

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

*Інститут педагогіки Національної академії педагогічних наук України  
Інститут цифровізації освіти Національної академії педагогічних наук України  
Національний університет «Чернігівський колегіум» імені Т.Г. Шевченка  
Вінницький державний педагогічний університет імені Михайла Коцюбинського  
Тернопільський національний педагогічний університет імені В. Гнатюка  
Université Côte d'Azur (France)  
Ліцей «ДОМІНАНТА» міста Києва  
Києво-Печерський ліцей № 171 «ЛІДЕР»  
Український державний університет імені Михайла Драгоманова  
Кафедра інформаційних технологій і програмування*

**МАТЕРІАЛИ**

*II Всеукраїнської науково-практичної конференції*

**ТЕОРІЯ І ПРАКТИКА  
ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
В УМОВАХ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ ОСВІТИ**



**КИЇВ – 2024**



УДК 37.014-044.922:004(082)

Т33

*Рекомендовано до друку Вченою радою  
Українського державного університету імені Михайла Драгоманова  
(протокол № 11 від 27 червня 2024 р.)*

**Т33** Теорія і практика використання інформаційних технологій в умовах цифрової трансформації освіти: *матеріали II Всеукраїнської науково-практичної конференції, 19 – 20 червня 2024 року м. Київ / Упорядник: Твердохліб І.А. Київ: Вид-во УДУ імені Михайла Драгоманова, 2024. – 242 с. - електронне видання.*

Збірник містить матеріали доповідей учасників II Всеукраїнської науково-практичної конференції «Теорія і практика використання інформаційних технологій в умовах цифрової трансформації освіти».

Доповіді присвячені методичним аспектам використання сучасних інформаційних технологій в освітньому процесі, проблемам модернізації змісту інформатичної середньої та вищої освіти в умовах цифрової трансформації суспільства, особливості впровадження STEAM в освітній процес. Розглянуто актуальні в даний час питання використання штучного інтелекту в освітньому процесі, досвід і перспективи цифровізації освіти України.

*Матеріали подано в авторській редакції*

## МЕТОДИЧНІ ПРОБЛЕМИ НАВЧАННЯ ПРОГРАМУВАННЯ МАЙБУТНІХ ПЕДАГОГІВ ПРОФЕСІЙНОЇ ОСВІТИ

*Назаренко Віктор Степанович,*

*доцент кафедри інформаційних систем і технологій, кандидат педагогічних наук  
Український державний університет імені Михайла Драгоманова, м. Київ, Україна*

Навчання програмування традиційно є одним із найскладніших напрямів із погляду методики викладання. Причому ця тенденція спостерігається як у загальноосвітній школі, так і у сфері середньої та вищої освіти. На сьогодні можна констатувати, що у ЗВО приходить значна кількість студентів, які мають невисокий рівень компетентності в галузі програмування, а деякі взагалі не володіють шкільним рівнем програмування та надалі не здатні відповідати вимогам вишу під час навчання програмування. Причому дуже низькі результати виконання мають завдання на алгоритмізацію і програмування, де найнижчий результат спостерігається при виконанні завдань на розуміння та аналіз готових алгоритмів.

Наш досвід викладання дисципліни «Алгоритми та структури даних», яка вивчається на першому курсі показує, що студенти досить успішно вміють вирішувати завдання, діючи за зразком. Прикладом, виконання таких завдань є завдання з вимогою вміння читати і виконувати програми мовою програмування, або завдання на вміння написати коротку (10–15 рядків) просту програму мовою програмування або записати алгоритм природною мовою. Якщо формулювання такого завдання трохи змінюється, та його рішення передбачає глибоке розуміння роботи системи програмування, то відсоток виконання різко погіршується.

Так, наприклад, при досить високому рівні знань про основні конструкції мови програмування та системи програмування та вміння читати та налагоджувати програми мовою програмування, спостерігається дуже низький рівень вміння аналізувати програму, що використовує різні конструкції мови програмування. Відповідно, якщо потрібно лише формально виконати алгоритм та отримати результат його роботи, то з таким завданням здатна впоратися доволі значна кількість студентів. Але відсоток студентів, здатних впоратися із завданням, де необхідно зрозуміти роботу алгоритму, проаналізувати можливі результати та вихідні дані, знайти помилки та виправити їх, суттєво знижується [3; 5].

Ще гірше студенти справляються з аналізом алгоритму, наприклад, із завданнями на вміння прочитати фрагмент програми мовою програмування та виправити допущені помилки або завданнями на аналіз алгоритму, що містить цикл та розгалуження, де необхідно вказати вхідні дані, які призведуть до конкретного результату. Все це говорить про відсутність глибокого розуміння механізмів написання програми деякою формальною мовою.

На наш погляд, одна з причин виникнення такої ситуації полягає у негнучкості застосування спектру різноманітних підходів під час навчання програмування. Звичною у навчанні програмування є демонстрація мови на прикладах простих програм із короткими коментарями [2; 4]. Цей принцип передбачає два методи навчання: демонстрація готових типових алгоритмів для простих програм із наступними коментарями чи покрокова демонстрація з паралельним поясненням принципів роботи типових алгоритмів. В обох випадках основна ідея однакова – студентом пропонується алгоритм (готове нове знання), у якому необхідно розібратися та користуватися ним у наступних, складніших завданнях.

o o

---

Досвід викладання мов програмування показує, що в сучасних умовах обидва ці методи не дуже ефективні, тому що належать до методів навчання, які задіюють лише репродуктивне мислення. Тому в більшості випадків студенти можуть вирішувати лише типові завдання з мінімальною заміною умови, що передбачає зміну знака, формули, числа, а складніші завдання вирішують уже важче.

Наведене вище дає підстави для розширення спектру підходів, що застосовуються у класичній методиці, яка давно зарекомендувала та виправдала себе. Одним із таких підходів може бути навчання програмування на основі суттєвого розширення когнітивної діяльності студента, тобто підхід, при якому алгоритм вирішення типового завдання не пропонується викладачем у готовому вигляді, а є продуктом самостійної мисленнєвої діяльності студента під контролем викладача.

Поняття когнітивної діяльності у вітчизняній педагогіці та психології не є новим. В основі когнітивної діяльності лежить когнітивна, або пізнавальна активність, яка спирається на суб'єктивний досвід людини [1]. Когнітивна діяльність - діяльність, в результаті якої людина приходить до певного рішення або отримує нове для себе знання. У процесі чого виникає розумова діяльність, що призводить до розуміння (інтерпретації) чогось [6]. Згідно з психологією, когнітивна діяльність безпосередньо пов'язана з поняттям мислення та сприяє розвитку пізнавальних процесів та потреб і мотиваційної сфери студентів, формуючи тим самим у них продуктивне мислення.

Таким чином, когнітивною діяльністю можна вважати такий вид діяльності, що спирається на пізнавальні здібності студента. До таких здібностей належать: логічні та смислові здібності, вміння правильно будувати питання (залежно від розв'язуваної задачі), прогнозувати результат, формулювати гіпотези, робити висновки, синтезувати отримані знання та ін. Головним принципом розвитку пізнавальних здібностей є первинність пізнання людиною реальності, а не вивчення готових знань про неї. Самостійне вивчення студентами одного і того ж реального освітнього об'єкта призводить до отримання індивідуальних освітніх продуктів – знань про цей об'єкт, а також способів їх отримання [1; 3].

Тому, для успішного вивчення програмування у вищій школі недостатньо оновити методику викладання програмування. Необхідно зробити програмування цікавим та неформальним, особисто потрібним але без втрати фундаментальності. Програмування має стати для студентів головним інструментом у розвитку обчислювального мислення – виду мислення, що визначає можливі варіанти постановки завдань та шляхи їх вирішення у формі, яка може бути ефективно реалізована за допомогою людини чи комп'ютера.

При організації когнітивної діяльності в процесі навчання програмуванню відбувається перехід від репродуктивного до продуктивного способу мислення, внаслідок чого збільшується ступінь новизни знання, що отримується в процесі розумової діяльності, відбувається природній розвиток інтелекту. Студент отримує більш глибоке розуміння та усвідомлення матеріалу, що вивчається.

Для успішної реалізації цього підходу необхідно слідувати низці принципів:

1. Навчання програмування має бути засноване переважно на самостійній діяльності. Так, типовий алгоритм на вирішення нового виду завдань має бути отриманий студентом самостійно (у парі, малій групі). Процес навчання має спиратися на інтелектуальний потенціал студента. Навчання досягне бажаних

цілей лише за організації правильного напрямку діяльності студента, що має для нього особистісний сенс.

2. Для досягнення результату необхідно використовувати активні методи навчання, а реалізація процесу навчання має відбуватися у співпраці. Викладач повинен використовувати системи спрямовуючих питань, допоміжних завдань, прикладів та контрприкладів, які дозволяють спрямовувати когнітивну діяльність студентів. Траєкторія навчання має бути ретельно продумана та побудована заздалегідь, щоб мати можливість швидко скоригувати діяльність студента.
3. Процес навчання програмування повинен відповідати наступним основним етапам навчання:
  - *актуалізація залишкових знань студентів.* Навчання буде продуктивним лише у разі опори на раніше засвоєний матеріал. Це можуть бути знання, отримані на заняттях при вивченні інформатичних, математичних, фізичних або інших дисциплін, залежно від тематики завдання, що вирішується;
  - *постановка мети та розуміння студентами пізнавальних завдань.* Цей етап характеризує перехід до активної когнітивної діяльності; здобуття студентами нового знання. Використовуються різні методичні прийоми задля досягнення необхідного результату: вправи, вирішення завдань, виконання проблемних та евристичних навчальних завдань;
  - *осмислення нового знання.* Відбувається у процесі активного обговорення та аналізу, отриманого знання. На цьому ж етапі відбувається корекція результату;
  - *первинне закріплення, формування нових умінь та навичок;*
  - *застосування знань.* Побудова зв'язку між отриманими знаннями та застосуванням їх на великому класі типових завдань.

Потрібно зазначити, що навчання програмування, організоване з урахуванням когнітивної діяльності, безпосередньо впливає розумовий розвиток та розвиток інтелекту, а організація когнітивної діяльності під час навчання програмуванню дуже трудомістка і вимагає від викладача глибоких знань у галузі програмування, методики та психології. Однак, наш досвід реалізації цього підходу при навчанні програмування свідчить, що активне включення до свідомої когнітивної діяльності дозволяє студентам ефективніше здобувати нові знання, формувати глибоке розуміння з теми, що вивчається, достатнє для аналізу програм, заснованих на вивченому типовому алгоритмі. Цей підхід дає студентам можливість самостійно усвідомити механізми алгоритмів, відчувати свій інтелектуальний потенціал, унаслідок чого бути успішнішими при подальшому вивченні наступних розділів програмування.

Насамкінець зауважимо, що реалізація такого підходу, спрямованого на розвиток особистості студента, його інтелектуальних і самоосвітніх здібностей, на формування продуктивних способів мислення та діяльності, умінь самостійної навчально-дослідницької роботи є ефективним як при навчанні програмуванню, так і в освіті загалом.

### *Список використаних джерел:*

1. Дуткевич Т. В. Загальна психологія. Теоретичний курс. [текст] навч. посіб. К.: Центр учбової літератури, 2016. 388 с.
2. Морзе Н. В. Методика навчання інформатики. Ч. 4. Методика навчання основ алгоритмізації та програмування. К. : Навчальна книга, 2004. 368 с.
3. Сейдаметова З. С., Манжос Л. О. Мови програмування в навчанні майбутніх програмістів. Науковий часопис НПУ імені М. П. Драгоманова. Серія 2. Комп'ютерно-орієнтовані системи навчання. 2010. Вип. 8 (15). С. 35–41.
4. Співаковський О. В. Алгоритмізація та програмування. Енциклопедичне видання : навчально-методичний посібник. К. : ТОВ Редакція «Комп'ютер», 2007. 128с.
5. Шевчук П. Г., Кривонос О. М. Значення стилю програмування в процесі навчання учнів та студентів. Комп'ютерно-інтегровані технології: освіта, наука, виробництво : Науковий журнал; Відп. ред. В. Д. Рудь. Луцьк, 2011. № 5 С. 148-150.
6. Cognitive networks: towards self-aware networks. Ed. by Qusay H.Mahmoud.: WileyIn-terscience, 2007. 368 p.

## **ДЕЯКІ ПИТАННЯ НАВЧАННЯ ОСНОВ СУЧАСНОЇ ІНФОРМАТИКИ МАЙБУТНІХ УЧИТЕЛІВ ІНФОРМАТИКИ**

*Нестерова Олена Дмитрівна,*

*старший викладач кафедри інформаційних технологій і програмування  
Український державний університет імені Михайла Драгоманова, м. Київ, Україна  
[o.d.nesterova@npu.edu.ua](mailto:o.d.nesterova@npu.edu.ua)*

*Струтинська Оксана Віталіївна,*

*професор кафедри інформаційних технологій і програмування,  
доктор педагогічних наук, професор  
Український державний університет імені Михайла Драгоманова, м. Київ, Україна  
[o.strutynska@udu.edu.ua](mailto:o.strutynska@udu.edu.ua)*

Освітні потреби початку ХХІ століття вимагають нових способів мислення, пов'язані з новими навчальними цілями та потребують оновлених підходів до навчання. Сьогоднішні учні та студенти кардинально відрізняються від учнів та студентів минулого, в першу чергу, завдяки своєму глибокому зануренню у цифрові технології. Усі аспекти їхнього життя пов'язані з глобальною цифровою трансформацією. Сучасні діти народилися та зростають у насиченому медіа та технологіями світі, для розв'язування більшості своїх інформаційних потреб вони використовують різноманітні гаджети, цифрові технології, віртуальні інструменти тощо [1, с. 1]. Освіта ХХІ століття має відповідати цим викликам. Одним із засобів їх подолання є формування у громадян цифрової компетентності, медіаграмотності та формування критичного мислення в питаннях використання сучасних цифрових технологій.

В Україні дисципліну «Інформатика» введено в освітній процес закладів загальної середньої освіти ще наприкінці 1980-х років на відміну від багатьох розвинених країн, що лише останні роки започатковують такий курс. З гордістю відзначимо провідних фахівців країни, які працювали на кафедрі основ інформатики та обчислювальної техніки Київського педагогічного інституту імені