

Мірошніченко М.Ю.

кандидат технічних наук, доцент,
доцент кафедри комп'ютерних наук
Таврійський державний агротехнологічний університет
імені Дмитра Моторного

Чернова Г.В.

кандидат педагогічних наук, доцент,
доцент закладу вищої освіти
кафедри вищої математики та інформатики
Харківський національний університет
імені В. Н. Каразіна

МОЖЛИВОСТІ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ПІДТРИМКИ ПРОГРАМУВАННЯ

Анотація. У статті висвітлені можливості штучного інтелекту для підтримки програмування. Зазначається, що штучний інтелект може бути використаний для створення програмного забезпечення (генерація та пояснення програмного коду, генерація тестів тощо) та під час навчання майбутніх ІТ-фахівців. З'ясовано, що системи штучного інтелекту, зокрема ChatGPT, можна розглядати як допоміжний та потужний освітній ресурс. Наголошується на окремих обмеженнях штучного інтелекту для створення та подальшого використання програмного коду.

Ключові слова: підготовка фахівців, штучний інтелект, програмування, онлайн сервіси, комп'ютерні науки.

Miroshnichenko M.Yu., Chernova H.V. Online services based on artificial intelligence for learning programming. The article highlights the potential of artificial intelligence (AI) to support programming. It is noted that AI can be successfully applied in software development (for code generation and explanation, test generation, etc.) as well as in the education of future programmers. It has been found that AI systems, particularly ChatGPT, can be considered as powerful and supportive educational resources. The article also emphasizes certain limitations of artificial intelligence in the creation and subsequent use of program code.

Key words: training of specialists, artificial intelligence, programming, online services, computer science.

Актуальність дослідження. В умовах стрімкого розвитку технологій штучний інтелект (ШІ) став корисним інструментом у багатьох сферах діяльності людини, зокрема в програмуванні. Завдяки інтеграції ШІ процес розробки програмного забезпечення (ПЗ) стає більш ефективним і продуктивним, допомагаючи професіоналам та новачкам створювати та тестувати програмні продукти. Системи на основі ШІ можуть автоматично генерувати код, аналізувати його, виявляти помилки та пропонувати оптимальні рішення, що суттєво зменшує рутинні дії та підвищує якість програмних продуктів. Водночас, постійний технологічний розвиток потребує аналізу нових можливостей штучного інтелекту під навчання програмуванню та розробки ПЗ.

Метою статті є висвітлення можливостей систем штучного інтелекту для розробки ПЗ та вивчення програмування.

Виклад основного матеріалу. В умовах стрімкого зростання можливостей інформаційно-комунікаційних технологій (ІКТ) та забезпечення вільного доступу до різноманітних електронних ресурсів процес вивчення програмування та створення програмного коду стає більш простим та ефективним. Інформаційні технології забезпечують доступ до інтерактивних і гнучких інструментів, які сприяють як формальному навчанню у закладах освіти, так і самостійній навчальній діяльності здобувачів вищої освіти.

Серед різноманіття цих ресурсів слід звернути увагу на масові відкриті онлайн курси, які містять значну кількість курсів з різних мов програмування [9, с. 18]. Особливістю онлайн курсів є наявність лекційного матеріалу, відеоуроків, тестових завдань, прогресу опанування онлайн курсом. Для вивчення програмування онлайн курси пропонують фрагменти програмного коду, пояснення до них, дошки оголошень для комунікації [6, с. 295] тощо. Досить часто користувачі переглядають відеороліки, розміщені на освітніх платформах та сервісах, зокрема YouTube. Вони зручні для швидкого опанування конкретними темами з мов програмування, фреймворків тощо. YouTube дозволяє переглядати коментарі інших переглядачів. Доповненням до таких ресурсів виступають інтерактивні середовища програмування, де користувач може створювати, тестувати та виконувати код безпосередньо в браузері [8, с. 80]. Окремий напрямок становлять онлайн-платформи для алгоритмічної практики, такі як Eolymp, LeetCode, HackerRank чи CodeSignal. Вони пропонують завдання різного рівня складності, організовані за категоріями, та забезпечують

автоматичну перевірку правильності виконання через компіляцію та тестування програмного коду. Такий підхід сприяє розвитку навичок алгоритмічного мислення, пошуку оптимальних рішень та підвищує рівень практичної підготовки [1, с. 56].

Останнім часом для навчання програмуванню та розробки програмного забезпечення використовується штучний інтелект, адже він відкриває нові можливості для автоматизації та оптимізації процесу розробки та підвищення якості розробки програмного забезпечення. На думку О. Гордієнко та А. Коваль, ШІ впливає на зміну підходів до розробки програмного забезпечення. Це втілюється у прискоренні циклу розробки ПЗ, коли автоматизація та генерація готових фрагментів програмного коду дозволяють розробникам витрачати час на вирішенні складних задач за рахунок звільнення від виконання рутинних операцій. За рахунок здатності систем ШІ аналізувати великі масиви даних і виявляти приховані закономірності забезпечується зменшення кількості критичних помилок, що впливає на підвищення якості програмного продукту та стабільності його роботи. Інноваційним підходом до розробки ПЗ є поява самонавчаючих систем, що дозволяють програмному забезпеченню адаптуватися до індивідуальних потреб користувачів та умов середовища. З точки зору освітньої діяльності завдяки ШІ-інструментам навіть нефахівці зможуть створити більш-менш складні програмні продукти за умови правильно складених запитів до системи [2, с. 42]. Особливо корисно це для підготовки майбутніх фахівців з ІТ-технологій, коли здобувач вищої освіти у відповідь на запит до системи штучного інтелекту отримує програмний код та пояснення до його роботи [5, с. 146].

Системи штучного інтелекту дедалі частіше застосовуються для підтримки розробників, надання рекомендацій щодо покращення структури програм, а також виявлення потенційних помилок ще до етапу тестування. Це робить штучний інтелект не лише технічним інструментом, а й партнером у процесі створення програмних рішень. В контексті допомоги розробникам ШІ має декілька переваг, а саме: написання фрагментів програмного коду з поясненнями його функціональності; зменшення кількості помилок завдяки автоматизованому тестуванню та аналізу коду, що допомагає виявляти проблеми ще на ранніх етапах розробки; аналіз готового програмного коду та пропонування альтернативи; можливість генерації тестів (юніт-

та інтеграційні тести), що підвищує якість програмного забезпечення та спрощує процес перевірки його надійності [3, с. 240].

Слід зазначити, що досить часто для вивчення програмування використовується ChatGPT. Це стосується не повноцінного курсу з вивчення конкретної мови програмування, але його окремих елементів на рівні консультації, пояснення програмного коду, надання готових фрагментів комп'ютерної програми. Зазначені можливості сали доступними завдяки появі великих мовних моделей (Large Language Model, LLM), що дозволили зробити досить великий прорив у сфері освітніх технологій. Вони аналізують і відтворюють тексти, використовуючи статистичні закономірності та мовні шаблони. Таким чином вони трансформують підхід до викладання й засвоєння навчального матеріалу. Подібні системи зменшують навантаження на викладачів, стандартизують оцінювання та дозволяють формувати більш персоналізовані освітні траєкторії [10, с. 129].

Уваги заслуговує здатність ChatGPT не лише створювати програмний код як розв'язок задач різної складності, але й пояснювати його логіку, пропонувати альтернативні підходи та демонструвати приклади використання бібліотек чи фреймворків. Це надає здобувачам вищої освіти можливість отримувати миттєвий зворотний зв'язок, що значно підвищує ефективність навчання, у тому числі програмуванню [1, с. 56]. Завдяки цьому ChatGPT може бути корисним інструментом для навчання здобувачів вищої освіти різних рівнів підготовки, які тільки починають вивчати програмування або мають значний досвід у створення програмних продуктів.

Сучасні інструменти штучного інтелекту не лише полегшують процес написання програмного коду, а й оптимізують усі етапи розробки програмного забезпечення. ШІ-системи постійно навчаються, що дозволяє їм розвивати здатність до самовдосконалення, точнішого розпізнавання контексту, адаптації до змінних вимог і більш глибокого розуміння запитів користувачів. Наразі існує значна кількість сервісів та інструментів штучного інтелекту для підтримки створення програмних продуктів від генерації програмного коду до оптимізації готових програмних рішень. Їх можна поділити на декілька видів, зокрема ШІ-інструменти для виправлення помилок та дебагінгу, генерації документації та пояснень програмного коду, ШІ-моделі для виконання специфічних потреб у програмуванні, нейронні мережі для оптимізації коду, ШІ-помічники для конкретних фреймворків та технологій, ШІ-

платформи для Data Science та машинного навчання, тестування вразливостей у програмному коді тощо. У довгостроковій перспективі такі інструменти можуть стати невід'ємною складовою середовища розробки, здатною формувати більш інноваційні, стабільні та адаптивні програмні рішення [4, с. 1317].

Слід зауважити, що використання ШІ для підтримки програмування пов'язано з певними ризиками. По-перше, згенерований програмний код не гарантує повної відсутності логічних, структурних чи алгоритмічних помилок, найбільш оптимального рішення для програмних завдань. По-друге, ефективність використання ШІ значною мірою залежить від активної участі людського розуму. Будь-який результат роботи системи потребує критичної оцінки з боку людини, включно з аналізом, коригуванням, додатковим навчанням і врахуванням контексту використання [4, с. 1316]. По-третє, актуальним питанням залишається безпека згенерованого коду. Автоматично створений код може містити вразливості, зокрема XSS та SQL-ін'єкції [7, с. 303], незахищені API-запити, відсутність перевірки прав доступу чи шифрування. Існує ризик плагіату або порушення авторського права, оскільки генеративні моделі іноді відтворюють фрагменти коду з відкритих репозиторіїв без зазначення джерела. Крім того, надмірна залежність від ШІ може знижувати рівень професійних навичок розробників. Постійне покладання на інструменти ШІ замість самостійного вирішення завдань, розвитку аналітичного мислення та вдосконалення власних навичок може призвести до поступової деградації технічної компетентності [3, с. 242].

Висновки. Отже, використання систем штучного інтелекту для вивчення програмування та розробки ПЗ дозволяє отримати багато переваг. Інтеграція ШІ не лише прискорює цикл розробки ПЗ, а й сприяє формуванню більш гнучких та високоефективних програмних рішень, що відповідають сучасним потребам цифрового середовища. Ці завдання досягаються за рахунок різноманітних онлайн сервісів, що дозволяють генерувати програмний код, пояснювати його функціональність, пропонувати оптимізовані рішення, створювати тести тощо.

Використання генеративного штучного інтелекту у вивченні програмування слід розглядати як допоміжний, але потужний освітній ресурс. Його поєднання з традиційними методами навчання та практичними завданнями дозволяє формувати комплексні професійні навички, сприяти розвитку критичного мислення та розширювати доступ

до якісної комп'ютерної освіти. Водночас, необхідне збалансоване поєднання автоматизації та інтелектуальної підтримки з боку ІІ з активною участю людини та критичним оцінюванням результатів.

Література

1. Величко В.Є., Ананьєв М.С., Іванюк С.В., Шеремет М.М. Електронне навчання у процесі вивчення програмування: Інформатика та методика її навчання. *Збірник наукових праць фізико-математичного факультету ДДПУ*. 2023. № 13. С. 54–61.
2. Гордієнко О., Коваль А. Майбутнє програмування: як штучний інтелект змінює розробку програмного забезпечення. *Інформаційні технології та суспільство*. 2024. № 4(15). С. 40–43.
3. Кравчук О. Штучний інтелект у програмуванні: як ІІ змінює підхід до розробки та автоматизації коду. *Вісник Хмельницького національного університету*. Технічні науки. 2024. Т. 345, № 6(2). С. 238–242.
4. Кривонос О.М. Використання генеративного АІ для створення програмного коду. *Наука і техніка сьогодні*. 2024. № 40. С. 1315–1326.
5. Михайлюк І., Мурава О. Впровадження штучного інтелекту в навчальний процес при вивченні дисципліни «основи програмування мовою Сі». *Інформаційні технології та суспільство*. 2025. № 1(16). С. 145–150.
6. Мірошніченко М.Ю. Використання МООС у якості додаткового засобу для вивчення об'єктно-орієнтованого програмування. *Українські студії в європейському контексті*. 2023. № 7. С. 293–299.
7. Назаров Є.М. Захист на рівні додатків: сучасні виклики та технології. *Українські студії в європейському контексті*. 2023. №7. С. 300–307.
8. Олефіренко Н.В., Пліско Л.О. Використання ресурсів YouTube для навчання школярів програмуванню ігор. *Науково-дослідна робота студентів як чинник удосконалення професійної підготовки майбутнього вчителя: зб. наук. пр.* 2000. № 19. С. 79–85.
9. Sharov S. et al. Using MOOC to Learn the Python Programming Language. *International Journal of Emerging Technologies in Learning (iJET)*. 2023. № 18(2023). Pp. 17–32.
10. Asy'ari M., Sharov S. Transforming education with ChatGPT: Advancing personalized learning, accessibility, and ethical AI integration. *International Journal of Essential Competencies in Education*. 2024. № 2(3). С. 119–157.