

МЕТОДИЧНІ ОСОБЛИВОСТІ НАВЧАННЯ ПРОГРАМУВАННЯ В УМОВАХ ВИКОРИСТАННЯ ІНСТРУМЕНТІВ ШТУЧНОГО ІНТЕЛЕКТУ

Пришляк Юрій Андрійович

здобувач першого рівня вищої освіти спеціальності Середня освіта (Інформатика)
Тернопільський національний педагогічний університет імені Володимира Гнатюка
pryshlyak_ya@fizmat.tnpu.edu.ua

Барна Ольга Василівна

кандидат педагогічних наук, доцент кафедри інформатики та методики її навчання
Тернопільський національний педагогічний університет імені Володимира Гнатюка
barna_ov@fizmat.tnpu.edu.ua

Швидкий розвиток інформаційних технологій та широке впровадження систем штучного інтелекту глибоко вплинули на всі аспекти суспільного життя, включаючи освіту. В останні роки інструменти штучного інтелекту стали невід'ємною частиною процесу розробки програмного забезпечення [2]. З'явилися інтелектуальні помічники, здатні генерувати програмний код, інтерпретувати алгоритми, знаходити помилки програмного забезпечення та допомагати розробникам вирішувати складні технічні проблеми. Такі системи, як GitHub Copilot, ChatGPT та Codeium, змінюють сприйняття людьми процесу програмування та ролі програмістів у розробці програмних продуктів. Тому вимоги до підготовки фахівців у сфері інформаційних технологій постійно змінюються [1].

Традиційне навчання програмуванню в першу чергу зосереджене на розвитку алгоритмічного мислення, вивченні синтаксису мови програмування та самостійному написанні коду. Однак сучасні інструменти штучного інтелекту значною мірою автоматизували ці процеси. Це створює нові виклики для систем освіти: їм потрібно не лише навчити учнів писати код, але й розвивати їхню здатність ефективно взаємодіяти з інтелектуальними системами. Тому вчителі стикаються з проблемою переосмислення методів навчання програмуванню та інтеграції інструментів штучного інтелекту в навчальний процес [4].

Це питання особливо актуальне у вищих навчальних закладах, де готують майбутніх програмістів. Освітні програми повинні враховувати сучасні технологічні тенденції та потреби ринку праці [3]. Сучасні програмісти повинні не лише володіти мовами програмування, а й розуміти принципи роботи систем штучного інтелекту та вміти використовувати їх як інструменти для підвищення ефективності своєї роботи. Тому дослідження методологій навчання програмуванню в контексті широкого застосування інструментів штучного інтелекту є особливо важливим.

Освіта програмування має довгу історію та пройшла кілька етапів. На ранніх етапах комп'ютерної освіти основна увага приділялася вивченню мов програмування низького рівня та розвитку базових алгоритмічних навичок. Пізніше, з появою мов високого рівня та об'єктно-орієнтованого програмування, акцент перемістився на розвиток логічного та алгоритмічного мислення учнів [4]. У сучасний час вирішальним також став розвиток навичок командної роботи, використання систем контролю версій, тестування програмного забезпечення та застосування сучасних інструментів розробки.

Поява генеративних систем штучного інтелекту суттєво вплинула на процес програмування. Сучасні асистенти на основі штучного інтелекту можуть автоматично генерувати фрагменти коду на основі текстових описів проблем, пропонувати оптимальні алгоритмічні рішення, виявляти помилки в програмах та надавати пропозиції щодо виправлення [1]. Це відкриває нові можливості для освіти. З одного боку, використання таких інструментів може допомогти учням швидше опанувати складні поняття та отримувати пояснення в режимі реального часу. З

іншого боку, надмірна залежність від штучного інтелекту може призвести до зниження самостійності учнів та недостатнього розвитку фундаментальних знань.

У зв'язку з сучасними тенденціями зростає актуальність створення інноваційних методичних підходів до викладання програмування. Одним із перспективних напрямів є інтеграція інструментів штучного інтелекту у навчальний процес як допоміжного засобу [3]. Зокрема, викладачі можуть залучати інтелектуальних помічників для демонстрації різних способів вирішення однієї задачі або детального аналізу типових помилок у коді [4]. Студенти, зі свого боку, здатні використовувати ці інструменти для перевірки власних рішень, отримання допоміжних підказок чи пошуку нових алгоритмічних підходів.

Ключовим аспектом у навчанні програмуванню залишається розвиток критичного мислення. Студенти мають навчитися не лише володіти ШІ-інструментами, але й об'єктивно оцінювати результати їхньої роботи. Оскільки штучний інтелект може пропонувати некоректні або неоптимальні рішення, майбутні ІТ-спеціалісти повинні вміти проводити ретельний аналіз коду, який генерує система. Такий підхід сприяє поглибленню розуміння основ програмування та підвищує рівень підготовки фахівців.

Одне з ключових питань, яке постає у зв'язку з використанням ШІ в навчанні, — це оцінювання досягнень студентів. Застосування генеративних інструментів для створення програмного коду ускладнює перевірку рівня самостійності у виконанні завдань [2]. Тому традиційні форми контролю знань варто доповнювати іншими методами, як-от усний захист проєктів, виконання практичних завдань в аудиторії чи впровадження проєктно-орієнтованого підходу до навчання.

Проєктне навчання в умовах використання штучного інтелекту є одним із найефективніших підходів до викладання програмування. У процесі роботи над проєктами студенти отримують можливість застосовувати теоретичні знання на практиці, співпрацювати у командах і вирішувати реальні задачі зі створення програмного забезпечення. Штучний інтелект може виступати помічником для генерування ідей, оптимізації алгоритмів або аналізу написаного коду [3, 5]. Однак фінальна відповідальність за якість створеного продукту залишається за студентами, що стимулює їх до самостійного мислення та відповідального підходу.

Окрім технічних аспектів, важливо формувати у студентів розуміння етичного використання ШІ-технологій. У сучасному цифровому середовищі особливої уваги потребують питання академічної доброчесності, дотримання авторських прав і відповідального ставлення до технологій [4]. Майбутні програмісти мають усвідомлювати обмеження застосування інструментів штучного інтелекту: ці технології є лише допоміжним засобом і не можуть замінити творчість, критичне мислення та професійні знання людини.

Впровадження штучного інтелекту в освітній процес відкриває широкі можливості для вдосконалення викладання програмування. Проте для ефективної реалізації цього підходу необхідне переосмислення існуючих педагогічних методів та розробка сучасних форматів оцінювання знань студентів [1]. Це дозволить долучити новітні технології до практики навчання, зберігаючи при цьому високі стандарти освіти.

Розвиток технологій штучного інтелекту помітно впливає на сучасну систему підготовки фахівців у сфері програмування. Інтелектуальні інструменти змінюють підхід до розробки програмного забезпечення, пропонуючи нові перспективи для навчання студентів. У нинішніх умовах важливо не лише навчити майбутніх програмістів створювати код, але й розвивати їхні навички ефективної взаємодії з системами штучного інтелекту, аналізу їхньої роботи та відповідального використання цих технологій.

Інтеграція засобів штучного інтелекту у навчальний процес сприяє підвищенню зацікавленості студентів, стимулює творче мислення та допомагає формувати сучасні професійні компетенції [3]. Однак при цьому необхідно дотримуватися балансу між використанням автоматизованих інструментів і забезпеченням глибокого освоєння основ програмування. Подальші дослідження в цьому напрямі повинні бути спрямовані на створення ефективних методичних моделей навчання програмуванню, які враховують як можливості, так і обмеження сучасних систем штучного інтелекту.

Список використаних джерел

1. Holmes W., Bialik M., Fadel C. Artificial intelligence in education: promises and implications for teaching and learning. Boston: Center for Curriculum Redesign, 2019. 46 p.
2. Luckin R., Holmes W., Griffiths M., Forcier L. B. Intelligence unleashed: an argument for AI in education. London: Pearson Education, 2016. 44 p.
3. Russell S., Norvig P. Artificial intelligence: a modern approach. 4th ed. Pearson, 2021. p. 1136.
4. Sommerville I. Software engineering. 10th ed. Boston: Pearson Education, 2016. 816 p.
5. Wing J. M. Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 2008. Vol. 366. P. 3717–3725.

ВИКОРИСТАННЯ ГЕНЕРАТИВНИХ МОДЕЛЕЙ ШТУЧНОГО ІНТЕЛЕКТУ ЯК ЗАСОБУ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ОСВІТНЬОГО ПРОЦЕСУ

Романенко Тетяна Василівна

доктор педагогічних наук, професор кафедри автоматизації та комп'ютерно-інтегрованих технологій

Черкаський національний університет імені Богдана Хмельницького
tan.romanenko25@gmail.com

Бодненко Світлана Дмитрівна

здобувач другого рівня вищої освіти спеціальності Середня освіта (за предметними спеціальностями)

Черкаський національний університет імені Богдана Хмельницького
bodnenko.svitlana1621@vu.cdu.edu.ua

Попри ризики використання штучного інтелекту в освіті, учасники навчального процесу активно використовують цифрові технології у різних видах діяльності. Значна увага у навчальному процесі приділена генеративним моделям, які здатні створювати тексти, зображення, програмні коди та інші види контенту. Завдяки ним відкриваються нові можливості для організації навчання здобувачів освіти. Однак, виникають певні виклики: методичні, педагогічні та інші. Тому, потреба у розробленні ефективних методичних підходів до впровадження генеративного ШІ, які забезпечать підвищення якості освіти й розвиток аналітичного мислення та цифрової грамотності здобувачів освіти.

Проблематика використання ШІ в освіті активно досліджується вітчизняними та зарубіжними науковцями. Реорганізація методичних підходів до впровадження генеративних моделей ШІ в освітній процес нині є актуальною [3].

Одне з важливих місць серед цифрових технологій займають генеративні моделі ШІ, які здатні створювати новий контент на основі аналізу великих масивів даних, що відкриває широкі перспективи для їх застосування у сфері освіти [1].

Генеративний штучний інтелект – категорія ШІ, що здатний до створення нового контенту (тексту, зображення чи аудіо), побудований на аналізі великих