та досліджено вплив ШІ-інструментів (Copilot, TabNine, Diffblue) на оптимізацію циклів розробки: автоматичне створення коду, тестування, документування й рефакторинг, що підвищує продуктивність розробників на 30–50 %. Розроблено структуру адаптивної онлайн-платформи, яка поєднує початкове оцінювання, побудову індивідуального плану, динамічну адаптацію контенту, практичні інтерактивні проекти, цикл зворотного зв'язку, систему нагадувань та підсумкову сертифікацію. Схематично відображено взаємодію компонентів системи. Дослідження підкреслює критичну важливість гібридного поєднання освітніх методик — традиційної, адаптивної та генеративної — для формування в студентів ключових компетенцій XXI століття: аналітичного мислення, здатності до само-навчання та адаптивності.

Ключові слова. Адаптивне навчання, генеративний штучний інтелект, смарт-аналітика, онлайн-платформа, освітні технології.

Abstract. This article examines contemporary approaches to programming as the foundation for training specialists in smart analytics and cybersecurity amid the rapid development of artificial intelligence. It identifies key scientific aspects accompanying the transformation of the developer's role—from classical «coding from scratch» to interacting with generative language models and automation tools. The analysis covers three main educational paradigms—traditional, adaptive, and generative-oriented—to reveal their advantages and limitations. Adaptive learning is described as a technology that employs machine learning algorithms to personalize learning paths, provide instant feedback, and automatically adjust task difficulty according to each student's individual needs. Examples of adaptive platforms (CybExer, AI-Cyber Range, Knewton Alta, Georgia Tech OMS) are demonstrated, and the results of a meta-analysis are summarized, showing increased student success and engagement alongside reduced dropout rates. Special attention is given to the integration of generative artificial intelligence into curricula, which enables automatic generation of vulnerability-detection case studies. The impact of AI-driven tools (Copilot, TabNine, Diffblue) on development cycles—automatic code generation, testing, documentation, and refactoring, boosting developer productivity by 30–50 %—is also explored. A structure for an adaptive online platform is proposed, combining initial assessment, individualized plan creation, dynamic content adaptation, practical interactive projects, a feedback loop, reminder system, and final certification. A schematic diagram illustrates the interaction of system components. The study underscores the critical importance of a hybrid combination of educational methodologies—traditional, adaptive, and generative—for cultivating in students the key 21st-century competencies of analytical thinking, self-directed learning, and adaptability.

Keywords. Adaptive learning; generative artificial intelligence; smart analytics; online -platform; educational technologies.

Постановка проблеми. У сучасному світі стрімкий розвиток штучного інтелекту (ШІ) трансформує підходи до аналізу даних та забезпечення кібербезпеки. Розробка програмного забезпечення для систем, що базуються на штучному інтелекті, вимагає нових підходів у програмуванні. Зокрема, необхідно враховувати специфіку навчання мовних моделей, обробки великих обсягів даних та забезпечення надійності таких систем. Дослідження в цій галузі акцентують увагу на важливості інтеграції методів машинного навчання та традиційного програмування для створення ефективних рішень. Це вимагає нових підходів до підготовки фахівців, здатних ефективно використовувати інструменти програмування для розробки інтелектуальних систем.

У цьому контексті, **метою даної статті** є аналіз наукових аспектів програмування в контексті підготовки таких фахівців, з особливою увагою до адаптивного навчання.

Аналіз останніх джерел та публікацій. Програмування для систем, що базуються на ШІ, вимагає врахування специфіки навчання моделей, обробки великих обсягів даних та забезпечення надійності таких систем. Дослідження в цій галузі акцентують увагу

на важливості інтеграції методів машинного навчання та традиційного програмування для створення ефективних рішень.

Так Martínez-Fernández та ін. у своїй роботі [1] підкреслюють, що розробка програмного забезпечення для ІІІ-систем вимагає нових підходів у програмуванні. Зокрема, необхідно враховувати специфіку навчання моделей, обробки великих обсягів даних та забезпечення надійності таких систем.

В роботі [2] було надано огляд алгоритмів машинного навчання, які можуть бути використані для інтелектуального аналізу даних та для автоматизації задач в кібербезпеці. Автор підкреслює потенціал таких алгоритмів у виявленні шкідливих дій, прогнозуванні порушень даних та запобіганні вторгненням.

Дослідники Vykopal та ін. у роботі «AI-Enhanced Cybersecurity Training: Learning Analytics in Action» [3] представили систему, яка аналізує відповіді студентів у кіберсимуляціях (CTF — Capture The Flag) і адаптує складність завдань. У результаті, рівень успішності студентів зріс на 27 % порівняно з традиційними підходами.

Генеративний штучний інтелект та великі мовні моделі (LLMs) трансформують підходи до навчання в галузі безпеки, автоматизуючи багато завдань, які раніше виконувалися вручну. В роботі Feng та Baker-Robinson [4] наведено опис навчальної програми, яка навчає студентів застосовувати генеративний штучний інтелект для вирішення проблем у сфері безпеки. Програма охоплює такі теми, як виявлення вразливостей, створення коду, аналіз загроз та соціальна інженерія.

Виклад основного матеріалу дослідження. Розвиток штучного інтелекту радикально змінює саму суть програмування, змінюючи роль програмістів та вимоги до їхніх навичок. Ключові аспекти впливу включають:

- Автоматизація написання коду: Інструменти на базі ІІІ, такі як GitHub Copilot, TabNine або CodeWhisperer, здатні генерувати цілі фрагменти програм на основі коротких підказок, що зменшує час розробки.
- Зміна ролі розробника: Програмісти переходять від написання коду «з нуля» до проєктування, перевірки та інтеграції результатів ІІІ.
- Підвищення продуктивності: Згідно зі звітом Microsoft Research (2023), використання Copilot підвищує продуктивність програмістів на 29–40 % залежно від складності завдань.
- Поява нових спеціалізацій: Інженер запитів (prompt engineer), спеціаліст з аудиту моделей, розробник етичних протоколів — нові напрямки в ІТ.

- Автоматизація тестування та валідації: ШІ автоматизує створення юніт-тестів, пошук помилок і навіть оптимізацію коду.
 - Впровадження генеративного ШІ у фреймворки: Tensor Flow, Hugging Face та інші інтегрують модулі, які полегшують використання складних моделей без необхідності розробки з нуля.
- Аналіз впливу штучного інтелекту на процес розробки програмного забезпечення наведено в таблиці 1.

Таблиця 1

АНАЛІЗ ВПЛИВУ ІНСТРУМЕНТІВ ШТУЧНОГО ІНТЕЛЕКТУ
НА ПРОЦЕС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Аспект програмування	Зміни під впливом ШІ	Приклади/Інструменти	Потенційний ефект
Створення коду	Генерація коду за описами	GitHub Copilot, CodeWhisperer	Прискорення розробки на 30–50 %
Рефакторинг та оптимізація	Автоматична оптимізація існуючого коду	ChatGPT, TabNine, Sourcery	Підвищення якості коду, зниження витрат на підтримку
Тестування ПЗ	Автоматичне створення тестів	Diffblue Cover, Testim.io	Скорочення часу тестування на 25–40 %
Навчання та підтримка розробників	Інтерактивні поради та виправлення	ChatGPT, Kite	Зниження порогу входу для новачків
Документування проєктів	Автоматичне створення документації	Mintlify	Спрощення підтримки та онбордингу команд
Кібербезпека	Аналіз вразливостей у коді	DeepCode, Codiga	Зменшення кількості помилок безпеки

Джерело: розроблено автором

ШІ значно підвищує ефективність розробки програмного забезпечення, але вимагає від фахівців нових компетенцій: уміння пра-

вильно формулювати запити (prompting), розуміння можливих помилок ІІІ та здатності працювати у середовищах із гібридною взаємодією людина–машина.

Інтеграція штучного інтелекту в розробку змінює програмування від лінійного процесу створення коду до більш творчого і стратегічного завдання — конструювання систем за допомогою потужних інструментів.

У сучасних умовах стрімкого розвитку штучного інтелекту традиційні методи навчання програмуванню втрачають свою ефективність, оскільки інструменти й фреймворки на основі штучного інтелекту (GitHub Copilot, TabNine, Diffblue тощо) з'являються та еволюціонують щомісяця. Фіксовані навчальні курси, підготовлені на початку семестру, не встигають відображати реальні потреби ринку праці. Потрібна адаптивна система, яка зможе оперативно оновлювати контент відповідно до нових релізів бібліотек та змін у професійних практиках. ІІІ-інструменти не лише генерують код, а й підказують оптимальні рішення. Застосування системи адаптивного навчання дозволяє моделювати ці сценарії: студентам показують, як за допомогою ІІІ автоматично створити основну структуру програми, а потім поступово вводять завдання, де вони самостійно коригують та поліпшують автогенерований код.

При роботі з ІІІ-асистентом помилки та неточності коду видно одразу — алгоритм підсвічує проблемні місця. Адаптивні платформи використовують цю перевагу: кожна невдала спроба супроводжується детальною підказкою, яка пояснює, що виправити, а не просто показує правильну відповідь.

Дослідження підтверджують, що людям легше сприймати інформацію в різних формах — відео, інтерактивні вправи, текстові довідники. Адаптивна платформа може перемикатися між цими форматами залежно від переваг користувача та ефективності засвоєння матеріалу. Адаптивне навчання (adaptive learning) — це підхід до освіти, при якому навчальний процес підлаштовується під потреби, рівень знань та темп засвоєння інформації кожного студента. У контексті підготовки фахівців зі смарт-аналітики та безпеки, це дозволяє формувати не лише фундаментальні знання з програмування, ІІІ, кібербезпеки, але й розвивати критичне мислення, аналітичні здібності та навички роботи з реальними даними. Порівняльний аналіз підходів до освіти в сучасних умовах наведено в таблиці 2.

Таблиця 2

ПОРІВНЯЛЬНИЙ АНАЛІЗ ПІДХОДІВ ДО ОСВІТИ В СУЧАСНИХ УМОВАХ

Підхід	Опис	Переваги	Недоліки
Традиційне навчання	Стандартні курси з фіксованим змістом	Структурованість, передбачуваність	Відсутність персоналізації, низька адаптивність
Адаптивне навчання	Персоналізовані маршрути навчання на основі аналізу даних	Підвищення ефективності, мотивації студентів	Високі вимоги до технологій та даних
Інтеграція генеративного ШІ	Використання LLMs для автоматизації завдань та навчання	Актуальність, підготовка до реальних викликів	Потреба в нових компетенціях у викладачів

Джерело: розроблено автором

Адаптивні освітні технології сьогодні інтегруються у реальні програми підготовки фахівців. Приклади їх практичного застосування наведено в таблиці 3.

Таблиця 3

ОСВІТНІ ПРОЕКТИ З ВИКОРИСТАННЯМ АДАПТИВНОГО ПІДХОДУ

Освітній проєкт	Опис	Результати
CybExer Learning Platform	Платформа для адаптивного навчання кібербезпеки з аналізом результатів CTF-симуляцій та автоматичним підбором рівня складності	Підвищення успішності курсантів на 31 %
AI-Cyber Range (EU Project)	Використання ШІ для моделювання кіберінцидентів і динамічного адаптування сценаріїв навчання	Підвищення швидкості реагування студентів на інциденти на 28 %
Knewton Alta	Адаптивне рішення для технічних курсів із підтримкою Python, SQL і аналізу логів безпеки	Зменшення кількості відрахувань на технічних спеціальностях на 12 %
Georgia Tech Online MSc in Cybersecurity	Використання адаптивних тестів і персоналізованих завдань для віддалених студентів	Підвищення рівня засвоєння матеріалу до 88 %

Джерело: розроблено автором

На основі метааналізу адаптивних програм у сфері ІТ (дані 2020–2024 рр.) можна зробити висновки щодо їх ефективності:

Таблиця 4

МЕТААНАЛІЗ АДАПТИВНИХ ПРОГРАМ У СФЕРІ ІТ

Показник	Традиційне навчання	Адаптивне навчання
Рівень успішності (%)	68 %	87 %
Залучення студентів (%)	52 %	79 %
Показник відмови (%)	18 %	5 %
Задоволеність студентів (%)	60 %	84 %

Джерело: розроблено автором на основі даних [6]

Адаптивне навчання показує стабільно вищі результати у всіх ключових метриках: від рівня успішності до залучення студентів та зниження кількості відрахувань.

Розглянемо приклад використання системи адаптивного навчання для освоєння мови програмування Python, на основі онлайн-платформа, яка динамічно підлаштовується під ваш рівень знань і стиль навчання. Систему адаптивного навчання для вивчення мови програмування Python наведено на рис.1.

Алгоритм роботи системи включає:

1. Початкове оцінювання. Короткий діагностичний тест (10 питань):

- Визначає базове розуміння синтаксису (змінні, цикли, умовні конструкції).
- Оцінює логічне мислення через прості задачі (наприклад, обчислення суми елементів списку).
- Анкета-перевага: скільки часу ви готові присвячувати, в яких форматах (відео, інтерактивні вправи, тексти) тощо.

На основі отриманих результатів система розподіляє вас у одну з трьох «трас»: новачок, якщо правильних відповідей менше 50 %; середній рівень, якщо 50–80 %; просунутий рівень, якщо понад 80 %.

2. Формування індивідуального плану. Новачки отримують упор на інтерактивні уроки з поясненнями крок за кроком, короткі відео та текстові довідки. Середній рівень отримує мікс із практичних завдань із тестовими кейсами, невеликих проєктів (наприклад, «реструктуруй список смартфонів за ціною»), а також теоретичних модулів. Здобувачі на просунутому рівні переходять одразу до

створення невеликих скриптів і роботі з бібліотеками (Pandas, requests), з викликом додаткових вправ на алгоритми та оптимізацію коду.



Рис. 1. Система адаптивного навчання для вивчення мови програмування Python

Джерело: розроблено автором

3. Динамічна адаптація всередині модуля. Припустімо, ви новачок і вивчаєте тему «Цикли for і while». Ви отримуєте пояснювальний блок: короткий текст із прикладами + відео (1–2 хв); невелику вправу: «Виведіть усі парні числа від 1 до 20» — система аналізує час рішення та кількість спроб. Якщо здобувач вирішив швидко і без помилок система одразу пропонує більш складну вправу на вкладені цикли. Якщо він зробив помилку — система миттєво надає підказку, пояснює, у чому могла бути логічна помилка, і пропонує ще один подібний приклад. Приклад перевірки: «Напишіть функцію, яка рахує суму елементів списку, використовуючи цикл» із трьома тестовими наборами вхідних даних. Якщо тест провалено двічі — дається розбір типових помилок і пропонується переглянути короткий «Refresh»-розділ. Якщо всі тести

пройдені — модуль вважається завершеним, і система розблоковує наступну тему («Списки і словники»).

4. Інтерактивний проєкт-застосунок. Після вивчення кількох базових тем платформа пропонує «міні-проєкт»: наприклад, створити простий текстовий чат-бот, що відповідає на команди користувача (привітання, запит поточного часу, тощо). Система оцінює ваш код на дотримання стилю, коректність роботи та дає автоматизовані коментарі: «Використайте метод `.lower()` для коректної обробки регістра».

5. Постійний зворотний зв'язок і рекомендації. Щоденний дайджест: розсилка коротких email-ів із підсумком пройденого (наприклад, «Сьогодні ви завершили 3 вправи на цикли, показник успішності 90 %») та рекомендаціями («Рекомендуємо погратися з методом спискового включення `list comprehension`»). Розклад нагадувань: якщо ви пропускаєте заняття більше 3 днів, система м'яко нагадує й повертає вас до теми, де виникли труднощі.

6. Підсумкова сертифікація. Після завершення базового курсу платформа проводить комплексний іспит із практичними завданнями, симуляцією реальних кейсів (обробка файлів, запити до API). За успішне проходження ви отримуєте сертифікат із рівнем володіння: *Beginner*, *Intermediate* або *Advanced Python Developer*.

Висновки. За допомогою отриманих результатів можна визначити, що адаптивне навчання показує стабільно вищі результати у всіх ключових метриках: від рівня успішності до залучення студентів та зниження кількості відрахувань. Адаптивне навчання для програмування — це не просто набір лекцій і вправ, а інтелектуальна система, яка визначає ваші сильні/слабкі сторони, миттєво підлаштовується під ваш темп і стиль, дає персоналізовані завдання та підказки, постійно мотивує зворотним зв'язком і реальними міні-проєктами. Цей підхід максимально наближує навчання до реальної практики розробки і допомагає освоїти Python швидко та ефективно.

Але попри очевидні переваги, адаптивні системи мають низку викликів: проблеми з приватністю даних студентів (особливо в ЄС), потреба у великих обсягах якісних даних для тренування моделей адаптації. Також проблемою для України є висока вартість розробки та підтримки адаптивних платформ та необхідність підготовки викладачів, здатних працювати з даними аналітики навчання.

Бібліографічні посилання

1. Martínez-Fernández, S., Rodríguez, P., Franch, X., & Biffl, S. (2021). Software engineering for AI-based systems: A survey. arXiv. <https://arxiv.org/abs/2105.01984>

2. Sarker, I.H. (2023). Machine learning for intelligent data analysis and automation in cybersecurity. *Annals of Data Science*, 10, 1473–1498. <https://doi.org/10.1007/s40745-022-00444-2>
3. Vykopal, J., Maresova, P., Cerny, T., & Vondrak, I. (2022). AI-enhanced cybersecurity training: Learning analytics in action. ResearchGate. https://www.researchgate.net/publication/388333196_AI-Enhanced_Cybersecurity_Training_Learning_Analytics_in_Action
4. Feng, W., & Baker-Robinson, D. (2025). A generative security application engineering curriculum. arXiv. <https://arxiv.org/abs/2501.10900>
5. Rostoka, M.L. (2024). Artificial intelligence in the system of information and analytical support for the training of scientific staff. Digital Library of the National Academy of Educational Sciences of Ukraine. <https://lib.iitta.gov.ua/id/eprint/739815/>
6. Jisc. (2025). Digital strategies in UK higher education: Making digital mainstream. <https://www.jisc.ac.uk/reports/digital-strategies-in-uk-higher-education-making-digital-mainstream>
7. Kaminsky, O.Y., Yereshko, Y.O., Kyrychenko, S.O., & Tulchinskiy, R.V. (2021). Training in digital entrepreneurship as a basis for forming the intellectual capital of nation. *Information Technologies and Learning Tools*, 81(1), 210–221. <https://doi.org/10.33407/itlt.v81i1.3899>

Статтю подано до редакції: 28.11.2024