

УДК 37.09:[37.016:004.421]

DOI: <https://doi.org/10.54662/veresen.1.2025.02>

**Наталія Махровська,**  
ORCID iD 0000-0001-9603-6902  
кандидат фізико-математичних наук,  
доцент кафедри теорії й методики  
природничо-математичної освіти  
та інформаційних технологій  
Миколаївський обласний інститут  
післядипломної педагогічної освіти  
вул. Адміральська, 4-а, 54001, м. Миколаїв, Україна  
[natalya.makhrovska@toippro.mk.ua](mailto:natalya.makhrovska@toippro.mk.ua)

**Ганна Погромська,**  
ORCID iD 0000-0002-6779-3995  
кандидат педагогічних наук, доцент,  
доцент кафедри теорії й методики  
природничо-математичної освіти  
та інформаційних технологій  
Миколаївський обласний інститут  
післядипломної педагогічної освіти  
вул. Адміральська, 4-а, 54001, м. Миколаїв, Україна  
[hanna.pohromska@toippro.mk.ua](mailto:hanna.pohromska@toippro.mk.ua)

## ПРАКТИКО-ТЕХНОЛОГІЧНІ АСПЕКТИ ВПРОВАДЖЕННЯ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ В ОСВІТНІЙ ПРОЦЕС

Стаття присвячена дослідженню перспектив упровадження основ алгоритмізації та програмування в шкільну освіту як інструменту розвитку структурованого мислення, важливого для формування ключових компетентностей XXI століття.

Особливу увагу приділено актуальності впровадження таких сучасних підходів до викладання програмування, як: гейміфікація, проєктне та персоналізоване навчання і співпраця. Висвітлено переваги кожного з підходів у мотивації учнів і підвищенні їхнього інтересу до вивчення програмування. Визначено критерії обирання мов програмування, що відповідають віковим особливостям учнів.

Підкреслено важливість підвищення кваліфікації педагогів у сфері програмування, запропоновано практичний курс «Старт у світ кодування: Python для початківців», спрямований на вдосконалення предметно-методичної та цифрової компетентності вчителів. Останній передбачає вивчення основ мови програмування Python, алгоритмічних структур та виконання практичних завдань, що сприяють розвиткові логічного мислення та використанню системного підходу до розв'язання проблем.

**Ключові слова:** алгоритмізація; алгоритмічне мислення; інформатика; методичні підходи; мови програмування; Python.

© Махровська Н. А., Погромська Г. С., 2025

**Вступ.** Сучасне суспільство все більш технологізується та оцифровується, що спричинює зростання попиту на навички алгоритмізації. Незалежно від того, чи йдеться про розроблення програмного забезпечення, веброзроблення або аналіз даних, мови програмування стають важливим інструментом як для бізнесу, так і

для приватних осіб. Знання основ кодингу зумовлює ефективний розвиток навичок ХХІ століття, що є актуальним для людини в сучасному світі. Однак фахівці вважають, що вивчення мов програмування є складним завданням, що часто є результатом сформованих стереотипів часів шкільної освіти. Традиційні методи викладання можуть бути неактуальними для сучасних учнів, що ускладнює засвоєння матеріалу. Водночас такі методи навчання не встигають за швидким розвитком комп'ютерних наук. Останнім часом з'являються нові ефективні підходи до освіти, що роблять програмування доступнішим і цікавішим для учнів.

Психолого-педагогічні дослідження Н. Брауна та Г. Вільсона свідчать про те, що викладання та навчання програмування є предметно-специфічною діяльністю (Brown N., Wilson G., 2018), оскільки має інший набір проблем і технік, ніж вивчення фізики, мови чи літератури. Інформатика є відносно молодого дисципліною порівняно з іншими шкільними предметами. Відповідно маємо менше досліджень та методичних напрацювань із її викладання.

Під час вивчення основ алгоритмізації та програмування ключовим є розвиток різних типів мислення, зокрема алгоритмічного. Представники багатьох освітніх систем визнали важливість останнього і навичок кодування та впроваджують зміни до навчальних програм для інтеграції програмування у формальну шкільну освіту. Необхідний і вкрай важливий фактор успіху передбачає підготовку та підтримку вчителів із цього питання. Готовність педагогів до сприйняття процесу кодингу перспективною мовою програмування із застосуванням сучасних освітніх методик є підґрунтям якісної підготовки здобувачів освіти.

**Аналіз останніх досліджень і публікацій.** Проблематику навчання основ алгоритмізації та програмування в шкільному курсі інформатики окреслюють дослідники Т. Кобильник, У. Когут, О. Сікора, В. Жидик

(Кобильник Т., 2022). Учені виокремлюють кілька ключових аспектів: рівень математичної підготовки учнів, забезпечення послідовності у викладанні змістової лінії «Алгоритмізація та програмування», вибір відповідної перспективної мови програмування відповідно до вікової категорії учнів, доступність навчально-методичних матеріалів і питання мотивації учнів. Особливості навчання програмування у своїх роботах розглядають Н. В. Морзе, В. В. Лапінський, П. Г. Шевчук, С. С. Жуковський, О. В. Коротун.

Питання вивчення інформатики у старшій школі досліджують автори статті (Юрченко А. О., Семеніхіна О. В., Хворостіна Ю. В., 2019) та приділяють увагу добиранню за певними критеріями мови програмування для навчання. У рамках навчального курсу базової школи, зазначає В. М. Базурін, доцільним є також і вибір відповідного середовища програмування (Базурін В. М., 2017).

На проблемі послідовності у вивченні програмування акцентують автори О. Власій, І. Яремій, М. Винничук (Власій О., Яремій І., Винничук М., 2021) та рекомендують використовувати MIT App Inventor як проміжний етап між вивченням програми Scratch і текстових мов програмування високого рівня (C++, Python, Java, C#).

Проблеми, пов'язані з навчанням програмування, з урахуванням психологічних аспектів та шляхів їх подолання розкриті в розвідці О. Семеніхіної, Ю. Руденко (Семеніхіна О., Руденко Ю., 2018).

У роботі Г. Бельмара (Belmar H., 2022) досліджено сучасний стан викладання програмування та обчислювального мислення на п'яти континентах. Автор зазначив, що формування алгоритмічного мислення в школах Англії було запроваджено у 2014 році, у Німеччині – з 2016 року на трансверсальному рівні в університетах, у Південній Кореї, Китаї та Тайвані – із 2016 року.

На думку Г. Каппеллі (Coppelli Ortiz G., 2018) протягом останнього десяти-

тиліття алгоритмічне мислення в базовій школі розвивається в різних країнах світу, що є стратегічним рішенням для технологічного прогресу та набуття навичок ХХІ століття. У сучасній системі освіти педагоги мають знати суспільні тенденції для інтегрування їх у процес навчання. Це спонукає освітян постійно аналізувати свою навчальну роботу, розвиватися та оновлювати зміст навчання учнів, а заклади вищої та післядипломної освіти вносити необхідні корективи до своїх програм підготовки вчителів, щоб урахувувати сучасні вимоги.

Уміння розв'язувати задачі, зокрема практико-орієнтовані та нестандартні, є одним з основних показників рівня розвитку творчого учня та глибини засвоєння навчального матеріалу. Під час вивчення інформатики незначну частину часу відведено на розв'язування задач. Важливим етапом є аналіз набутих результатів. Це стимул виявити можливі помилки або неточності та критично підійти до оцінювання діяльності. Автори статті розглянули аспекти впровадження в освітній процес під час навчання програмування онлайн-систем з автоматизованою перевіркою розв'язків (Махровська Н., Погромська Г., 2020). На важливості застосування онлайн-засобів для вивчення програмування під час дистанційного навчання наголошено в роботі Г. Шевченко (Шевченко Г., 2023), де акцентовано на перевагах середовищ розроблення, що працюють онлайн.

Погоджуємося з думкою вчених А. Рохаса-Лопеса, Ф. Х. Гарсія-Пеньялво, що концепція розвитку алгоритмічного мислення набуває важливого значення (Rojas-López A., García-Peñalvo F., 2020). Це відбувається під час навчання програмування. Напрямок алгоритмізації та кодингу нині у світі впроваджується для учнів на різних циклах навчання для формування навичок обчислювального мислення, що виходять далеко за рамки лише інформатики, адже їх можна інтегрувати з іншими галузями знань, природничими науками і технологіями, зокрема мистецтвом (STEAM).

У дослідженні автори Р. Паукар-Кура-сма, К. Вільялба-Кондорі застосували методику оцінювання обчислювального мислення через залучення алгоритмічних конструкцій послідовностей, циклів, паралелізму, умовних виразів, операторів і маніпулювання даними та довели оптимальність його для використання здобувачів освіти під час розв'язання проблемних питань (Paucar-Curasma R., Villalba-Condori K., 2022).

Проаналізували вплив алгоритмічного мислення на якість знань із математики через освітню робототехніку дослідники І. Соуза, В. Андраде, М. Сампайо (Souza I., Andrade W., Sampaio M., 2019). Результати оцінювання показали, що учні надають перевагу роботам, що мають більше функцій взаємодії, доєднання та програмування, можуть значно сприяти розвитку їхніх навичок алгоритмічного та математичного мислення (Souza I., Andrade W., Sampaio M., 2019).

Аналіз джерел дав змогу дійти висновку про важливість розвитку ключових компетентностей, зокрема алгоритмічного мислення, під час навчання програмування на різних циклах навчання з використанням прогресивних мов програмування, відповідно до віку здобувачів освіти. Невід'ємною частиною цього мають стати доцільно підібрані засоби та методи навчання.

**Метою** статті є розкрити особливості впровадження алгоритмізації та програмування в освітній процес з інформатики.

Відповідно до мети окреслено **завдання**:

1. Проаналізувати сучасні мови програмування та доцільність їх використання у школі.
2. Виокремити чинні освітні технології навчання алгоритмізації та програмування в шкільній інформатиці в Україні та світі.
3. Запропонувати варіант програми підвищення кваліфікації вчителів із поглиблення професійної компетентності в галузі програмування.

**Виклад основного матеріалу.**

Сучасна школа має на меті не лише навчити учнів базових комп'ютерних навичок, зокрема послуговування прикладними програмами, але й розвивати їхнє логічне, структуроване та аналітичне мислення.

Програмування відкриває перед учнями можливість навчитися розв'язувати задачі через побудову структурованих алгоритмів і системного аналізу. Саме це створює підґрунтя для успішного використання навичок XXI століття в майбутньому. Водночас перед учителем постає проблема вибору мови програмування та ступеня заглиблення у її вивчення в школі.

Окреслимо ключові аспекти цього процесу. По-перше, учні здобувають базові знання архітектури програмного забезпечення, розуміють, як працюють сучасні комп'ютерні системи. По-друге, навчаються розв'язувати проблеми комплексно, застосовуючи алгоритмічне мислення. По-третє, процес кодингу розвиває креативність і дає змогу формувати навички, які будуть корисними в багатьох сферах, незалежно від обраної професії.

Є багато досліджень та рекомендацій щодо вибору мови програмування та відповідного IDE для шкільного навчання, що, на думку дослідників, повинні забезпечувати навчально-методичну підтримку, бути поширеними, актуальними, зручними у використанні та відповідати технічним можливостям шкільного обладнання. Серед основних критеріїв вибору програмного середовища виділяють: технічні характеристики комп'ютерів і системні вимоги середовища; наявність необхідних операційних систем і додаткового програмного забезпечення; функціональні можливості та зручність інтерфейсу; рівень професійної підготовки вчителя та доступність методичних матеріалів для роботи з обраною мовою.

Ігнорування цих критеріїв може призводити до труднощів. Наприклад, складний для відповідної вікової категорії здобувачів освіти або іншомовний чи «заплута-

ний» інтерфейс значно ускладнює процес навчання. Отже, вибір доцільної мови програмування є складним і відповідальним завданням, яке потребує творчого підходу. Учитель має критично обирати середовище програмування з урахуванням циклу навчання учнів.

Неперервність змістової лінії «Основи алгоритмізації та програмування» закладена у зміст чинних програм ШКІ для 2–9 класів. Але питання вибору мови програмування досі є відкритим, і рішення приймає вчитель особисто залежно від умов. Цілком погоджуємося з думкою авторів Т. Кобильник, У. Когут, О. Сікора, В. Жидик (Кобильник Т., Когут У., Сікора О., Жидик В., 2022), що для навчання основ алгоритмізації у 2–6 класах доцільно використовувати блочні візуальні середовища програмування (Scratch, Blockly, Inventor тощо); у 7–9 класах рекомендовано обрати мову Python.

Наразі кількість мов програмування становить понад дві з половиною тисячі, і з кожним роком ця кількість збільшується. На основі індексу ТЮВЕ (січень, 2025), який визначає частоту запитів щодо вибору мови програмування, найпопулярнішими сьогодні є Python, C++, Java, C, C#, JavaScript. За рейтингом мов програмування від Асоціації розробників DOU (за версією ІТ-спеціалістів з України) домінувальними є JavaScript, TypeScript, Python, Java, C#.

Потенціал мов і систем програмування, а також легкість вивчення за критеріями мультиплатформності та функціональних можливостей порівняно в роботі О. Джуманкозієва та М. Р. Файзієвої (Jumanqo'ziyev O., Fayziyeva M., 2022).

Беручи до уваги аналіз особливостей мов програмування та функціоналу середовищ програмування, потрібного для організації та реалізації освітнього процесу, автори статті пропонують критерії для ефективного вибору мов програмування (див. таблицю 1). Рекомендуємо до вивчення у базовій школі обрати мову програмування Python.

*Порівняльна характеристика мов програмування*

	Python	C++	C#	Java	Java Script	Pascal	Scratch
Парадигма програмування	Процедурна, ООП	Процедурна, ООП	ООП	ООП	Процедурна	Процедурна, ООП	ООП
Типізація змінних	Динамічна	Статична	Статична	Статична	Динамічна	Статична	Динамічна
«Поріг входження» (простота вивчення)*	Простий	Середній	Складний	Складний	Простий	Простий	Простий
Середовища програмування	IDLE, VSCode, PyCharm, Thonny	Code;Block, CLion, Dev-Cpp, Visual Studi, Qt	Visual Studio, VS-Code	NetBeans, Eclipse	VSCode, Web-Storm	Lazarus, АЛГО	Scratch
Практичне застосування (місце за ТЮВЕ, на січень 2025 р.)	I	II	V	III	VI	-	XII
Рівень навчання	7–11	10–11	10–11	10–11	7–11	7–11	2–6
Методична підтримка в чинних підручниках з інформатики	+	–	–	–		+	+

\*Під час обрання мови як засобу навчання програмування потрібно враховувати легкість її вивчення, або «поріг входження»: простоту синтаксису, доступність для сприйняття керувальних конструкцій, засоби для формування структур даних.

Обов'язково потрібно навчити дітей на уроках інформатики не просто основ комп'ютерної грамотності, засвоювання офісних програм і графічних редакторів, а логічного, алгоритмічного, аналітичного мислення й основ сучасного програмування. На уроках інформатики вчитель має закласти необхідні базові знання методів програмування і системного підходу до розв'язування задач, розвивати обчислювальне мислення і на конкретних прикладах ознайомлювати учнів із принципами

побудови сучасних комп'ютерних систем.

За умов розвитку сучасної освіти згідно з принципами НУШ зазначимо пріоритет для розвитку компетенцій у сфері ІКТ (від дитячого садка до базової освіти), зокрема в розвитку компетенцій та/або навичок кодування. Ураховуючи ступінь особливостей вивчення кодування, вважаємо доречним розвивати в учнів навички логічного мислення та розв'язування проблем за допомогою підходів програмування та алгоритмічного мислення. Останнє відкри-

ває перспективи та можливості програмування роботів, мобільних пристроїв, додатків на базі Arduino, навчання на основі ігор тощо. Важливо обговорювати досвід, який опрацьовують у всьому світі в спеціалізованих сферах, зокрема проєкт Європейського Союзу TASCLE 3 – Кодування (Tasacle 3 – Coding), що присвячений виявленню, обміну та оцінюванню найефективніших практик і досвіду (містять технологічні та методологічні питання), де акцентовано на розвитку алгоритмічного мислення та відповідних навичок на будь-якому рівні початкової та базової освіти.

Спираючись на аналіз досліджень і власний досвід, можемо з упевненістю сказати, що алгоритмічне мислення – це набір таких навичок, як абстракція, декомпозиція, розпізнавання образів, алгоритмізація, налагодження та розв'язання проблем, які можна поглибити до навичок критичного мислення. На думку авторів, таке мислення є новою парадигмою, яка розширює когнітивні навички учнів, з погляду надання видимості проблемам, які досі були типовими для комп'ютерних техніків і професіоналів, але це, безсумнівно, стане великим внеском у вивчення природничих наук, математики та багатьох інших галузей знань. Зазначене дає змогу через абстракцію та декомпозицію проблем, паралелізм і багаторівневе мислення забезпечити необхідне залучення здобувачів освіти до розвитку навичок моделювання, що дозволяють об'єднати всі навички в широкі знання, здатні інтегрувати різноманітні пізнавальні здібності.

Вивчення програмування є основоположним для набуття навичок алгоритмічного мислення, що відкриває нові можливості вміння абстрагування, дезагрегації проблем, структурування, паралельного оброблення завдань, налагодження та розпізнавання шаблонів.

Алгоритмічне мислення розглядаємо як широку структуру, що передбачає відповідні необхідні навички, процеси та підходи для розв'язання проблем, а «програмування» як ключову практику для підтримки та розвитку когнітивних завдань.

Аналіз джерел підтверджує, що комп'ютерне мислення є важливим для освіти та промисловості в XXI столітті, оскільки воно забезпечує складні когнітивні навички, необхідні не лише для науково-технічного розвитку суспільства, але й для викликів, пов'язаних із майбутньою професійною діяльністю. Автори С. Бокконі, А. Чіоккарієлло, Г. Детторі, А. Феррарі, К. Енгельхардт зазначають, що одним із завдань на рівні держави є підготовка учнів до життя в технологічному суспільстві, однак це пов'язано з дефіцитом підготовлених кадрів у сфері інформатики (Bocconi S., Chiocciariello A., Dettori G., Ferrari A., Engelhardt K., 2016).

Розвивати алгоритмічне та логічне мислення можна під час навчання програмування. Цей розділ традиційно вважають складним як для учнів, так і для вчителів через низку вище окреслених причин. Оскільки розвиток навичок XXI століття наразі є пріоритетною задачею освіти, серед яких зокрема виділяємо комп'ютерне мислення, то важливо знайти нові та інноваційні способи ефективного навчання змістової лінії «Алгоритмізація і програмування».

Розглянемо навчальні стратегії до ефективного вивчення мов програмування в ЗЗСО. Репродуктивні методи, зокрема робота за зразком, не сприяють досягненню описаних вище цілей. Окреслимо доцільні, на нашу думку, освітні технології: гейміфікація, проєктне та персоналізоване навчання, 4 навчання у співпраці.

Гейміфікація передбачає застосування ігрових елементів (бали, значки, нагороди тощо) для мотивації учнів до вивчення програмування, елементів ігрового дизайну в неігрових контекстах для мотивації та залучення школярів. У контексті такого навчання доречно створювати ситуації-виклики або головоломки, розв'язання яких потребує від здобувачів освіти написання коду. Перетворюючи навчання на гру, учні з більшою ймовірністю залишатимуться залученими та вмотивованими. Цей підхід не тільки робить навчання цікавішим, але

й заохочує учнів самостійно досліджувати концепції програмування.

У рамках проєктного навчання учні працюють над інтегрованими, практико-орієнтованими завданнями. В оглядах А. М. Добровольської (Добровольська А. М., 2018) та О. М. Тадеуша (Тадеуш О. М., 2017), зазначено, що в сучасному розумінні метод проєктів кваліфікують як педагогічну технологію, основними принципами якої є «активність у визначенні завдання та його опрацюванні; практичний характер проєкту, його актуальність, доцільність; інтерес учнів до роботи; поєднання теорії з практикою, знань і навичок; спроможність втілення проєкту; самостійність; творчість; колективна діяльність» (покликання). У такий спосіб здобувачі освіти не лише вивчають концепції алгоритмізації, але й розвивають навички розв'язання проблем. Це мотивує до творчості та стимулює критичне мислення.

Підхід «навчання у співпраці» зумовлює колективне (учні працюють разом у групах) обговорення проблем програмування. Співпраця та спільне залучення мають значний упорядкувальний вплив на ставлення дітей. Іншими словами, на думку К. Шарма, С. Папавласопулу, М. Джаннакос, захоплива об'єднувальна діяльність із програмування значно пом'якшує ставлення дітей до процесу навчання (Sharma K., Papavlasopoulou S., 2019). Такий підхід не лише сприяє навчанню одне в одного, але й розвиває важливі навички роботи в команді.

Персоналізоване навчання – це підхід, який передбачає пристосування освітнього процесу до індивідуальних потреб кожного учасника. Такого ефекту можна досягти завдяки використанню адаптивних

технологій навчання, які регулюють рівень складності вправ із програмування на основі прогресу учня.

Зазначені підходи роблять програмування більш доступним і цікавим для учнів, допомагають їм розвивати важливі для сучасного фахівця навички.

**Практичне значення.** Вищезазначене дозволило дійти висновку про актуальність розроблення практико-орієнтованого курсу з вивчення мови програмування Python «Старт у світ кодування: Python для початківців» як варіанту підвищення кваліфікації вчителів щодо поглиблення професійної компетентності в галузі програмування.

Метою курсу є вдосконалення предметно-методичної та інформаційно-цифрової компетентностей учителів інформатики, природничої та математичної освітніх галузей.

Завданням курсу є:

- ознайомити педагогів з основами програмування на Python: синтаксис, базові алгоритмічні структури й принципи роботи з мовою програмування;
- поглибити їхній інтерес до програмування;
- розвинути логічне мислення учасників;
- ознайомити освітян із роботою в середовищі розробок (наприклад, IDLE, VS Code);
- сприяти співпраці та обміну ідеями здобувачів освіти;
- сформувати в учителів базу для поступового та усвідомленого навчання програмування.

Орієнтовні теми діяльності учасників представлені в таблиці 2.

Таблиця 2.

**Орієнтовні теми діяльності учасників**

№ з/п	Тема заняття	Обсяг навчального часу (в акад. год)
1.	Вступ. IDE для Python. Введення-виведення даних	2

2.	Змінні, типи даних, основні операції мови Python	2
3.	Базові алгоритмічні структури: структура слідування	2
4.	Цілочислова арифметика	2
5.	Логічні операції	2
6.	Структура розгалуження	2
7.	Структура вибору	2
8.	Структура повторення: цикл for	2
9.	Цикл for: функція range	2
10.	Структура повторення: цикл while	2
11.	Оператори переривання	2
12.	Вкладені цикли	2
13.	Числові типи даних	2
14.	Рядковий тип даних	2
15.	Основи роботи зі списками	2

Розроблений курс (покликання на програму: <https://cutt.ly/4e8pSdoa>) створено в рамках сертифікованого заходу Миколаївського ОППО та розраховано на 30 акад. год (1,5 кредиту ЄKTS).

Робота під час сертифікованого заходу сприятиме вдосконаленню предметно-методичної та інформаційно-цифрової компетентностей учителів інформатики, природничої та математичної освітніх галузей, розвитку математичної та інформаційно-комунікаційної компетентностей здобувачів освіти.

**Висновки та перспективи досліджень.** Сучасні мови програмування відкривають широкі можливості для навчання учнів, сприяючи розвиткові різних видів мислення, творчості та навичок розв'язання проблем. Проаналізовано та узагальнено можливості сучасних мов програмування за критеріями: парадигма програмування, типізація змінних, «поріг входження» (простота вивчення), середовища програмування, практичне застосування, рівень навчання, методична підтримка в чинних підручниках з інформатики. Автори рекомендують обирати Scratch для молодших

класів з огляду на його візуальну природу, Python – для базової школи через простоту синтаксису, для ознайомлення з веброзробкою обирати JavaScript (старші класи).

Розглянуто чинні технології навчання алгоритмізації та програмування в курсі шкільної інформатики в Україні та світі, зокрема виділено гейміфікацію, проєктне та персоналізоване навчання, навчання у співпраці. Висвітлено переваги кожного з підходів у мотивації учнів для підвищення їхнього інтересу до вивчення програмування.

Запропоновано авторську програму сертифікованого заходу підвищення кваліфікації вчителів для поглиблення професійної компетентності в галузі програмування на прикладі курсу з вивчення мови програмування Python «Старт у світ кодування: Python для початківців» (покликання: <https://cutt.ly/4e8pSdoa>).

**Перспективність** досліджень полягає у вивченні можливостей інтеграції програмування з математикою, фізикою, інженерією чи іншими науками для реалізації STEAM-підходу та розробленні програм підвищення кваліфікації педагогів у галузі програмування.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Базурін В. М. Середовища програмування як засіб навчання учнів основ програмування / В. М. Базурін // Інформаційні технології і засоби навчання. – 2017. – Т. 59.



– Вип. 3. – С. 13–27.

2. Власій О. Проблема послідовності вивчення програмування / О. Власій, І. Яремій, М. Винничук // Молодь і ринок. – 2021. – №10 (196). – С. 52–57.

3. Добровольська А. М. Метод проектів: формування ІТ-компетентності майбутніх фахівців / А. М. Добровольська // Фізико-математична освіта. – 2018. – №1(15). – С. 35–47.

4. Кобильник Т. Деякі проблемні аспекти навчання основ алгоритмізації та програмування у школі / Т. Кобильник, У. Когут, О. Сікора, В. Жидик // Молодь і ринок. – 2022. – №3–4 (201–202). – С. 97–101. [Електронний ресурс]. – Режим доступу:

5. Махровська Н. А., Погромська Г. С. Застосування онлайн-змагань з програмування в системі практичної підготовки студентів спеціальності «Комп'ютерні науки» / Н. А. Махровська, Г. С. Погромська // Інформаційні технології і засоби навчання. – 2020. – Том 79, № 5. – С. 260–275. DOI: <https://doi.org/10.33407/itlt.v79i5.3084>.

6. Семеніхіна О. В. Проблеми навчання програмувати учнів старших класів та шляхи їх подолання / О. В. Семеніхіна, Ю. О. Руденко // Інформаційні технології і засоби навчання. – 2018. – Том 66. – № 4. – С. 54–64.

7. Тадеуш О. М. Метод проектів як форма продуктивного навчання студентів / О. М. Тадеуш // Науковий часопис НПУ імені М. П. Драгоманова. Серія 16 : Творча особистість учителя : проблеми теорії і практики. – 2017. – Вип. 29. – С. 142–146. [Електронний ресурс]. – Режим доступу: [http://nbuv.gov.ua/UJRN/Nchnpu\\_016\\_2017\\_29\\_33](http://nbuv.gov.ua/UJRN/Nchnpu_016_2017_29_33)

8. Шевченко Г. Переваги використання онлайн-середовища розробки «Replit» для вивчення мови програмування «Python» у закладах освіти під час дистанційного навчання / Г. Шевченко // Вересень. – 2023. – № 96 (1). – С. 139–151.

9. Юрченко А. О. Навчання програмувати в старшій школі крізь призму чинних навчальних програм / А. О. Юрченко, О. В. Семеніхіна, Ю. В. Хворостіна, О. М. Удовиченко, С. І. Петренко // Фізико-математична освіта. – 2019. – Випуск 2(20). – Ч. 2. – С. 48–55.

10. Belmar H. Review on the teaching of programming and computational thinking in the world // *Frontiers in Computer Science*. – 2022. – Volume 4. – P. 1–19. | DOI: <https://doi.org/10.3389/fcomp.2022.997222>

11. Bocconi S., Chiocciariello A., Dettori G., Ferrari A., Engelhardt K., 2016. Developing Computational Thinking in Compulsory Education – Implications for policy and practice // JRC Research Reports, Joint Research Centre. 2016. Retrieved from: <https://ideas.repec.org/p/ipt/iptwpa/jrc104188.html>.

12. Brown N., Wilson G. Ten quick tips for teaching programming // *PLoS Comput Biol*. – 2018. – 14(4). Retrieved from: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006023#pcbi.1006023.ref002>. DOI: <https://doi.org/10.1371/journal.pcbi.1006023>.

13. Coppelli Ortiz G. The economic globalization of the 21st Century. Between globalization and deglobalization // Instituto de Estudios Internacionales – Universidad de Chile. 2018. – № 191. – P. 57–80. DOI: <https://doi.org/10.5354/0719-3769.2019.52048>

14. Jumanqo'ziyev O. O., Fayziyeva M. R. Development tendencies and classification of programming languages taught in high schools // *Galaxy International Interdisciplinary Research Journal*. – 2022. – 10(12). – 185–189. Retrieved from <https://www.giirj.com/index.php/giirj/article/view/4618>

15. Paucar-Curasma R., Villalba-Condori K., Arias-Chavez D., Le N. T., Garcia-Tejada G., Frango-Silveira I. Evaluation of computational thinking using four educational robots with primary school students in Peru // *Education in the Knowledge Society*. – 2022. – № 23. DOI: <https://doi.org/10.14201/eks.26161>

16. Rojas-López, A., García-Peñalvo, F. J. Evaluación del pensamiento computacional para el aprendizaje de programación de computadoras en educación superior // *Revista de*

Educación a Distancia (RED). – 2020. – Núm. 63, Vol. 20. Artíc. 4. DOI: <https://doi.org/10.6018/red.409991>.

17. Sharma K., Papavlasopoulou S., Giannakos M. Coding games and robots to enhance computational thinking: How collaboration and engagement moderate children's attitudes? // International Journal of Child-Computer Interaction. – 2019. – Volume 21. – September, 65–76. DOI: <https://doi.org/10.1016/j.ijcci.2019.04.004>.

18. Souza I., Andrade W., Sampaio M. Analyzing the effect of computational thinking on mathematics through educational robotics // IEEE Frontiers in Education Conference (FIE) (Covington, KY: IEEE), 16–19 October 2019. – 1–7. DOI: 10.1109/FIE43999.2019.9028384  
16. Tackle 3 Coding / Portal. Retrieved from <http://www.tackle3.eu/en/>

## IMPLEMENTING ALGORITHMIZATION AND PROGRAMMING IN EDUCATION: CHALLENGES AND PERSPECTIVES

**Makhrovska Natalia,**

*PhD, Associate Professor  
of the Department of Theory  
and Methods for Teaching*

*the Natural Sciences, Mathematics  
and Information Technologies*

*Mykolaiv In-Service Teachers Training Institute  
4-a, Admiralska Street, 54001, Mykolaiv, Ukraine  
[natalya.makhrovska@moippo.mk.ua](mailto:natalya.makhrovska@moippo.mk.ua)*

**Pohromska Hanna,**

*PhD, Docent, Associate Professor  
of the Department of Theory  
and Methods for Teaching*

*the Natural Sciences, Mathematics  
and Information Technologies*

*Mykolaiv In-Service Teachers Training Institute  
4-a, Admiralska Street, 54001, Mykolaiv, Ukraine  
[hanna.pohromska@moippo.mk.ua](mailto:hanna.pohromska@moippo.mk.ua)*

*The article explores the prospects of integrating the fundamentals of algorithmization and programming into school education as a tool for developing algorithmic thinking, which is essential for shaping key 21st-century competencies. It highlights the necessity of adapting school curricula to the demands of the digital society, where programming skills are crucial for solving complex problems, fostering critical thinking, and encouraging a creative approach.*

*Special attention is given to the relevance of implementing modern teaching approaches to programming, such as gamification, project-based learning, personalized learning, and collaboration. The advantages of each approach in motivating students and increasing their interest in programming are discussed. The criteria for selecting programming languages that align with students' age-specific characteristics are outlined. In particular, Scratch is recommended for younger students, Python for middle school, and Python or JavaScript for high school students.*

*The importance of improving teachers' qualifications in programming is emphasized, and a practical course, «Introduction to Coding: Python for Beginners», is proposed. This*

course aims to enhance teachers' subject-specific, methodological, and digital competencies. It includes learning the basics of Python, algorithmic structures, and solving practical tasks that develop logical thinking and promote a systematic approach to problem-solving.

The article underscores the importance of fostering algorithmic thinking as a tool that not only enhances students' cognitive abilities but also prepares them for the challenges of digitalization in society.

**Keywords:** algorithmic thinking; algorithmization; IT education; programming languages; Python; teaching methodologies.

## REFERENCES

1. Bazurin, V. M. (2017). Seredovyscha prohramuvannia yak zasib navchannia uchniv osnov prohramuvannia [Programming environments as a means of teaching students the basics of programming]. *Informatsiini tekhnologii i zasoby navchannia*, 59, 3, 13–27 (ukr).
2. Belmar H. (2022). Review on the teaching of programming and computational thinking in the world. *Frontiers in Computer Science*. Vol. 4, 1–19. | DOI: <https://doi.org/10.3389/fcomp.2022.997222> (eng).
3. Bocconi S., Chiocciariello A., Dettori G., Ferrari A. & Engelhardt K. (2016). Developing Computational Thinking in Compulsory Education – Implications for policy and practice. JRC Research Reports, Joint Research Centre. Retrieved from: <https://ideas.repec.org/p/ipt/iptwpa/jrc104188.html> (eng).
4. Brown N. & Wilson G. (2018). Ten quick tips for teaching programming. *PLoS Comput Biol*, 14(4). Retrieved from: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006023#pcbi.1006023.ref002>. DOI: <https://doi.org/10.1371/journal.pcbi.1006023> (eng).
5. Coppelli Ortiz G. (2018). The economic globalization of the 21st Century. Between globalization and deglobalization. Instituto de Estudios Internacionales – Universidad de Chile, 191, 57–80. DOI: <https://doi.org/10.5354/0719-3769.2019.52048> (eng).
6. Dobrovol'ska, A. M. (2018). Metod proektiv: formuvannia IT-kompetentnosti maibutnikh fakhivtsiv [The Project method: formation of IT competence of the future specialists]. *Fizyko-matematychna osvita*, issue 1(15), 35–47 (ukr).
7. Jumanqo'ziyev O. O. & Fayziyeva M. R. (2022). Development tendencies and classification of programming languages taught in high schools. *Galaxy International Interdisciplinary Research Journal*, 10(12), 185–189. Retrieved from: <https://www.giirj.com/index.php/giirj/article/view/4618> (eng).
8. Kobylnyk, T., Kohut, U., Sikora, O. & Zhydyk, V. (2022). Deiaki problemni aspekty navchannia osnov alhorytmizatsii ta prohramuvannia u shkoli [Some problematic aspects of learning the basics of algorithmization and programming at school]. *Molod i rynek*, 3–4 (201–202), 97–101. Retrieved from: <http://mir.dspu.ele/view/259956/256328> (ukr).
9. Makhrovska, N. A. & Pohromska, H. S. (2020). Zastosuvannia onlain zmahan z prohramuvannia v systemi praktychnoi pidhotovky studentiv spetsialnosti «Kompiuterni nauky» [Application of online programming competitions in the system of practical training of students majoring in «Computer Science»]. *Informatsiini tekhnologii i zasoby navchannia*. Tom 79, 5, 260–275. DOI: <https://doi.org/10.33407/itlt.v79i5.3084> (ukr).
10. Paucar-Curasma R., Villalba-Condori K., Arias-Chavez D., Le N. T., Garcia-Tejada G. & Frango-Silveira I. (2022). Evaluation of computational thinking using four educational robots with primary school students in Peru, 23. DOI: <https://doi.org/10.14201/eks.26161> (eng).
11. Rojas-López A. & García-Peñalvo F. J. (2020). Evaluación del pensamiento computacional para el aprendizaje de programación de computadoras en educación superior.

Revista de Educación a Distancia (RED), Núm. 63, Vol. 20, Artíc. 4. DOI: <https://doi.org/10.6018/red.409991> (eng).

12. Semenikhina, O. V. & Rudenko, Yu.O. (2018). Problemy navchannia prohramuvaty uchniv starshykh klasiv ta shliakhy yikh podolannia [Problems of teaching high school students to program and ways to overcome them]. *Informatsiini tekhnologii i zasoby navchannia*. Tom 66, 4, 54–64 (ukr).

13. Sharma K., Papavlasopoulou S. & Giannakos M. (2019). Coding games and robots to enhance computational thinking: How collaboration and engagement moderate children's attitudes? *International Journal of Child-Computer Interaction*, 21, 65–76. DOI: <https://doi.org/10.1016/j.ijcci.2019.04.004> (eng).

14. Shevchenko, H. (2023). Perevahy vykorystannia onlain-seredovyscha rozrobky «Replit» dlia vyvchennia movy prohramuvannia «Python» u zakladakh osvity pid chas dystantsiinoho navchannia [Advantages of using the online development environment «Replit» for learning the programming language «Python» in educational institutions during distance learning]. *Veresen*, 96, 139–151. DOI: <https://doi.org/10.54662/veresen.1.2023.11> (ukr).

15. Souza I., Andrade W. & Sampaio M. (2019). Analyzing the effect of computational thinking on mathematics through educational robotics. *IEEE Frontiers in Education Conference (FIE)*, 16–19 October, 1–7. DOI: 10.1109/FIE43999.2019.902838416. Tackle 3 Coding / Portal. Retrieved from <http://www.tackle3.eu/en/> (eng).

17. Tadeush, O. M. (2017). Metod proektiv yak forma produktyvnoho navchannia studentiv [Project method as an effective form teaching students]. *Naukovyi chasopys NPU imeni M. P. Drahomanova. Serii 16: Tvorchia osobystist uchytelia: problemy teorii i praktyky*, issue 29, 142–146. Retrieved from: [http://nbuv.gov.ua/UJRN/Nchnpu\\_016\\_2017\\_29\\_33](http://nbuv.gov.ua/UJRN/Nchnpu_016_2017_29_33) (ukr).

18. Vlasii, O. (2021). Problema poslidoynosti vyvchennia prohramuvannia [The problem of the sequence of studying programming]. *Molod i rynek*, 10 (196), 52–57 (ukr).

19. Yurchenko, A. O., Semenikhina, O. V., Khvorostina, Yu. V., Udovychenko, O. M. & Petrenko, S. I. (2019). Navchannia prohramuvaty v starshii shkoli kriz pryzmu chynnykh navchalnykh prohram [Learning to program in high school through the prism of current curricula]. *Fizyko-matematychna osvita*. Vypusk 2 (20). Ch. 2, 48–55 (ukr).

Стаття надійшла до редакції: 02.02.2025

Статтю подано до друку: 26.03.2025