

## ШТУЧНИЙ ІНТЕЛЕКТ ЯК ІНСТРУМЕНТ КОГНІТИВНОЇ ПІДТРИМКИ ТА МЕНТОРСТВА У НАВЧАННІ ПРОГРАМУВАННЯ

Кривонос О. М., Житомирський державний університет імені Івана Франка, Україна

**Ключові слова:** генеративний ШІ, навчання програмуванню, цифровий ментор, методика навчання інформатики, когнітивна підтримка

Стрімкий розвиток генеративного штучного інтелекту (GenAI) спричинив фундаментальні зміни в ІТ-індустрії, що автоматично висуває нові вимоги до змісту та методики навчання інформатичних дисциплін. Традиційна модель навчання програмування, зорієнтована на механічне засвоєння синтаксису та написання шаблонних алгоритмів, втрачає свою ефективність, оскільки ці задачі тепер успішно делегуються інтелектуальним агентам. У педагогічному аспекті постає гостра потреба в перегляді цілей навчання: відтворення коду поступається місцем розвитку високорівневих когнітивних навичок – архітектурному мисленню, системному аналізу та здатності до верифікації складних технічних рішень.

Водночас інтеграція ШІ в освітній процес відкриває безпрецедентні можливості для реалізації концепції персоналізованого навчання та адаптивного менторства. Використання великих мовних моделей дозволяє кожному учневі отримати індивідуальну підтримку в режимі реального часу, що частково вирішує проблему дефіциту кваліфікованих викладачів-практиків та різного рівня базової підготовки в учнівських групах. Дослідження методичних засад використання ШІ як «цифрового ментора» є критично важливим для формування нової освітньої екосистеми, де технології не підміняють інтелектуальне зусилля учня, а стають інструментом інтенсифікації його когнітивного розвитку та професійного становлення.

Аналіз наукового доробку останніх років свідчить про активну фазу переосмислення ролі штучного інтелекту в ІТ-галузі та освіті. Зокрема, у роботах О. Гордієнка та А. Ковалю [1], а також О. Кравчука [2] детально висвітлено трансформаційний вплив ШІ на індустрію розробки програмного забезпечення, де акцентується на переході до автоматизації рутинних когнітивних операцій. Паралельно з цим, В. П. Олексюк та співавтори [5] розглядають цей процес крізь призму розвитку цифрової компетентності науково-педагогічних працівників, наголошуючи на необхідності готовності викладача до інтеграції інтелектуальних інструментів у професійну діяльність.

Особливої уваги заслуговують дослідження, присвячені безпосередньому впровадженню ШІ у навчальний процес. У праці О. М. Кривоноса [3] ґрунтовно аналізуються можливості використання генеративного ШІ для створення програмного коду, що закладає теоретичний фундамент для розуміння функціональних меж сучасних моделей. Ці ідеї знаходять своє прикладне втілення у роботі І. Михайлюка та О. Мурави [4], які описують досвід інтеграції штучного інтелекту при вивченні базових дисциплін, зокрема мови С. Разом з тим, педагогічний аспект використання ШІ як «цифрового ментора», здатного забезпечувати суб'єкт-суб'єктну взаємодію та когнітивну підтримку учня, потребує подальшого глибинного дослідження та методичного обґрунтування.

Сьогодні ШІ – це вже не просто програма, яка пише код за вас, а справжній «цифровий наставник». Це вимагає нових підходів до викладання інформатики. Тепер головне завдання ШІ не дати учню готовий результат, а підштовхнути його до самостійного рішення (так званий метод «підтримки»). Замість того, щоб просто копіювати код, учень стає критиком – він має перевірити, наскільки якісним є рішення від ШІ, чи швидко воно працює і чи немає в ньому помилок. Так ми вчимо дитину не просто механічно писати рядки коду, а глибоко аналізувати та оцінювати складні системи.

Важливою частиною навчання стає вміння спілкуватися з ШІ – так званий промпт-інжиніринг. По суті, це нова форма логічного мислення: щоб отримати добрий результат, учень має чітко розбити велику задачу на маленькі кроки та зрозуміло описати умови. Вчитель

тут допомагає учням налагодити «діалог» із системою: вчить ставити уточнюючі запитання, просити пояснення на прикладах або вимагати розв'язання задачі по частинах. Це перетворює просте читання теорії на активне дослідження. Навіть якщо ШІ помиляється, це стає корисним досвідом, адже змушує учня заглянути в підручники та документацію, щоб знайти правильну відповідь.

Окремо варто зупинитися на проблемі «інтелектуальних лінощів» або когнітивної пасивності. Коли готове рішення можна отримати за секунду, виникає спокуса вчитися поверхнево. Щоб цього уникнути, ми пропонуємо обов'язково додавати до занять етап рефлексії або «зворотного пояснення». Тепер мало просто показати працюючий код – учень має пояснити логіку кожного рядка, який видав ШІ, обґрунтувати, чому було обрано саме ці інструменти, і запропонувати свій варіант вирішення. Таким чином, оцінка ставиться не за те, що програма просто «запустилася», а за те, наскільки глибоко учень розуміє, як вона побудована.

З погляду індивідуального підходу, ШІ-ментор дозволяє кожному учневі йти власним шляхом навіть під час групових занять. Завдяки інтелектуальним помічникам можна підлаштовувати складність підказок під рівень конкретного студента: початківцю ШІ детально пояснить базові команди, а просунутому учневі запропонує обговорити складні архітектурні рішення. Це створює середовище, де ніхто не відчуває себе «відстаючим» або «перевантаженим», адже кожен отримує зворотний зв'язок у зручному для себе темпі. Зрештою, головна цінність ШІ полягає не в тому, щоб замінити вчителя, а в тому, щоб звільнити його час від рутини для спільного розв'язання справді творчих і складних задач.

Використання ШІ як «цифрового ментора» у навчанні програмування — це перехід від простого зазубрювання команд до розвитку системного мислення та вміння перевіряти складні рішення. Головна цінність такого підходу в тому, що учень отримує безперервну підтримку саме тоді, коли вона йому потрібна. Це дозволяє кожному вчитися у своєму темпі, але водночас ставить нові завдання перед викладачем. Тепер вчитель має оцінювати не просто результат роботи програми, а те, як учень пояснює свою логіку. Зрештою, ШІ не замінює педагога, а стає його союзником, перетворюючи вчителя на наставника, який допомагає опанувати складні інтелектуальні процеси та готує студентів до майбутнього у світі високих технологій.

### Список використаних джерел

1. Гордієнко, О., & Коваль, А. (2024). Майбутнє програмування: як штучний інтелект змінює розробку програмного забезпечення. *Інформаційні технології та суспільство*, 4(15), 40-43.
2. Кравчук, О. (2024). Штучний інтелект у програмуванні: як ШІ змінює підхід до розробки та автоматизації коду. *Вісник Хмельницького національного університету. Технічні науки*, 345(6/2), 238-242.
3. Кривонос, О. М. (2024). Використання генеративного AI для створення програмного коду. *Наука і техніка сьогодні*, (40), 1315-1326.
4. Михайлюк, І., & Мурава, О. (2025). Впровадження штучного інтелекту в навчальний процес при вивченні дисципліни «Основи програмування мовою Сі». *Інформаційні технології та суспільство*, 1(16), 145–150.
5. Олексюк, В. П., Спірін, О. М., Іванова, С. М., Мінтій, І. С., Вакалюк, Т. А., & Кільченко, А. В. (2025). Огляд досвіду використання штучного інтелекту для розвитку цифрової компетентності науково-педагогічних працівників. *Information Technologies in Education*, 2(58), 145-158.