

УДК 004.896:378.147:004.42

DOI <https://doi.org/10.32689/maup.it.2025.1.18>

Ірина МИХАЙЛЮК

кандидат педагогічних наук, доцент,
доцент кафедри інженерії програмного забезпечення,
Івано-Франківський національний технічний університет нафти і газу,
iryna.mykhailiuk@nung.edu.ua
ORCID: 0000-0002-6489-3982

Ольга МУРАВА

студентка II курсу факультету інформаційних технологій,
Івано-Франківський національний технічний університет нафти і газу,
olha.murava-ip231@nung.edu.ua

ВПРОВАДЖЕННЯ ШТУЧНОГО ІНТЕЛЕКТУ В НАВЧАЛЬНИЙ ПРОЦЕС ПРИ ВИВЧЕННІ ДИСЦИПЛІНИ «ОСНОВИ ПРОГРАМУВАННЯ МОВОЮ СІ»

Анотація. У статті досліджуються можливості впровадження штучного інтелекту (ШІ) у навчальний процес із дисципліни «Основи програмування мовою Сі».

Мета дослідження – вивчення шляхів інтеграції ШІ в освітній процес для підвищення ефективності навчання програмуванню мовою Сі. Враховуючи ключову роль мови Сі в ІТ-освіті, її опанування є важливим етапом для студентів, які прагнуть розвивати навички програмування та працювати з сучасними мовами, такими як С++, Java чи Python. Окрім того, знання мови Сі необхідні для професійної діяльності в галузях, пов'язаних із розробкою операційних систем, драйверів та вбудованих систем.

Методологія дослідження базується на аналізі сучасних наукових джерел, що розглядають питання застосування ШІ у навчальному процесі. Основна увага приділяється алгоритмам автоматизації перевірки програмного коду, методам аналізу ефективності алгоритмів, а також персоналізованому підходу до навчання студентів. У статті розглядаються механізми інтеграції ШІ у навчальний процес, зокрема автоматичне виявлення та аналіз помилок у коді, оптимізація алгоритмів, генерація тестових завдань та можливості доповнення програмного коду.

Наукова новизна дослідження полягає у практичному підході до інтеграції ШІ в навчання мовою Сі. Використання інтелектуальних систем дозволяє адаптувати освітній процес до рівня підготовки студентів, що сприяє глибшому розумінню принципів програмування та підвищенню їхньої компетентності у роботі з алгоритмами та структурами даних. Особливий акцент зроблено на взаємодії студента з ШІ, де алгоритми аналізують помилки, пропонують виправлення та надають миттєвий зворотний зв'язок, що підсилює навчальний

Висновки статті підкреслюють перспективність використання ШІ в освітньому процесі. Основні переваги впровадження таких технологій включають адаптивність навчання, швидкий доступ до зворотного зв'язку, можливість створення варіативних завдань та розвиток практичних навичок студентів. Впровадження ШІ у навчання мовою Сі дозволяє не лише підвищити ефективність засвоєння матеріалу, але й зробити освітній процес більш інтерактивним та цікавим для студентів.

Ключові слова: штучний інтелект, мова програмування Сі, автоматизація, персоналізація навчання, оптимізація алгоритмів.

Iryna MYKHAILIYK, Olha MURAVA. INTEGRATION OF ARTIFICIAL INTELLIGENCE INTO THE EDUCATIONAL PROCESS FOR TEACHING THE DISCIPLINE «FUNDAMENTALS OF PROGRAMMING IN C LANGUAGE»

Abstract. The article explores the possibilities of integrating artificial intelligence (AI) into the educational process for the course «Fundamentals of Programming in C».

The aim of the study is to examine ways of integrating AI into education to improve the effectiveness of learning programming in C. Given the key role of the C language in IT education, mastering it is an essential step for students who aspire to develop programming skills and work with modern languages such as C++, Java, or Python. Additionally, knowledge of C is crucial for professional activities in fields related to operating system development, drivers, and embedded systems.

The research methodology is based on the analysis of modern scientific sources that discuss the application of AI in education. The primary focus is on algorithms for automating code verification, methods for analyzing algorithm efficiency, and personalized approaches to student learning. The article examines AI integration mechanisms in education, including automatic error detection and analysis in code, algorithm optimization, test task generation, and code augmentation.

The scientific novelty of the study lies in its practical approach to integrating AI into C language education. The use of intelligent systems enables the adaptation of the educational process to the students' level of preparation, fostering a deeper understanding of programming principles and enhancing their competence in working with algorithms and data structures. Special emphasis is placed on student interaction with AI, where algorithms analyze errors, suggest corrections, and provide instant feedback, strengthening the learning effect. Another important aspect of the study is the evaluation of the effectiveness of an interactive learning approach, implemented through adaptive algorithms and automated learning platforms.

The article's conclusions highlight the promising prospects of AI usage in education. The key benefits of implementing such technologies include adaptive learning, quick access to feedback, the ability to create variable tasks, and the development of students' practical skills. AI integration into C language education not only enhances material comprehension efficiency but also makes the learning process more interactive and engaging for students.

Key words: artificial intelligence, C programming language, automation, personalized learning, algorithm optimization.

Постановка проблеми в загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями. Впровадження штучного інтелекту (ШІ) в освітній процес стає важливим напрямом розвитку сучасної освіти, особливо у сфері технічних наук. Одним із найактуальніших питань є інтеграція ШІ у навчання програмуванню, зокрема дисципліни «Основи програмування мовою Сі». Традиційні методи викладання часто не відповідають швидким змінам у технологічному середовищі, що створює додаткові труднощі для студентів та викладачів. Студенти стикаються з проблемами в розумінні базових концепцій програмування, таких як структури даних, цикли та функції, а викладачі витрачають значний час на перевірку рутинних завдань і пояснення типових помилок.

Аналіз останніх досліджень та публікацій. Сучасні дослідження демонструють високий потенціал використання ШІ для розв'язання цих проблем. У статті [3] акцентується увага на важливості інтеграції ШІ в навчальні програми, його ролі у створенні персоналізованих стратегій викладання та адаптації матеріалів до рівня знань студентів. Дослідження [2] зосереджується на тому, як штучний інтелект сприяє автоматизації оцінювання завдань, генеруванню персоналізованих рекомендацій та забезпеченню зворотного зв'язку. Висновки авторів підкреслюють можливості оптимізації навчального процесу завдяки таким технологіям.

Робота [6] аналізує вплив адаптивних алгоритмів на ефективність навчання студентів з різним рівнем підготовки. Автори розглядають, як ці технології можуть персоналізувати навчальний процес та покращити засвоєння матеріалу відповідно до індивідуальних потреб студентів. У статті розглядаються такі інструменти, як CodeSignal, LeetCode і HackerRank, які забезпечують автоматичне оцінювання коду, аналіз помилок і адаптивні рекомендації. Автори також зазначають, що такі платформи найбільше використовуються студентами молодшого віку.

Особливу увагу питанням інтерактивного навчання з використанням ШІ приділено в роботі [8]. Дослідники підкреслюють важливість інтеграції ШІ для створення інтерактивних матеріалів і забезпечення оперативного зворотного зв'язку, що сприяє підвищенню якості навчання.

Наукові публікації [1] висвітлюють етичні аспекти впровадження ШІ у навчання, зокрема забезпечення рівного доступу до технологій і мінімізацію ризиків залежності від автоматизованих рішень.

Таким чином, аналіз сучасних досліджень свідчить про значний потенціал впровадження ШІ у навчальний процес дисципліни «Основи програмування мовою Сі».

Метою цієї статті є дослідження можливостей використання ШІ для автоматизації рутинних завдань, персоналізації навчання та адаптації матеріалів до потреб студентів. Основна увага приділяється підвищенню ефективності навчання через автоматизацію перевірки коду, персоналізацію навчальних завдань та розвиток критичного мислення й практичних навичок студентів.

Виклад основного матеріалу

Значення мови програмування Сі. Мова програмування Сі є однією з найважливіших для студентів ІТ-спеціальностей. Її значення важко переоцінити, адже саме Сі часто стає базою, з якої починається знайомство з програмуванням і розробкою програмного забезпечення [7, 10]. Вивчення цієї мови забезпечує розуміння фундаментальних принципів алгоритмізації та структурування програм, що є основою для подальшого освоєння сучасних мов програмування, таких як C++, Java чи Python.

Однією з ключових причин вивчення Сі є її ефективність і продуктивність. Завдяки можливості прямого управління пам'яттю та низькорівневим особливостям, Сі дозволяє створювати програми, які працюють максимально швидко та оптимально використовують ресурси системи. Це робить її незамінною в таких галузях, як розробка операційних систем, драйверів та програмного забезпечення для вбудованих систем [9]. Для студентів, які планують працювати у цих сферах, знання Сі є важливою складовою професійної компетенції.

Крім того, Сі використовується у багатьох сферах: від вбудованих систем і наукових обчислень до ігрової індустрії. Це універсальна мова, яка залишається актуальною протягом десятиліть. Її стабільність та широке застосування гарантують, що знання Сі завжди будуть корисними для будь-якого ІТ-спеціаліста.

Інтеграція ШІ у вивчення мови програмування Сі. ШІ має великий потенціал у вдосконаленні навчального процесу [10], зокрема, програмуванню мовою Сі. Одним із важливих аспектів є автоматизація перевірки коду. Інтелектуальні алгоритми здатні аналізувати код студентів, знаходити помилки та навіть пояснювати їх суть. Наприклад, після завантаження програми з помилкою, ШІ знаходить помилку та в коментарях описує проблему (рис. 1):

```
int main() {
    int x;
    printf("%d", x); // Помилка: змінна не ініціалізована.
    return 0;
}
```

Рис. 1. ШІ вказує на помилку у програмному коді

ШІ може запропонувати рекомендації для виправлення (рис. 2):

```
int main() {
    int x = 0;
    printf("%d", x); // Проблема вирішена.
    return 0;
}
```



Рис. 2. ШІ виправляє помилку

Розглянемо практичні завдання з використанням ШІ:

1. Завдання з доповнення коду. ШІ пропонує шаблон функції, а студент має завершити її (рис. 3):

```
int findMax(int arr[], int n) {
    int max = arr[0];
    // Ваш код тут
    return max;
}
```

Рис. 3. Запропонований ШІ шаблон функції

Завдання для студента: завершити функцію для знаходження максимального/мінімального елемента.

Студенти вчаться правильно визначати параметри та повертати значення функцій, використовувати порівняння для аналізу даних, застосовувати ітерацію для обробки списку значень та розвивати самостійне мислення, що допомагає у творчому розв'язанні задач

2. Аналіз помилок. ШІ може генерувати код із навмисно внесеними помилками, які студенти повинні виявити та виправити (рис. 4):

```
int sumArray(int arr[], int n) {
    int sum = 0;
    for (int i = 0; i <= n; i++) { // Помилка: цикл виходить за межі масиву
        sum += arr[i];
    }
    return sum;
}
```



Рис. 4. Згенерований ШІ код з помилками

Аналіз помилок у коді допомагає студентам зрозуміти причини виникнення помилок, розвиває аналітичні навички та готує їх до вирішення складніших завдань у майбутньому.

3. *Оптимізація алгоритмів.* Оптимізація коду полягає у створенні програм, які працюють швидше, займають менше пам'яті та є більш зрозумілими для інших розробників. ШІ допомагає студентам зрозуміти ключові принципи ефективного кодування та розвиває у них важливі навички

ШІ може пропонувати студентам базовий алгоритм, який потребує вдосконалення (рис. 5).

```
void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}
```

Рис. 5. Згенерований ШІ фрагмент програми для оптимізації

Завдання для студента: замінити алгоритм сортування на більш ефективний, наприклад, швидке сортування (quick sort) (рис. 6).

```
void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}
```

Рис. 6. Приклад швидкого сортування

4. *Генерація тестів.* ШІ допомагає створювати тести для перевірки функцій, наприклад, для перевірки сортування масиву (рис. 7).

```
void testSortingAlgorithm() {
    int testArray[] = {5, 3, 8, 1, 2};
    quickSort(testArray, 0, 4);
    // Перевірка результату
    for (int i = 0; i < 4; i++) {
        if (testArray[i] > testArray[i + 1]) {
            printf("Тест провалено.\n");
            return;
        }
    }
    printf("Тест пройдено успішно.\n");
}
```

Рис. 7. Згенерована ШІ функція для перевірки сортування масиву

Завдання для студента: розробити власні тести для інших функцій, таких як пошук мінімального елемента чи підрахунок суми елементів.

Ця програма демонструє базову ідею тестування функцій. Вона легко модифікується для інших цілей, зокрема для: *перевірки правильності математичних функцій* (розрахунок факторіалів, піднесення до степеня, знаходження кореня числа, обчислення середнього, мінімального або максимального значення в масиві); *автоматизації тестування логічних функцій* (перевірка, чи є число простим, знаходження найбільшого спільного дільника (НСД), перевірка парності числа); *роботи з рядками* (підрахунок довжини рядка, перетворення рядка у верхній або нижній регістр, перевірка, чи є рядок паліндромом); *тестування масивів* (сортування масиву, пошук елемента в масиві, перевірка, чи є масив симетричним)

Крім технічних аспектів, використання ШІ у навчанні сприяє розвитку важливих м'яких навичок, таких як критичне мислення та творчий підхід до вирішення задач [4]. Студенти під час роботи з кодом навчаються критично оцінювати його структуру та логіку. Вони повинні розуміти, як працює кожен рядок коду, яка його мета та чи правильно він взаємодіє з іншими частинами програми. Завдяки такому аналізу студенти розвивають уважність до деталей, розуміння алгоритмів і логічне мислення.

Використання ШІ дозволяє студентам експериментувати з різними підходами до розв'язання задач [10]. Наприклад, ШІ може запропонувати кілька різних способів обробки даних, а студент обирає оптимальний. Це мотивує студентів до пошуку нестандартних рішень і глибшого аналізу задач.

Отже, переваги впровадження ШІ у навчальний процес:

1. Автоматизація перевірки завдань. ШІ швидко аналізує код, виявляє помилки та надає пояснення, що економить час викладачів і забезпечує студентам миттєвий зворотний зв'язок.
2. Персоналізація навчання. Завдяки адаптивним алгоритмам ШІ налаштовує рівень складності завдань під кожного студента.
3. Генерація варіативних завдань. ШІ дозволяє створювати різні варіанти однієї задачі, що сприяє зменшенню ризику списування.
4. Розвиток практичних навичок. Студенти отримують можливість розвивати навички оптимізації, аналізу та творчого підходу.

Висновки та перспективи дослідження. Мова програмування Сі залишається важливим елементом ІТ-освіти завдяки своїй універсальності, ефективності та низькорівневим можливостям. Її вивчення формує фундаментальні навички, необхідні для розуміння принципів програмування та роботи з сучасними мовами. Водночас впровадження ШІ у процес навчання мови Сі відкриває нові можливості для вдосконалення освітнього процесу.

ШІ дозволяє автоматизувати рутинні завдання, такі як перевірка коду, оцінювання ефективності алгоритмів і генерація тестів. Це не лише підвищує якість навчання, а й звільняє час викладачів для фокусування на більш складних аспектах викладання. Крім того, персоналізація навчання за допомогою адаптивних алгоритмів сприяє глибшому засвоєнню матеріалу, надаючи кожному студенту завдання, що відповідають його рівню підготовки.

Особливо цінним є внесок ШІ у розвиток критичного мислення та творчого підходу. Завдяки аналізу помилок, оптимізації алгоритмів і пошуку нестандартних рішень студенти вчаться критично оцінювати код, розуміти його структуру та пропонувати інноваційні ідеї. Інтерактивні завдання, створені з використанням ШІ, допомагають студентам не лише засвоювати технічні знання, але й формувати важливі практичні навички.

Таким чином, впровадження ШІ у навчальний процес дисципліни «Основи програмування мовою Сі» відкриває значний потенціал для покращення якості освіти, підвищення ефективності навчання та підготовки студентів до сучасних викликів у галузі технологій.

Подальші дослідження можуть бути спрямовані на створення інтерактивних платформ та розробку методик, які дозволять максимально ефективно інтегрувати ШІ у процес навчання програмуванню. Особливу увагу слід приділити використанню ШІ для аналізу прогресу студентів, виявлення їхніх індивідуальних потреб і надання персоналізованих рекомендацій, що сприятиме ще більш ефективному засвоєнню матеріалу та розвитку професійних компетенцій.

Список використаних джерел:

1. Afshan Younas, Subramanian Kabaly, Al-Haziati Mohammed, Hussainy Syed, Kindi Ahmed. A Review on Implementation of Artificial Intelligence in Education. *International Journal of Research and Innovation in Social Science*. 2023. VII. С. 1092–1100. URL: https://www.researchgate.net/publication/373983762_A_Review_on_Implementation_of_Artificial_Intelligence_in_Education (date of access: 24.04.2025)

2. Babajanov M. R., Ishniyazov O. O. Use of Innovative Technologies to Learn Programming. *Journal of Science, Research and Teaching*. 2024. Vol. 3, No. 6. URL: <https://jsrt.innovascience.uz/index.php/jsrt/article/view/564/469> (date of access: 24.04.2025)
3. Huang Chien-Hsing. Programming Teaching in the Era of Artificial Intelligence. *Eximia Journal*, 2024. Vol. 13, С. 583–589, January. URL: <https://eximiajournal.com/index.php/eximia/article/view/488/325> (date of access: 24.04.2025)
4. Джонатан Хейбер. Критичне мислення. 2023. Видавництво: ArtHuss. 198 с.
5. Коваль О. Використання штучного інтелекту в процесі навчання програмування в умовах змішаного навчання. *Відкрите освітнє е-середовище сучасного університету*. 2024. № 5. С. 1–15. URL: <https://openedu.kubg.edu.ua/journal/index.php/openedu/article/view/543> (дата звернення: 24.04.2025)
6. Литвинова С., Рашевська Н., Проскура С. The use of artificial intelligence in teaching students programming languages. *Proceedings of the IX International Workshop on Professional Retraining and Life-Long Learning using ICT (3L-Person 2024)*. 2024. С. 10–29. URL: <https://ceur-ws.org/Vol-3781/paper01.pdf> (date of access: 24.04.2025)
7. Ma B., Chen L., Konomi S. Enhancing Programming Education with ChatGPT: A Case Study on Student Perceptions and Interactions in a Python Course. arXiv preprint arXiv:2403.15472. 2024. URL: <https://arxiv.org/abs/2403.15472> (date of access: 24.04.2025)
8. Отрошко Т., Школа О., Калініченко О. Впровадження штучного інтелекту в освітній процес магістрантів-інформатиків. *Distance Education in Ukraine: Innovative, Normative-Legal, Pedagogical Aspects*. 2024. С. 351–356. URL: https://www.researchgate.net/publication/383560632_VPROVADZENNA_STUCNOGO_INTELEKTU_V_OSVITNIJ_PROCES_MAGISTRANTIV-INFORMATIKIV (дата звернення: 24.04.2025)
9. Phung T., Pădurean V.-A., Cambroner J., Gulwani S., Kohn T., Majumdar R., Singla A., Soares G. Generative AI for Programming Education: Benchmarking ChatGPT, GPT-4, and Human Tutors. arXiv preprint arXiv:2306.17156. 2023. URL: <https://arxiv.org/abs/2306.17156> (date of access: 24.04.2025)
10. Яценко О. Технології штучного інтелекту: основні напрями впровадження в освіту. *Наукові записки. Серія: Педагогічні науки*. 2023. № 3. С. 378–382. URL: <https://eprints.zu.edu.ua/37808/1/Yatsenko.pdf> (дата звернення: 24.04.2025)