УДК 004.42:004.8:371.3:373.5

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ЯК ІНСТРУМЕНТА ДЛЯ САМОПЕРЕВІРКИ ТА РОЗВИТКУ НАВИЧОК КОДУВАННЯ В УЧНІВ 7 КЛАСУ

Короленко Павло

Науковий керівник: кандидат педагогічних наук, доцент Лупан І. В.

Центральноукраїнський державний університет імені Володимира Винниченка, м. Кропивницький, Україна

Особливу цінність у процесі навчання програмування має самоперевірка – уміння аналізувати власний код. Однак початківці часто стикаються з помилками, які вони не можуть виправити самостійно, а вчитель не завжди має можливість надати допомогу кожному учневі індивідуально. У такому разі інструменти штучного інтелекту стануть у пригоді.

У статті показано як можна організувати практичні завдання з використанням інструментів штучного інтелекту для формування в учнів 7 класу навичок самоперевірки у процесі навчання програмування.

Ключові слова: навчання програмування; самоперевірка; штучний інтелект; практичні завдання.

Using Artificial Intelligence As A Tool For Self-Checking And Development Of Programming Skills In 7th Grade Students

P. Korolenko

Scientific supervisor: Candidate of Pedagogical Sciences, Associate Professor Lupan I.V.

Volodymyr Vynnychenko Central Ukrainian State University, Kropyvnytsky, Ukraine

Self-checking – the ability to analyze your own code – is a key to learning programming. However, beginners often f run into errors they cannot fix on their own, and the teacher can't always provide individual assistance. In such cases, artificial intelligence tools can be helpful.

This article shows how practical tasks using AI tools can be organized to help 7th-grade students develop self-checking skills while learning programming.

Keywords: learning programming; self-checking; artificial intelligence; practical tasks.

Постановка проблеми. Вивчення основ кодування у 7 класі є важливим елементом підготовки учнів до подальшого засвоєння інформаційних технологій. Однак традиційні підходи нерідко викликають труднощі: учні стикаються з помилками в коді, не завжди розуміють логіку алгоритмів та

потребують додаткових пояснень, які вчитель фізично не завжди може надати кожному індивідуально. У цьому контексті штучний інтелект (ШІ) стає ефективним інструментом підтримки, здатним забезпечити персоналізовану допомогу, гнучке навчальне середовище та можливість оперативного зворотного зв'язку. Особливу цінність у процесі навчання програмуванню має самоперевірка – уміння аналізувати власний код, знаходити помилки та вдосконалювати його. Інструменти ШІ можуть суттєво підсилити цей компонент, пропонуючи учням не лише автоматичний аналіз програм, а й пояснення причин помилок та рекомендації щодо покращення коду.

Аналіз досліджень і публікацій. Під штучним інтелектом розуміють технологію, яка забезпечує комп'ютерним системам здатність аналізувати інформацію, робити висновки, навчатися, узагальнювати попередній досвід та адаптуватися до нових умов [1].

Як зазначає О.В. Саган [2] однією з ключових педагогічних можливостей ШІ є персоналізація навчання. Інтелектуальні системи аналізують рівень підготовки учня, його типові помилки та динаміку навчання, пропонуючи індивідуально підібрані завдання. Для учнів 7 класу, які тільки починають знайомство з алгоритмами та програмуванням, персоналізовані рекомендації дозволяють уникнути когнітивного перевантаження та забезпечити поступове формування навичок.

В огляді *«A Systematic Review on Artificial Intelligence in Supporting Teaching Practice: Application Types, Pedagogical Roles, and Technological Characteristics»* [3] зроблено висновок, що використання інструментів ШІ сприяє формуванню індивідуальних освітніх траєкторій, підвищує мотивацію учнів до вивчення цифрових технологій, а також дозволяє здійснювати ефективну самоперевірку знань під час опанування базових навичок кодування.

Сучасні освітні технології активно інтегрують інструменти штучного інтелекту, що дозволяє підвищити ефективність навчального процесу, зокрема у викладанні кодування та програмування.

Основними типами інструментів ШІ для освіти називають [4]:

1. *Системи автоматизованого оцінювання та зворотного зв'язку*. Такі системи аналізують написаний учнем код, визначають наявність синтаксичних та логічних помилок, пропонують варіанти їхнього виправлення та оцінюють якість алгоритму. Це дозволяє учням проводити самоперевірку і отримувати практичні поради щодо вдосконалення свого коду без прямого втручання вчителя.

2. *Інтелектуальні тьютори та чат-боти.* Інтерактивні агенти на базі ШІ можуть пояснювати алгоритми, відповідати на запитання, надавати підказки під час виконання практичних завдань. Наприклад, платформи на кшталт ChatGPT, CodePilot або Codex здатні створювати приклади коду, аналізувати їх і давати поради щодо покращення.

3. *Системи адаптивного навчання*. Ці платформи аналізують результати виконання завдань учнями та підбирають наступні вправи відповідно до їхніх знань і навичок. Адаптивні системи дозволяють створювати індивідуальні освітні траєкторії, що особливо ефективно для учнів 7 класу, які мають різний рівень підготовки у програмуванні.

4. *Інструменти для моделювання та симуляцій*. ШІ дозволяє створювати інтерактивні симуляції алгоритмів та програм, де учні можуть експериментувати з різними сценаріями виконання коду. Це сприяє глибокому розумінню логіки програмування і розвитку аналітичного мислення.

Цікавий педагогічний підхід – використання агентів для саморегульованого навчання, таких як система Betty's Brain, було запропоновано дослідниками Leelawong K. та Biswas G. [5]. В описаній ними методиці учні «вчили» агента (Betty), будуючи моделі причинно-наслідкові зв'язки, що стимулювало їх до рефлексії стосовно власних помилок і того, як покращити пояснення. Таким чином система допомагала учням формувати навички самооцінювання й моніторингу.

Важливою педагогічною функцією ШІ є підтримка самоперевірки та автоматизованого оцінювання. Системи на основі штучного інтелекту здатні аналізувати написаний учнями код, знаходити синтаксичні або логічні помилки,

пропонувати пояснення та варіанти виправлення. Це не лише підвищує ефективність навчання, але й сприяє розвитку критичного мислення та самостійності [6].

В огляді *«The Influence of Artificial Intelligence Tools on Learning Outcomes in Computer Programming»* [7] зроблено висновки про те, що ШІ сприяє розвитку алгоритмічного мислення через:

- пропонування алгоритмічних підказок (які кроки слід виконати далі);
- демонстрацію альтернативних способів розв'язування задачі;
- спрощення аналізу логічних блоків програми;
- миттєву відповідь на запитання «чому програма працює саме так?».

Інтеграція штучного інтелекту як навчального помічника відкриває нові можливості для підтримки кожного учня незалежно від рівня його підготовки [8]. ШІ виконує семантичний аналіз написаної учнем програми: читає структуру алгоритму, співставляє її з умовами завдання, оцінює коректність використаних операторів, перевіряє логіку порівнянь, циклів, умовних конструкцій. У випадку, якщо в коді є неточності, ШІ пояснює їх у доступній формі, розкриваючи причину помилки. При цьому учень отримує не просто вказівку «що неправильно», а й роз'яснення «чому саме це не працюватиме».

Метою статті є розробка практичного заняття для формування в учнів 7 класу навичок самоперевірки у процесі навчання програмування з використанням ШІ.

Виклад основного матеріалу (результатів) дослідження. У 7 класі учні тільки починають опановувати абстрактні концепції алгоритмів, тому самоперевірка з автоматизованим зворотним зв'язком є особливо ефективною: учень отримує підтримку, не чекаючи перевірки вчителем. Повторні спроби та доступ до пояснень допомагають зміцнити розуміння базових структур (цикли, умовні оператори), що є фундаментом подальшого вивчення програмування. Формування самоконтролю у 7 класі створює основу для майбутньої автономної роботи в програмуванні: учні навчаться самостійно шукати помилки, тестувати

й виправляти код – навички дуже корисні і в старших класах, і в позашкільному програмуванні.

Самоперевірка є критичною складовою процесу навчання програмування, особливо на початкових етапах вивчення алгоритмів та кодування учнями середньої школи. У контексті 7 класу вона сприяє розвитку відповідальності, самостійності та глибшого розуміння логіки програмування.

Самоперевірка через автоматизовані інструменти у навчанні програмування є потужним педагогічним засобом: вона дає формувальний зворотний зв'язок, сприяє когнітивному розвитку, підвищує мотивацію та самостійність учнів. Для учнів 7 класу, які лише починають знайомитися з кодуванням, така форма самоперевірки є надзвичайно цінною, оскільки забезпечує підтримку, яка не залежить від миттєвого втручання вчителя, і допомагає систематично формувати навички алгоритмічного мислення та якості програмного коду.

Організація практичного заняття з використанням ШІ

Практичне заняття проводиться у комп'ютерному класі з використанням онлайн-платформи зі штучним інтелектом (наприклад, ChatGPT, Code.org AI Assistant або Replit AI). Нижче наведено повний сценарій заняття, інструкції для учнів і приклади виконаних робіт.

Структура проведення заняття.

Етап 1. Підготовка робочого середовища (Рис. 1)

1. Учні запускають браузер та переходять на обраний інструмент (наприклад, https://chat.openai.com).
2. Створюють новий файл main.py.
3. У чаті з ШІ вводять запит: «Допоможи мені на уроці програмування. Я вчуся у 7 класі.»

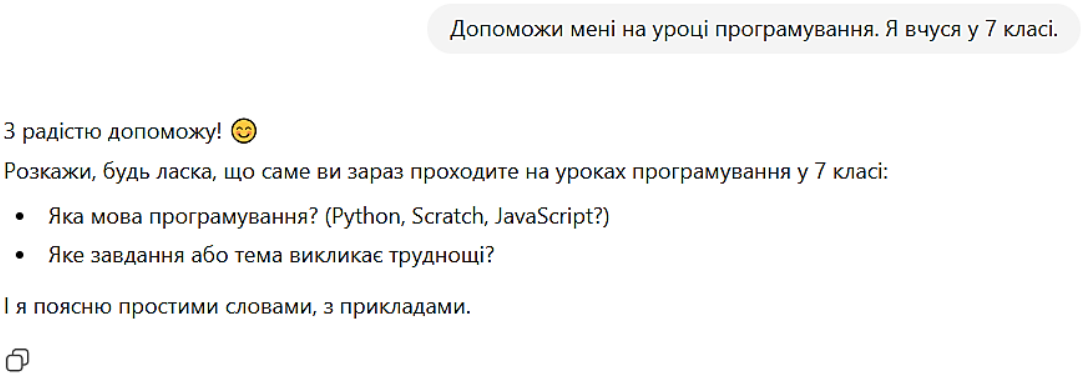

Рис. 1. Підготовка робочого середовища

Етап 2. Формування першого запиту до ШІ (Рис. 2)

Учень відкриває чат та вводить запит, який визначає майбутню програму.

Приклад запиту: «Створи просту програму на Python для 7 класу, де комп'ютер загадує число від 1 до 10, а користувач намагається його вгадати».

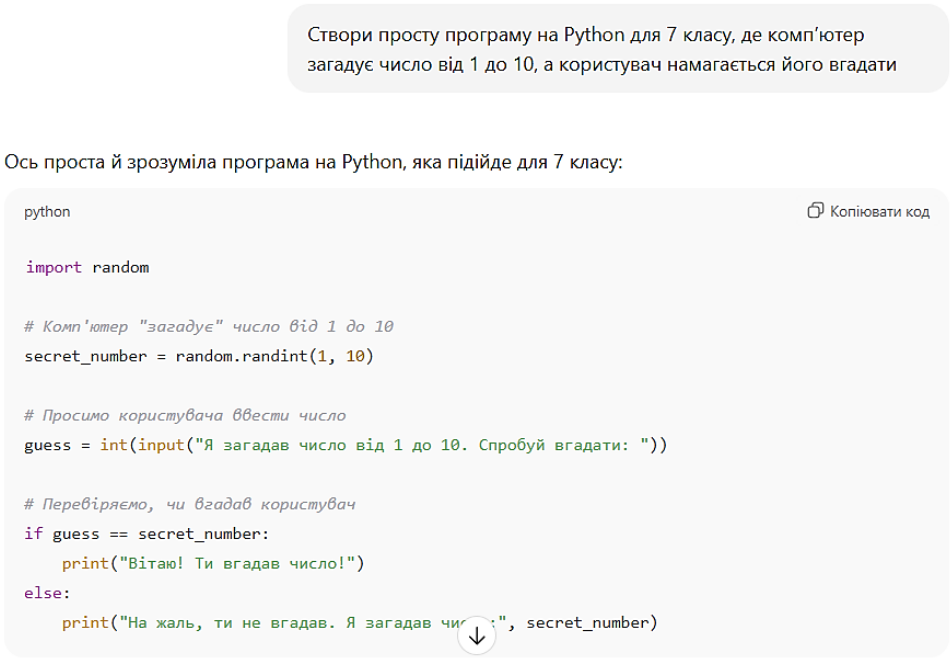

Рис. 2. Написання першого запиту, формулювання задачі

ШІ генерує зрозумілий учневі код, який є стартовою точкою роботи.

Етап 3. Отримання та перевірка коду, який згенерував ШІ

1. Учень читає згенерований код:

```
import random

# Комп'ютер "загадує" число від 1 до 10
secret_number = random.randint(1, 10)

# Просимо користувача ввести число
guess = int(input("Я загадав число від 1 до 10. Спробуй вгадати: "))
```

```
# Перевіряємо, чи вгадав користувач
if guess == secret_number:
    print("Вітаю! Ти вгадав число!")
else:
    print("На жаль, ти не вгадав. Я загадав число:", secret_number)
```

2. Ставить уточнювальні питання до ШІ, якщо щось незрозуміло, наприклад, «Поясни, що означає рядок random.randint(1, 10)».
3. Може попросити ШІ додати покращення: «Додай лічильник спроб» (Рис. 3).
4. Або ускладнити програму: «Додай підказки: більше чи менше».

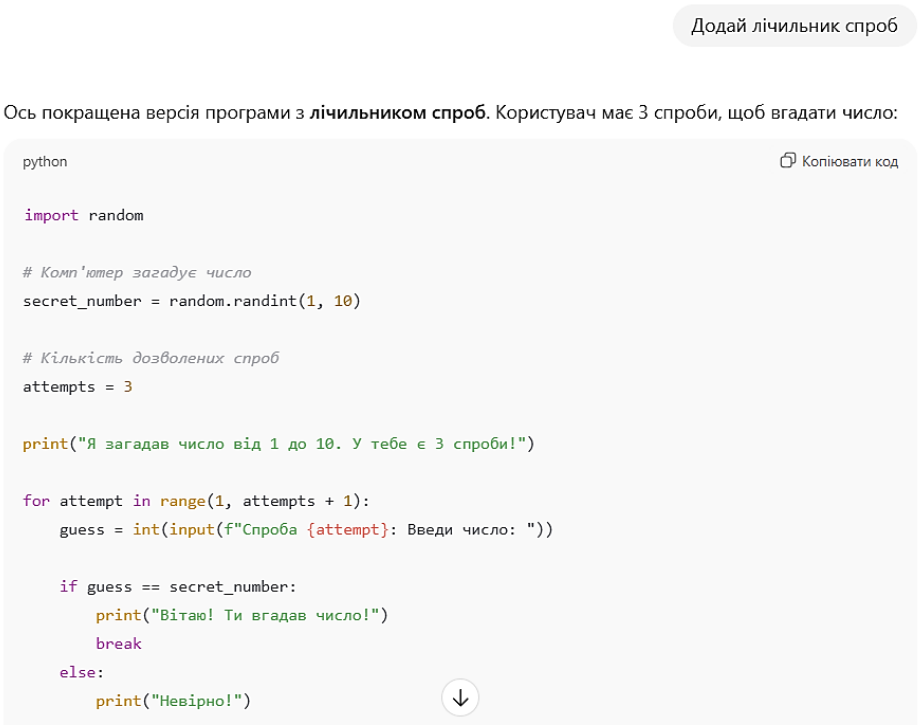

Рис. 3. Покращення результату

Етап 4. Перевірка та аналіз результатів

У процесі практичної роботи учень переносить код, створений ШІ, в середовище програмування (Рис. 4, Рис. 5).

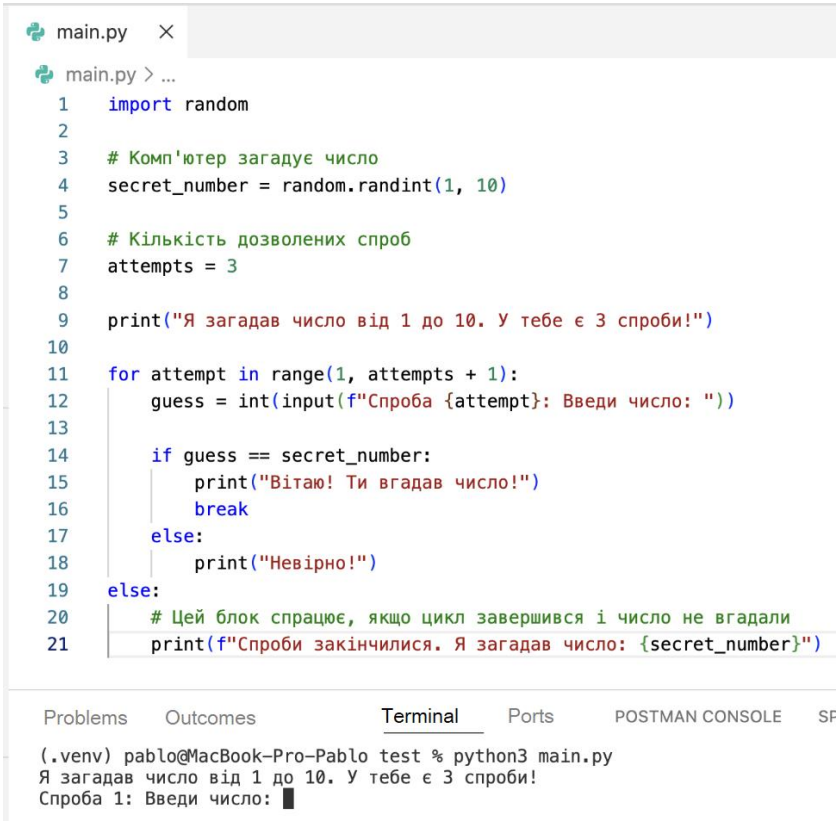

Рис. 4. Запуск програми, згенерованої ШІ

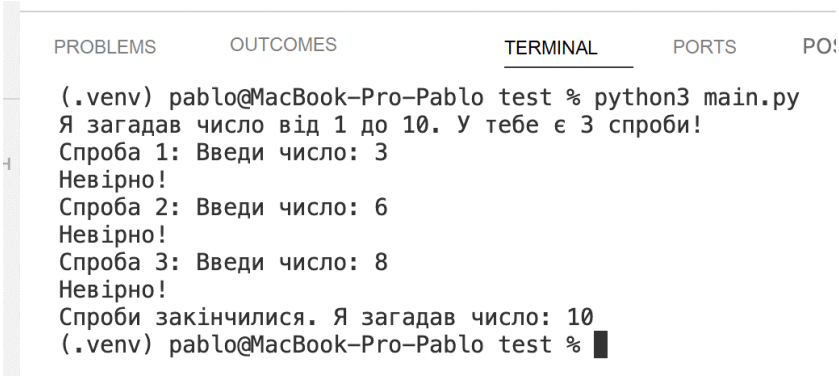

Рис. 5. Коректний результат програми

На наведеному скріншоті (Рис. 5) видно реальний результат роботи програми:

- програма коректно виводить вступне повідомлення: *«Я загадав число від 1 до 10. У тебе є 3 спроби!»*;
- кожна спроба супроводжується підказкою: *«Спроба 1: Введи число:»*;
- після введення числа програма правильно перевіряє його та виводить: *«Невірно!»* – якщо відповідь не співпадає із загаданим числом;
- кількість спроб зменшується відповідно до логіки циклу;

– після вичерпання всіх трьох спроб програма коректно завершує роботу повідомленням: *«Спроби закінчилися. Я загадав число: 10»* – де «10» є дійсним згенерованим випадковим числом.

Аналіз показує, що програма працює повністю коректно, тому що:

1. Генерує випадкове число у заданому діапазоні.
2. Приймає і обробляє введення користувача без збоїв.
3. Правильно порівнює введене число із загаданим.
4. Коректно відраховує кількість спроб.
5. Виводить результат, який відповідає логіці гри.
6. Завершує роботу без помилок і з правильним повідомленням.

Завдяки застосуванню ШІ процес створення програм у 7 класі перетворюється на послідовність дій: запит → генерація коду → пояснення → перенесення у середовище програмування → виконання → виявлення помилок → покращення → готова програма. Учень отримує досвід повноцінної роботи з програмою – від ідеї до реального запуску – у співпраці з інструментами штучного інтелекту.

Висновки та перспективи подальших пошуків у напрямі дослідження. Викладання основ програмування у 7 класі традиційно передбачає засвоєння базових понять: алгоритм, змінна, умова, цикл, послідовність команд. Для більшості учнів ці поняття є складними, оскільки вимагають розвитку абстрактного й алгоритмічного мислення. Завдяки інтелектуальним рекомендаціям учні швидше засвоюють базові структури алгоритмів – послідовність, розгалуження, повторення – та краще розуміють логіку роботи програм.

Наш досвід впровадження інструментів ШІ у навчання програмування показав, що за умови відповідної постановки завдання ШІ може стати гарним помічником для вчителів і учнів, адже використання ШІ забезпечує миттєвий зворотний зв’язок, надає індивідуальні підказки, підтримує самостійну роботу учнів. За допомогою ШІ учні вчаться аналізувати та виправляти свої помилки, що, у свою чергу, сприяє розвитку алгоритмічного мислення та цікавості до

навчання. У подальшому планується розробити рекомендації до використання засобів ШІ для самостійної роботи під час навчання програмування у 7 класі.

Список використаної літератури:

1. Copeland B.J. artificial intelligence: веб-сайт. URL: https://www.britannica.com/technology/artificial-intelligence. (Дата звернення: 25.11.2025)
2. Саган О. В. Організація персоналізованого навчання за допомогою штучного інтелекту. *Педагогічні науки: зб. наук. праць.* Івано-Франківськ : ХДУ, 2024. Вип. 108. С. 37-43. URL: https://ekhsuir.kspu.edu/handle/123456789/20708
3. Shi Lehong, Ikseon Choi. A Systematic Review on Artificial Intelligence in Supporting Teaching Practice: Application Types, Pedagogical Roles, and Technological Characteristics', in Xiaoming Zhai, and Joseph Krajcik (eds), *Uses of Artificial Intelligence in STEM Education* (Oxford, 2024; online edn, Oxford Academic, 21 Nov. 2024). https://doi.org/10.1093/oso/9780198882077.003.0015 (дата звернення: 25.11.2025).
4. Top 10 AI Education Tools for Modern Learning (2025). (n.d.). Disco. URL: https://www.disco.co/blog/top-10-ai-education-tools (Дата звернення: 25.11.2025)
5. Leelawong K., Biswas G. Designing Learning by Teaching Agents: The Betty's Brain System. *International Journal of Artificial Intelligence in Education,* 18(3). 2008. P. 181-208. DOI: https://doi.org/10.3233/irg-2008-18(3)02
6. Мар’єнко М. В., Коваленко В. В. Використання вчителями сервісів штучного інтелекту у навчанні природничо-математичних предметів у закладах загальної середньої освіти. *Освіта та розвиток обдарованої особистості.* № 1. 2024. С. 78-83. https://doi.org/10.32405/2309-3935-2024-1(92)-78-83
7. Alanazi M., Soh B., Samra H., Li A. The Influence of Artificial Intelligence Tools on Learning Outcomes in Computer Programming: A Systematic Review and Meta-Analysis. *Computers, 14*(5), 185. 2025. 37 pp. DOI: https://doi.org/10.3390/computers14050185
8. Araujo L., Lopez Ostenero F., Plaza L., Martinez Romo J. Automated Formative Feedback for Algorithm and Data Structure Self Assessment. *Electronics, 14*(5), 1034, 2025. 30 pp. DOI: https://doi.org/10.3390/electronics14051034

Відомості про автора:

Короленко Павло Володимирович – студент IV курсу бакалаврату факультету інформаційних технологій, математики та природничих наук Центральноукраїнського державного університету імені Володимира Винниченка, e-mail: 3849408859@cuspu.edu.ua.