

Список використаних джерел

1. Білик Т., Акименко А. Бінарні відношення. Транзитивність. 2023.
2. Artificial Intelligence and Mathematics Teaching (n.d.). National Council of Teachers of Mathematics. URL: <https://www.nctm.org/standards-and-positions/Position-Statements/Artificial-Intelligence-and-Mathematics-Teaching/> (available at: 25.03.2025).
3. Degni, F. AI in Education : The Impact of Artificial Intelligence on Education, 2024.
4. How AI and Human Teachers Can Collaborate to Transform Education. (2025, January). World Economic Forum. URL: <https://www.weforum.org/stories/2025/01/how-ai-and-human-teachers-can-collaborate-to-transform-education/> (available at: 25.03.2025).
5. Coscia A., Mathew S., Vemmou M. The Impact of Transitivity, Similarity and Implicit Bias on Trust in AI.

ФОРМУВАННЯ АЛГОРИТМІЧНОГО МИСЛЕННЯ УЧНІВ 7–9 КЛАСІВ ЗАСОБАМИ МОВ ПРОГРАМУВАННЯ

Валігура Михайло Ігорович

здобувач другого (магістерського) рівня вищої освіти спеціальності Середня освіта
(Інформатика),

Тернопільський національний педагогічний університет імені Володимира Гнатюка,
vmisha200113@gmail.com

Мартинюк Сергій Володимирович

кандидат фізико-математичних наук, доцент кафедри інформатики та методики її навчання,
Тернопільський національний педагогічний університет імені Володимира Гнатюка,
sergmart65@tnpu.edu.ua

Ми живемо в час, коли технології розвиваються шаленими темпами, і майже все навколо – від банального онлайн-замовлення їжі до керування розумним будинком – пов'язане з алгоритмами. Саме тому навичка алгоритмічного мислення вже давно перестала бути привілеєм програмістів – вона стала необхідністю для кожної сучасної людини.

Важливо розвивати цю навичку у школярів, адже учні 7–9 класів – це діти, у яких якраз активно формується вміння логічно мислити, аналізувати, будувати причинно-наслідкові зв'язки. І саме зараз – найкращий момент, щоб навчити їх не боятися складних задач, а вміти бачити структуру проблеми й крок за кроком знаходити шлях до її розв'язання [1]. Особливо корисними для цього є мови програмування – вони дозволяють застосовувати абстрактні алгоритмічні поняття на практиці. Але попри це, все ще існує проблема – дієвість методів навчання програмування часто залишає бажати кращого.

Використання мов програмування часто асоціюється в учнів із труднощами у сприйнятті синтаксису та логічної побудови програм, що може викликати втрату мотивації до подальшого вивчення предмету. Таким чином, актуальним є питання пошуку оптимальних підходів до формування алгоритмічного мислення засобами мов програмування, що забезпечить підвищення рівня підготовки школярів у цій сфері [2].

Одним із сучасних і справді дієвих способів розвитку алгоритмічного мислення у школярів є метод «програмування через експерименти». Його суть

проста й зрозуміла: учням пропонують працювати в інтерактивному середовищі, де вони можуть змінювати частини коду й одразу бачити, що з цього виходить. Такий формат уроку дозволяє відійти від нудного зазубрювання правил і натомість дати можливість дітям самим досліджувати, як саме працює програма.

Найголовніше в цьому методі – миттєвий зворотний зв'язок. Учень щось змінює в коді – і тут же бачить результат. Наприклад, у середовищах на базі Python або JavaScript можна налаштувати візуалізацію, де будь-які зміни в параметрах алгоритму показуються у вигляді графіків, анімацій чи поведінки якихось віртуальних об'єктів. Це не тільки зручно, а ще й весело: діти буквально «граються» з кодом, експериментують, змінюють умови – і завдяки цьому поступово починають краще розуміти внутрішню логіку програм [3].

Такий підхід має багато плюсів. По-перше, він допомагає розвивати аналітичне мислення: учні вчаться помічати закономірності, робити висновки, шукати оптимальні рішення. По-друге, навчання перетворюється на захопливий процес. Динаміка, візуальні ефекти, швидкий результат – усе це справді тримає увагу й викликає інтерес до теми.

Крім того, зникає страх перед помилками: учень бачить, що щось не працює, одразу пробує інший варіант – і вчиться на власному досвіді.

Втім, є і певні труднощі. Такий підхід вимагає технічного оснащення: потрібні комп'ютери, стабільний доступ до інтернету, відповідне програмне забезпечення. Також учителі повинні витратити час і сили на створення інтерактивних завдань, що не завжди просто й швидко.

Реалізувати метод можна так [4]:

Використовувати онлайн-платформи для кодування – Replit, Jupyter Notebook, p5.js – де все відбувається в реальному часі.

Створювати навчальні матеріали з прикладами, у яких можна змінювати параметри алгоритмів і бачити результат.

Давати учням завдання, які передбачають самостійне дослідження алгоритмів: змінювати код, тестувати варіанти, аналізувати результат.

Оцінювати потрібно не лише фінальний результат, а й сам процес – хід мислення, пошук рішень, логіку експериментів.

Щоб краще пояснити, як саме працює метод «програмування через експерименти», уявімо звичайний шкільний урок, присвячений темі алгоритмів сортування. Але замість того, щоб просто переписувати код із підручника чи слухати пояснення вчителя, учні отримують доступ до інтерактивного симулятора. У ньому можна змінювати вхідні дані – наприклад, впорядковувати різні набори чисел – і в реальному часі спостерігати, як саме працює той чи інший алгоритм.

Спочатку учні запускають «бульбашкове» сортування і бачать, як числа повільно «спливають» у правильному порядку. Потім перемикаються на «швидке» сортування й одразу помічають: процес іде інакше, більш ефективно. Після цього вчитель пропонує учням не просто дивитися, а самостійно змінити фрагменти коду: наприклад, змінити умови обміну елементів або впровадити підрахунок кількості операцій. І ось тут починається найцікавіше – справжній елемент дослідження. Діти експериментують, помиляються, пробують ще раз і, зрештою,

самостійно доходять висновків: який алгоритм працює швидше, чому і в яких випадках.

Це вже не просто урок, а майже лабораторна робота з програмування – жива, цікава й наочна. Учні не просто слухають теорію, вони «проживають» її. А ще – вчаться мислити, аналізувати й доводити власну точку зору на основі конкретних прикладів.

Метод «програмування через експерименти» – це не просто новація, а реальний спосіб зробити уроки інформатики цікавішими й ефективнішими. Він допомагає учням краще зрозуміти логіку програм, вчить аналітичного мислення й підвищує зацікавленість у предметі. Попри певні виклики, пов'язані з технічним забезпеченням та підготовкою матеріалів, такий підхід може істотно підвищити якість навчання алгоритмізації й програмування. У майбутньому варто зосередитися на створенні готових методичних матеріалів, які полегшать учителям впровадження цієї моделі в освітній процес.

Список використаних джерел

1. Бондаренко С. М., Мельник О. І. Формування алгоритмічного мислення в учнів основної школи засобами програмування. *Комп'ютер у школі та сім'ї*, 2022. № 2(158). С. 3–8.
2. Лапінський В. В., Рижова І. М. Програмування в школі сучасні тенденції та практичні аспекти. *Інформатика та інформаційні технології в навчальних закладах*, 2023. № 1. С. 14–18.
3. Пітенко Ю. В., Савченко А. П. Методика навчання програмування у базовій школі: інноваційні підходи. *Освіта та розвиток обдарованої особистості*, 2021. 3(86). С. 27–32.
4. Комісаренко Т. М. Програмування як засіб розвитку логічного та алгоритмічного мислення. *Інформаційні технології і засоби навчання*, 2020. № 5(79). С. 120–133. URL: <https://journal.iitta.gov.ua/index.php/itlt/article/view/3600> (дата звернення: 03.04.2025).

МЕТОД ПРОЄКТІВ ЯК ЗАСІБ РЕАЛІЗАЦІЇ STEM-ОСВІТИ

Гетманюк Оксана Іванівна

асистент кафедри математики та методики її навчання,
Тернопільський національний педагогічний університет імені Володимира Гнатюка,
getmanuyk_oksana@tnpu.edu.ua

Сучасний тренд «STEM-освіта» – це концепція інтегрованого навчання учнів за чотирма профільними дисциплінами (Science – наука, Technology – технології, Engineering – інженерія, Mathematics – математика) в міждисциплінарному та прикладному контексті – є надзвичайно актуальним феноменом в аспекті стратегічного розвитку провідних країн світу. Саме STEM-освіта сприяє цілісному сприйняттю світу та показує прикладне спрямування отриманих знань з окремих предметів для вирішення однієї проблеми. Сам підхід дозволяє розвинути особистість різносторонньо і забезпечує високий науковий потенціал держави [2].

Метод проєктів – це педагогічний підхід, який орієнтований на активну діяльність учнів у процесі створення конкретного продукту або вирішення практичної задачі. Цей метод ідеально поєднується з концепцією STEM (Science, Technology, Engineering, Mathematics), оскільки обидва підходи базуються на