

вимагає не лише знань, а й уміння ефективно представляти та використовувати інформацію.

Створення вебресурсів має значний позитивний вплив на розвиток таких особистісних якостей учнів, як самовираження та навички структурованого мислення. Учні, які розробляють власні вебсайти та інші цифрові ресурси, значно покращують свої комунікаційні та організаційні навички. Вони вчаться не лише чітко висловлювати свої ідеї, але й адаптувати їх до різних форматів та аудиторій. Цей досвід стане у пригоді в майбутньому, незалежно від обраної професії. Підсумовуючи, інтеграція створення вебресурсів у навчальний процес позитивно впливає на розвиток структурованого мислення в учнів та готує їх до викликів сучасного суспільства.

Важливо також включати інтерактивні елементи у вебресурси. Наприклад, проекти, в яких учні створюють опитування або вікторини на вебсайті, не тільки перевіряють знання своїх однолітків, але й дозволяють отримати зворотний зв'язок щодо викладеного матеріалу, що розвиває комунікативні навички та вміння працювати в команді.

Створення вебматеріалів також позитивно впливає на розвиток особистісних якостей учнів. Вони розвивають навички, які є важливими для їх особистого та професійного розвитку. Вчителям важливо пропонувати методичні підходи, які інтегрують створення вебресурсів в освітній процес, такі як проекти, дослідження тощо, які мобілізують і стимулюють творчість.

### Список використаних джерел

1. Chen P.-S. D., Lambert A. D., Guidry K. R. Engaging online learners: The impact of Web-based learning technology on college student engagement. *Computers & Education*, 2010. Vol. 54, No. 4. P. 1222–1232. URL: <https://doi.org/10.1016/j.compedu.2009.11.008> (дата звернення: 06.11.2024).
2. Dr. Safia Urooj, Prof. Dr. Muhammad Shahid Farooq. Impact of Students' Ubiquitous Learning through Web 2.0 Tool on Students' 21st Century Skills: Critical Thinking and Collaboration. *sjesr.*, 2023. Vol. 6, No. 3. P. 29–43. URL: [https://doi.org/10.36902/sjesr-vol6-iss3-2023\(29-43\)](https://doi.org/10.36902/sjesr-vol6-iss3-2023(29-43)) (дата звернення: 06.11.2024).
3. Hadjerrouit S. Developing Web-Based Learning Resources in School Education: A User-Centered Approach. *Interdisciplinary Journal of e-Skills and Lifelong Learning*, 2010. Vol. 6. P. 115–135. URL: <https://doi.org/10.28945/1172> (дата звернення: 06.11.2024).

## ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ПРИ ВИВЧЕННІ ПРОГРАМУВАННЯ

**Дзюбата Наталія Миколаївна**

викладач комп'ютерних дисциплін,

Відокремлений структурний підрозділ «Тернопільський фаховий коледж Тернопільського національного технічного університету імені Івана Пулюя»,  
natalrost98@gmail.com

Зважаючи на швидкий прогрес у сфері інформаційних технологій та підвищені вимоги до підготовки фахівців у сфері програмування, необхідність у сучасних методах навчання постійно зростає. Важливо студентам не тільки надати відповідні знання та навички, але й навчити їх використовувати інструменти, які є невід'ємною частиною професійної діяльності IT-фахівців. Адже, кожен студент має різний рівень підготовки та швидкість засвоєння матеріалу, а це потребує індивідуального підходу до навчання.

Одним із інструментів, який може допомогти краще зрозуміти структуру та алгоритми програми є використання штучного інтелекту (ШІ) при вивченні

програмування. Він здатний аналізувати великі обсяги даних, пропонувати оптимальні способи організації коду для підвищення його ефективності, дозволяє навчати більш інтерактивно та цікаво.

Великий потенціал для цього має GitHub Copilot – інструмент, заснований на штучному інтелекті, який пропонує рекомендації з написання коду в реальному часі. Він інтегрується з Visual Studio, тому дає можливість краще зрозуміти користувачу основні концепції програмування, синтаксис, налагодження, довідку коду.

З допомогою вбудованого чату Copilot Chat у Visual Studio існує можливість проми і бачити відповіді безпосередньо в коді у вікні редактора. GitHub Copilot пропонує автоматичні підказки та фрагменти коду, що дозволяє студентам швидше знаходити рішення, а не витратити час на написання базового коду. При розробці проєкту можна також використовувати коментарі для формулювання завдань.

Ще одним можливим інноваційним підходом є використання Copilot у командних проєктах, що надає можливість згенерувати читабельний та зрозумілий код, який можна передати іншим учасникам команди для подальшої роботи.

Розглянемо, як саме це можна реалізувати. Перш за все, потрібно розподілити завдання між усіма учасниками проєкту. Кожен писатиме свою версію коду з підказками від Copilot та додавати коментарі для пояснення логіки програми. Потім учасники обмінюються ним для взаємного ревізю. Кожен аналізує код свого колеги, даючи один одному зворотний зв'язок з допомогою GitHub Copilot Chat, вивчає їхні підходи до вирішення проблем і дає рекомендації щодо можливих змін та покращень. Кожен може залишати коментар до конкретних рядків коду, запропонувати зміни, зробити зауваження або задати питання. Це досить важливий інструмент для отримання конструктивного зворотного зв'язку, вдосконалення своїх навиків і співпраці над кодом.

Після цього можна обговорити та проаналізувати запропоновані зміни. Такий формат дозволяє учасникам проєкту не лише переглядати та критикувати коди інших, а й аргументувати свої пропозиції, змушує замислитися про ефективність власного підходу.

По-друге, перевіряючи коди учасників, викладач також може надавати свої коментарі чи готові фрагменти кодів з GitHub Copilot щодо вдосконалення початкового коду, використовуючи при цьому оптимальні методи, стилі або підходи до програмування. Одночасно Copilot забезпечує дотримання єдиного стилю написання коду, що усуває деякі неточності та розбіжності між учасниками проєкту та автоматично може згенерувати тести та прискорити процес тестування. Завдяки цьому можна досягнути високої якості коду.

При оцінюванні такої проєктної роботи слід дотримуватися комплексного підходу, брати до уваги як індивідуальний внесок кожного студента, так і загальні результати команди. Потрібно враховувати наступні критерії: якість коду, конструктивність зворотного зв'язку, застосування рекомендацій, активність та взаємодію з іншими колегами, здатність самостійно приймати рішення щодо виправлення помилок у своєму та чужому коді.

Використання такого допоміжного інструменту як GitHub Copilot вдосконалює навички студентів, потрібні для командного програмування, спільну відповідальність за якість написання коду, вміння знаходити оптимальні

рішення та адаптуватися до різних стилів написання коду. Для команд-початківців із різним рівнем знань та досвіду Copilot може стати помічником при написанні кодів, цим самим надати можливість іншим зосередитись на складних завданнях.

Використання штучного інтелекту може істотно полегшити навчальний процес і зробити його більш ефективним. Він сприяє розвитку критичного мислення, підвищує якість підготовки фахівців у галузі інформаційних технологій. Водночас не потрібно забувати, що ШІ – це лише додатковий інструмент при написанні коду, і він не замінить глибокого розуміння основ програмування та критичного мислення. Використання ШІ допомагає прискорити роботу та полегшити певні завдання, але остаточні рішення і якість коду залежать від самого розробника.

### Список використаних джерел

1. Глибовець М. М, Олецкий О. В. Штучний інтелект : Київ, КМ Академія, 2002, 271 с.
2. Кокін М., Гнатченко Д. Використання штучного інтелекту для автоматизованого тестування програмного забезпечення *Інформаційні технології та кібербезпека в умовах воєнного часу*: матеріали студентської науково-практичної конференції (м. Київ, 18 квітня, 2024) Київ : Державний торговельно-економічний університет, 2024. С.100–105.
3. Штучний Інтелект в освітньому процесі. URL: [https://www.futureschool.online/post/artificial\\_intelligence](https://www.futureschool.online/post/artificial_intelligence) (дата звернення: 01.11.2024).
4. Tech Summer for Educators : AI Edition URL: <https://softserve.academy/course/info.php?id = 645> (дата звернення: 01.11.2024).

## ВИКЛИКИ ТА МОЖЛИВОСТІ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ВИЩОЇ ОСВІТИ

### Іваницький Роман Іванович,

кандидат технічних наук, асистент кафедри інформатики і методики її навчання  
Тернопільського національного педагогічного університету імені Володимира Гнатюка  
romik\_iv@ukr.net

### Ковальчук Ольга Ярославівна,

кандидат фізико-математичних наук, доцент кафедри теорії права та конституціоналізму  
Західноукраїнського національного університету  
olhakov@gmail.com,

У сучасному освітньому просторі відбувається стрімка трансформація, зумовлена впровадженням інноваційних технологій штучного інтелекту (ШІ), зокрема ChatGPT. Ця передова система, що базується на моделях глибокого навчання, здатна генерувати контент, який за своєю якістю та структурою наближається до людського, що відкриває нові горизонти для вищої освіти. ChatGPT пропонує широкий спектр можливостей: від автоматизації рутинних завдань до створення персоналізованих навчальних матеріалів та інтерактивних освітніх сценаріїв. Ця технологія потенційно може підвищити ефективність навчального процесу, розширити доступ до освіти та сприяти розвитку інноваційних педагогічних підходів. Проте, інтеграція такої потужної системи в академічне середовище породжує низку серйозних викликів та етичних дилем.

Генеративний штучний інтелект можна визначити як технологію, яка використовує моделі глибокого навчання для генерації контенту, подібного до людського (наприклад, зображень, слів) у відповідь на складні та різноманітні запити (наприклад, мови, інструкції, питання). Оскільки генеративний ШІ продовжує швидко розвиватися, у найближчі кілька років він стимулюватиме