



- ” Момот Р., Юрченко А., Семеніхіна О. Візуальне моделювання блок-схем у професійній підготовці вчителів інформатики. *Освіта. Інноватика. Практика*, 2025. Том 13, № 7. С. 84-89. <https://doi.org/10.31110/2616-650X-vol13i7-012>.
- Momot R., Yurchenko A., Semenikhina O. Vizualne modeliuvannya blok-skhem u profesiinii pidhotovtsi vchyteliv informatyky [Visual modeling of flowcharts in the professional training of computer science teachers]. *Osvita. Innovatyka. Praktyka – Education. Innovation. Practice*, 2025. Vol. 13, No 7. S. 84-89. <https://doi.org/10.31110/2616-650X-vol13i7-012>.

УДК 004.421.2:37.091.313:378.147-051(07)

DOI: 10.31110/2616-650X-vol13i7-012

Роман МОМОТ<sup>1</sup>, Артем ЮРЧЕНКО<sup>2</sup>, Олена СЕМЕНІХІНА<sup>3</sup>

<sup>1-3</sup> Сумський державний педагогічний університет імені А. С. Макаренка, Україна

<sup>1</sup> <https://orcid.org/0000-0003-0861-1925>  
roman.bo95@gmail.com

<sup>2</sup> <https://orcid.org/0000-0002-6770-186X>  
a.yurchenko@fizmatsspu.sumy.ua

<sup>3</sup> <https://orcid.org/0000-0002-3896-8151>  
e.semenikhina@fizmatsspu.sumy.ua

## ВІЗУАЛЬНЕ МОДЕЛЮВАННЯ БЛОК-СХЕМ У ПРОФЕСІЙНІЙ ПІДГОТОВЦІ ВЧИТЕЛІВ ІНФОРМАТИКИ

**Анотація.** Формування алгоритмічного мислення як ключового компоненту професійної підготовки майбутніх учителів інформатики є фундаментальним для їх подальшого успішного оволодіння програмуванням та викладанням цієї дисципліни. Незважаючи на те, що алгоритмічний підхід складає основу програмної діяльності та вимагає вміння аналізувати задачі, мислити структурно та візуалізувати логіку рішень, сучасна університетська освіта часто демонструє фрагментарність у його розвитку. Це призводить до того, що навчання зводиться до механічного розв'язування вправ або заучування синтаксису без глибокого розуміння принципів побудови алгоритмів. У статті досліджено педагогічний потенціал побудови блок-схем як інструмента розвитку алгоритмічного мислення у процесі професійної підготовки майбутніх учителів інформатики. Особливу увагу приділено окресленню педагогічних умов, за яких візуальне моделювання алгоритмів сприяє осмисленому формуванню логіко-структурного мислення студентів, а також аналізу цифрових середовищ, придатних для реалізації такого підходу в навчальному процесі. Аргументовано, що блок-схема виступає проміжною ланкою між вербальним описом задачі та програмною реалізацією, дозволяючи студентам не лише формалізувати алгоритм, а й розвивати здатність до його пояснення в педагогічному контексті. У результаті дослідження виокремлено три групи умов ефективного застосування блок-схем: когнітивно-методичні, організаційно-технологічні та професійно-орієнтовані. Проведено порівняльний аналіз функціоналу цифрових інструментів (Flowgorithm, Visual Paradigm Online, diagrams.net, Blockly, EduBlocks), за результатами якого сформульовано рекомендації щодо їх доцільного використання на різних етапах навчання алгоритмів. Підкреслено необхідність поєднання технічного освоєння середовищ з розвитком педагогічної рефлексії у студентів, що дозволяє сформувати методичну готовність до викладання основ алгоритмізації в умовах шкільної інформатики.

**Ключові слова:** алгоритмічне мислення; програмування; блок-схема; візуальне моделювання; професійна підготовка; майбутні вчителі інформатики; цифрові інструменти; методика навчання програмування; візуалізація; інформатика.

Roman MOMOT<sup>1</sup>, Artem YURCHENKO<sup>2</sup>, Olena SEMENIKHINA<sup>3</sup>

<sup>1-3</sup> Sumy State Pedagogical University named after A.S. Makarenko, Ukraine

<sup>1</sup> <https://orcid.org/0000-0003-0861-1925>  
roman.bo95@gmail.com

<sup>2</sup> <https://orcid.org/0000-0002-6770-186X>  
a.yurchenko@fizmatsspu.sumy.ua

<sup>3</sup> <https://orcid.org/0000-0002-3896-8151>  
e.semenikhina@fizmatsspu.sumy.ua

## VISUAL MODELING OF FLOWCHARTS IN THE PROFESSIONAL TRAINING OF COMPUTER SCIENCE TEACHERS

**Abstract.** The development of algorithmic thinking as a key component of professional training for future computer science teachers is fundamental to their subsequent successful mastery of programming and effective teaching of this discipline. Although the algorithmic approach forms the basis of software development and requires skills in problem analysis, structured thinking, and solution logic visualization, modern university education often demonstrates a fragmented approach to its development. This results in learning being reduced to mechanical exercise solving or syntax memorization without a deep understanding of algorithm design principles. The article explores the pedagogical potential of constructing flowcharts as a tool for developing algorithmic thinking in the process of professional training of future computer science teachers. Special attention is given to defining the pedagogical conditions under which visual modeling of algorithms contributes to the meaningful formation of students' logical-structural thinking, as well as to analyzing digital environments suitable for implementing such an approach in the educational process. It is argued that the flowchart serves as an intermediate link between the verbal description of a problem and its programming implementation, allowing students not only to formalize the algorithm but also to develop the ability to explain it in a pedagogical context. As a result of the study, three groups of conditions for the effective application of flowcharts are identified: cognitive-methodological, organizational-technological, and professionally-oriented. A comparative analysis of the functionality of digital tools (Flowgorithm, Visual Paradigm Online, diagrams.net, Blockly, EduBlocks) is conducted, based on which recommendations for their appropriate use at different stages of learning algorithms are formulated. The necessity of combining technical mastery of environments with the development of students' pedagogical reflection is emphasized, which makes it possible to form methodological readiness for teaching the fundamentals of algorithmization in school informatics.

**Keywords:** algorithmic thinking; programming; flowchart; visual modeling; professional training; future computer science teachers; digital tools; methodology of teaching programming; visualization; informatics.

**Постановка проблеми.** У сучасній системі підготовки майбутніх учителів інформатики ключове місце займає формування алгоритмічного мислення, що є основою для подальшого оволодіння програмуванням і викладанням цього змістового компонента в школі. Оскільки алгоритми становлять фундамент будь-якої програмної діяльності, уміння аналізувати задачі, мислити алгоритмічно та візуалізувати логіку їх розв'язання є необхідними складовими професійної компетентності вчителя. Проте в реальній практиці університетської освіти спостерігається фрагментарність підходів до розвитку алгоритмічного мислення, що нерідко зводиться до механічного розв'язання задач або запам'ятовування структур програмного коду без достатнього осмислення логіки його побудови.

Одним із ефективних засобів подолання цієї проблеми є використання візуального моделювання, зокрема побудови блок-схем як способу представлення алгоритмів у графічній формі. Блок-схема виступає інструментом структурованого мислення, який дозволяє подати складну послідовність дій у вигляді логічно пов'язаних блоків і переходів між ними. Такий підхід не лише полегшує засвоєння сутності алгоритмічних конструкцій, а й сприяє формуванню здатності до декомпозиції задач, аналізу умов, циклів і логічних розгалужень, що є необхідними для ефективного навчання програмуванню в умовах шкільної освіти. Водночас використання блок-схем у навчальному процесі потребує відповідного програмного забезпечення, яке забезпечує візуальне моделювання алгоритмів. На сьогодні існує низка цифрових середовищ, що дають змогу будувати блок-схеми з різним ступенем інтерактивності та складності. Проте у більшості випадків студенти педагогічних спеціальностей не мають достатньої підготовки щодо вибору та ефективного використання таких інструментів, що знижує якість методичної складової їхньої професійної підготовки. Таким чином, постає потреба в комплексному дослідженні потенціалу блок-схем як інструмента візуального моделювання в підготовці майбутніх учителів інформатики.

**Аналіз актуальних досліджень.** Питання формування алгоритмічного мислення у майбутніх учителів інформатики набуває особливої значущості в умовах поширення цифрових технологій та зростання ролі програмування в загальній середній освіті. Дослідники наголошують, що ефективне навчання програмуванню починається не зі синтаксису мови, а з формування здатності логічно аналізувати проблему та представляти її у вигляді структурованого алгоритму. У цьому контексті блок-схеми виступають дієвим засобом візуального моделювання алгоритмів, що забезпечує когнітивну доступність матеріалу та стимулює розвиток мислення учнів і студентів [1; 8].

Блок-схеми дозволяють відобразити логіку виконання алгоритму через систему умов, циклів, послідовних або паралельних дій. Їх використання як проміжного рівня між описом задачі та програмною реалізацією дає змогу майбутнім учителям краще усвідомити структуру майбутньої програми, сприяє формуванню навичок декомпозиції та вибору адекватної структури управління. У роботах [5; 7] наголошено, що засвоєння основ алгоритмізації потребує особливої методичної підтримки, яка забезпечуватиме наочність і прикладну спрямованість навчання. Автори підкреслюють важливість формалізації задач у вигляді структурованих схем та потребу у відповідному дидактичному інструментарії для педагогічної підготовки.

Науковці акцентують увагу на зв'язку між математичною та алгоритмічною підготовкою, вказуючи, що саме графічні представлення алгоритмів сприяють побудові міждисциплінарних зв'язків і формуванню операційного стилю мислення [3]; учні, які працювали з візуальними представленнями алгоритмів, демонстрували кращі результати при переході до текстового кодування [4]. Дослідники розглядають візуальну складову як інструмент формування мотивації учнів і студентів до вивчення програмування [6]. Зокрема, робота з графічними елементами, включно з елементами побудови блок-схем або графічними об'єктами, сприяє кращому розумінню зв'язку між діями та результатом виконання коду. Важливо, що у дослідженні зазначено: студенти педагогічних спеціальностей, які мали досвід роботи з такими інструментами, продемонстрували вищу методичну готовність до викладання програмування.

Водночас аналіз чинних практик демонструє недостатню увагу до цілеспрямованого використання спеціалізованих середовищ побудови блок-схем у курсах фахової підготовки. Незважаючи на доступність таких інструментів, як Flowgorithm, Visual Paradigm, diagrams.net, Lucidchart чи Python Tutor, вони залишаються слабо інтегрованими в навчальний процес. Це вказує на потребу розроблення методичних рішень, що забезпечать їх ефективне застосування у зв'язку з опрацюванням типових обчислювальних задач.

**Мета і методи дослідження.** Метою цього дослідження є з'ясування умов до навчання побудови блок-схем як інструмента розвитку алгоритмічного мислення майбутніх учителів інформатики, а також аналіз доступних цифрових середовищ для реалізації візуального моделювання алгоритмів у фаховій підготовці вчителів інформатики.

Методологічну основу дослідження становлять принципи когнітивної візуалізації, концепція візуально-орієнтованого навчання та положення педагогіки алгоритмізації. Теоретичне підґрунтя

було забезпечено шляхом аналізу наукових праць, присвячених проблемам формування алгоритмічного мислення, методиці викладання програмування та візуалізації алгоритмів у шкільній і вищій освіті. У ході дослідження застосовано такі методи: теоретичний аналіз фахових джерел для виявлення стану досліджуваної проблеми та обґрунтування ефективності візуального підходу у навчанні алгоритмізації; порівняльний аналіз функціоналу програмних середовищ (Flowgorithm, draw.io, Visual Paradigm Online, diagrams.net, Lucidchart, EduBlocks, Scratch, Python Tutor) щодо можливостей побудови, збереження, інтеграції та коментування блок-схем.

**Результати дослідження.** Дослідження дало змогу окреслити сукупність педагогічних умов, за яких навчання побудови блок-схем стає ефективним інструментом формування алгоритмічного мислення у майбутніх учителів інформатики. Узагальнення результатів аналізу наукової літератури, спостережень за освітнім процесом, а також якісних відгуків студентів дало підстави виокремити три групи таких умов: когнітивно-методичні, організаційно-технологічні та професійно-орієнтовані.

До когнітивно-методичних умов належить забезпечення поступового переходу від опису задачі природною мовою до її формалізації через побудову блок-схеми, яка виступає інструментом структурованого мислення. На цьому етапі особливо важливою є наявність чіткої інструкції щодо правил побудови блок-схем, типів блоків, символів та зв'язків, що дозволяє уникнути формалізму і сприяє усвідомленню логіки алгоритму. Блок-схема у цьому випадку виконує роль медіатора між мовою задачі та мовою програмування, що дозволяє студенту зосередитися на логічній структурі розв'язання до написання коду.

Організаційно-технологічні умови включають наявність доступного, інтуїтивно зрозумілого програмного середовища для побудови блок-схем, яке дозволяє студентам зосередитися на логіці алгоритму, а не на складнощах інтерфейсу чи синтаксису. Практичні заняття мають передбачати самостійну та групову роботу з побудови блок-схем, супроводжувану рефлексією, взаємоаналізом і представленням рішень, що створює умови для розвитку критичного мислення, навичок аналізу та комунікації. Зокрема, ефективним є застосування проєктного підходу, коли студенти моделюють блок-схеми до задач, які вони згодом реалізують у коді.

До професійно-орієнтованих умов належить включення елементів методичної підготовки: аналіз типових учнівських помилок, обговорення педагогічної доцільності використання блок-схем на різних етапах навчання програмуванню, моделювання фрагментів уроків. Таке спрямування активізує у студентів педагогічну рефлексію, сприяє формуванню готовності адаптувати візуальні методи до вікових особливостей і пізнавальних потреб учнів. Залучення студентів до побудови власних дидактичних матеріалів із використанням блок-схем підвищує усвідомленість у виборі засобів навчання та розвиває педагогічне передбачення. Таким чином, ефективність використання блок-схем у розвитку алгоритмічного мислення забезпечується за умови дотримання тріади: осмисленість дій (когнітивно-методичний рівень), технологічна підтримка діяльності (організаційний рівень) та професійне моделювання майбутньої педагогічної практики (методико-рефлексивний рівень). У поєднанні ці умови створюють навчальне середовище, що сприяє не лише засвоєнню алгоритмічних структур, а й розвитку здатності працювати з ними на рівні розуміння, пояснення й викладання.

У межах дослідження здійснено аналіз і порівняння сучасних цифрових середовищ, які можуть бути ефективно використані в процесі навчання побудови блок-схем майбутніми вчителями інформатики. Метою аналізу було виявити функціональні можливості, дидактичний потенціал і доступність середовищ для візуального моделювання алгоритмів з урахуванням специфіки педагогічної підготовки.

До розгляду було включено такі програмні рішення, як Flowgorithm, Visual Paradigm Online, draw.io (diagrams.net), Lucidchart, Blockly, EduBlocks, а також Python Tutor як приклад середовища для візуалізації виконання коду. Результати порівняння свідчать про те, що Flowgorithm є одним із найбільш придатних для формування базових алгоритмічних уявлень завдяки поєднанню візуальної побудови блок-схеми з можливістю автоматичного генерування коду кількома мовами (C++, Java, Python). Це середовище забезпечує поетапну візуалізацію виконання алгоритму, підтримує введення/виведення, умовні переходи, цикли та обробку змінних, що робить його універсальним інструментом для навчання на базовому та проміжному рівнях.

Visual Paradigm Online і Lucidchart, хоча й призначені передусім для професійного моделювання процесів, мають переваги в аспекті гнучкості налаштувань, організації складних блок-схем та використання шаблонів. Ці середовища доцільно залучати на етапах формування складніших алгоритмів, зокрема тих, що включають вкладені структури, декомпозицію на підпрограми, або потребують командної роботи над моделюванням. Draw.io, як безкоштовна альтернатива, забезпечує необхідний базовий функціонал і добре інтегрується в хмарні сервіси Google та Microsoft, що зручно для колективного використання.

Blockly і EduBlocks є візуальними середовищами, орієнтованими переважно на початковий рівень навчання, але їх потенціал у підготовці вчителя полягає в тому, що вони дозволяють студентам ознайомитися з принципами блокового програмування, яке широко використовується у школах

(зокрема в робототехніці, STEM-проектах). Python Tutor виконує іншу функцію: воно не дозволяє будувати блок-схеми, але візуалізує виконання написаного коду, що корисно як інструмент для аналізу алгоритму вже після його реалізації у вигляді тексту.

Здійснений аналіз дозволив розташувати зазначені середовища за ступенем дидактичної відповідності до етапів навчання алгоритмів. Початковий етап – Blockly, EduBlocks; базовий і проміжний – Flowgorithm, draw.io; просунутий і методичний – Visual Paradigm, Lucidchart. Така класифікація сприяє цілеспрямованому добору інструментів для навчальних цілей залежно від рівня складності задач і дидактичної мети.

З метою систематизації отриманої інформації та забезпечення можливості порівняльного аналізу цифрових інструментів, що використовуються для побудови блок-схем, було укладено узагальнену таблицю, яка відображає функціональні характеристики середовищ, їхній дидактичний потенціал, доступність для студентів та можливості колаборації. Таблиця 1 дозволяє окреслити рівень складності, на який орієнтоване кожне середовище, а також визначити сферу його доцільного педагогічного застосування.

Таблиця 1

Порівняльна характеристика цифрових середовищ для побудови блок-схем

Середовище	Рівень складності	Підтримка виконання алгоритму	Автоматична генерація коду	Підтримка циклів/умов	Інтуїтивність інтерфейсу	Можливість колаборації	Тип використання
Flowgorithm	Базовий-середній	Так	Так	Так	Висока	Обмежена	Навчальний, університетський
Visual Paradigm Online	Середній-просунутий	Ні	Ні	Так	Середня	Так	Професійне моделювання
draw.io (diagrams.net)	Базовий	Ні	Ні	Так	Висока	Так	Освітній, базовий
Lucidchart	Середній-просунутий	Ні	Ні	Так	Середня	Так	Професійне моделювання
Blockly	Початковий	Так (через симуляцію)	Так (частково)	Так	Висока	Обмежена	Шкільний STEM
EduBlocks	Початковий	Так (через код)	Так (Python)	Так	Висока	Обмежена	Шкільний STEM
Python Tutor	Післяпроектний аналіз	Так	Ні	Так	Середня	Ні	Аналіз коду

Наведена таблиця дозволяє зробити висновок, що для різних етапів формування алгоритмічного мислення доцільно залучати різні інструменти. Flowgorithm виявляється найефективнішим для формування базових навичок, тоді як Visual Paradigm або Lucidchart можуть бути використані для поглибленої методичної роботи або в рамках проектної діяльності. Blockly й EduBlocks особливо корисні у контексті майбутнього викладання в шкільному середовищі, що вимагає візуальної простоти та швидкої інтерактивної відповіді. Включення Python Tutor як інструмента для аналізу вже написаного коду допомагає студентам розуміти динаміку виконання алгоритму, закріплюючи зв'язок між схемою і програмним втіленням.

Таким чином, ефективне використання цифрових середовищ для побудови блок-схем у професійній підготовці майбутніх учителів інформатики можливе лише за умови обґрунтованого вибору інструменту відповідно до освітнього етапу, навчальної мети та рівня підготовки студентів. Оптимальна комбінація середовищ із різним рівнем інтерактивності забезпечує поетапне формування вмій з аналізу, побудови й тлумачення алгоритмів.

**Обговорення.** Результати дослідження підтверджують, що побудова блок-схем є ефективним інструментом формування алгоритмічного мислення лише за дотримання певних педагогічних умов, які забезпечують осмислене, поступове й професійно орієнтоване навчання. Поєднання когнітивно-методичної логіки побудови блок-схем з адаптованими цифровими інструментами створює сприятливе середовище для того, щоб студенти переходили від вербального формулювання задачі до її формалізованого й структурованого подання. У межах першого завдання було показано, що сам процес побудови блок-схеми стимулює студентів до осмислення логіки алгоритму, формулювання вхідних параметрів, виокремлення повторюваних дій, а також рефлексії щодо цілей і очікуваного результату кожного кроку.

Це цілком узгоджується з висновками досліджень [1; 2; 5; 9-10], які підкреслюють, що візуальні інструменти сприяють розвитку здатності до логічного передбачення дій та усвідомленого контролю над структурою алгоритму. Методичне залучення студентів до рефлексії щодо побудованих блок-схем активізує не лише когнітивні, а й професійно-методичні ресурси: майбутні вчителі починають розглядати блок-схему не лише як засіб розв'язання, а й як дидактичний інструмент для пояснення, узагальнення і перевірки знань учнів.

У межах другого завдання дослідження було виявлено, що вибір відповідного цифрового середовища значною мірою визначає успішність засвоєння принципів побудови блок-схем. Результати аналізу показали, що Flowgorithm є найбільш придатним для педагогічної підготовки на етапі формування базових навичок алгоритмізації, тоді як draw.io або Visual Paradigm Online доцільніше використовувати для моделювання складніших структур і створення дидактичних матеріалів. Інші середовища, як-от Blockly чи EduBlocks, можуть ефективно виконувати роль «мосту» до шкільного навчання, допомагаючи майбутнім учителям зрозуміти, як адаптувати логіку блок-схем до рівня сприйняття учнів.

Водночас дослідження підтвердило, що недостатньо просто включити блок-схеми до навчальної програми необхідно методично супроводжувати їх використання, поєднуючи технічні навички із педагогічною рефлексією. Це особливо важливо в контексті підготовки майбутнього вчителя, який має не лише розуміти алгоритм, а й уміти пояснити його доступно, наочно і педагогічно доцільно. Тому друге завдання дослідження підкреслило значущість добору цифрового інструменту не лише за функціоналом, а й за потенціалом його використання в методичній практиці.

**Висновки.** Побудова блок-схем, за умови цілеспрямованої педагогічної організації, відіграє суттєву роль у формуванні алгоритмічного мислення майбутніх учителів інформатики. Візуальне моделювання алгоритмів сприяє не лише структурованому розумінню логіки програм, а й розвитку здатності аналізувати, декомпонувати та адаптувати навчальний матеріал до рівня підготовки майбутніх учнів. Розгляд блок-схеми як проміжної форми між вербальним описом задачі та програмним кодом дозволяє студентам осмислювати алгоритм не лише як технічний об'єкт, а як дидактичний засіб.

Окреслено педагогічні умови, що забезпечують ефективність побудови блок-схем у навчанні. Зокрема, це: поетапність подання навчального матеріалу; наявність доступних середовищ для моделювання з інтуїтивним інтерфейсом; інтеграція блок-схем у методичну підготовку; використання схем для аналізу, рефлексії та презентації навчального контенту. Встановлено, що візуалізація алгоритмів активізує пізнавальну діяльність студентів, знижує абстрактність матеріалу і сприяє засвоєнню ключових понять, зокрема розгалужень, циклів, умов зупинки та параметрів функцій.

Результати порівняльного аналізу цифрових середовищ свідчать про необхідність диференційованого підходу до їх добору відповідно до рівня підготовки студентів і навчальних цілей. Flowgorithm доцільно використовувати для базової підготовки та поетапного моделювання обчислювальних процедур. Draw.io та Visual Paradigm Online можуть застосовуватись для створення складніших схем, а також як інструменти для розробки навчальних посібників, відеоінструкцій і презентацій. Blockly та EduBlocks мають перспективу як засоби ознайомлення з блоковим програмуванням і методикою його викладання в школі.

Узагальнюючи отримані результати, можна запропонувати такі практичні рекомендації:

- під час навчання алгоритмізації включати до кожної теми етап побудови блок-схеми із наступним її обговоренням і рефлексією;
- забезпечити поетапний перехід від простих схем (лінійні алгоритми) до складніших (розгалуження, вкладені цикли, ітераційні процедури);
- використовувати блок-схеми як інструмент педагогічного моделювання уроків, зокрема при поясненні нових тем або аналізі типових учнівських помилок;
- включати до структури освітньої програми модулі з ознайомлення студентів із функціоналом різних цифрових середовищ для візуального моделювання;
- поєднувати вивчення алгоритмів із проєктною діяльністю, де студенти створюють дидактичні матеріали на основі побудованих блок-схем.

Таким чином, побудова блок-схем у процесі професійної підготовки майбутніх учителів інформатики має розглядатися не як допоміжний, а як інтегрований компонент навчання, що забезпечує розвиток алгоритмічного мислення, методичної рефлексії та цифрової компетентності.

**Конфлікт інтересів.** Автори підтверджують відсутність фінансових, особистих чи інших інтересів, що можуть розглядатися як потенційний конфлікт інтересів щодо публікації цієї статті.

**Фінансування.** Робота виконана за відсутності фінансової підтримки з боку будь-яких організацій.

**Доступність даних.** Це теоретичне дослідження не передбачає використання додаткових наборів даних.

**Використання штучного інтелекту.** Інструменти штучного інтелекту не використовувались при написанні цієї роботи.

#### Список використаних джерел

1. Coşkunserçe, O. (2023). Comparing the use of block-based and robot programming in introductory programming education: Effects on perceptions of programming self-efficacy. *Computer Applications in Engineering Education*, 31(5), 1234–1255. <https://doi.org/10.1002/cae.22637>
2. Espinal, A., Vieira, C., & Guerrero-Bequis, V. (2022). Student ability and difficulties with transfer from a block-based programming language into other programming languages: A case study in Colombia. *Computer Science Education*, 33(4), 567–599. <https://doi.org/10.1080/08993408.2022.2079867>
3. Khvorostina, Y., Shamonina, V., & Semenikhina, O. (2025). The connection between the study of mathematics and programming through the prism of scientific and pedagogical research. *Вісник науки та освіти*, 4(34), 932–945. [https://doi.org/10.52058/2786-6165-2025-4\(34\)-932-945](https://doi.org/10.52058/2786-6165-2025-4(34)-932-945)
4. Rudenko, Y., Drushlyak, M., Osmuk, N., Shvets, O., Kolyshkin, O., & Semenikhina, O. (2022). Problems of teaching pupils of non-specialized classes to program and ways to overcome them: Local study. *International Journal of Computer Science and Network Security*, 22(1), 105–112. <https://doi.org/10.22937/IJCSNS.2022.22.1.16>
5. Semenikhina, O. V., & Rudenko, Y. O. (2018). Проблеми навчання програмувати учнів старших класів та шляхи їх подолання. *Інформаційні технології і засоби навчання*, 66(4), 54–64.
6. Дегтярьова, Н., Петренко, С., & Удовиченко, О. (2023). Робота з графічними віджетами при вивченні мови програмування Python в закладах загальної середньої освіти. *Освіта. Інноватика. Практика*, 11(4), 26–34. <https://doi.org/10.31110/2616-650X-vol11i4-004>
7. Кобильник, Т., Когут, У., & Жидик, В. (2021). Методичні аспекти вивчення основ алгоритмізації і програмування мовою Python у шкільному курсі інформатики у старших класах. *Фізико-математична освіта*, 31(5), 36–44. <https://doi.org/10.31110/2413-1571-2021-031-5-006>
8. Пенко, В., & Пенко, О. (2023). Використання візуалізації на різних етапах вивчення дисципліни «Програмування». *Освіта. Інноватика. Практика*, 11(2), 31–39. <https://doi.org/10.31110/2616-650X-vol11i2-005>
9. Юрченко, А.О., Беспалий, В.Р., & Семеніхіна, О.В. (2025). Навчання програмуванню в цифрову епоху: роль МООС і візуалізації у формуванні алгоритмічної грамотності. *Сусільство та національні інтереси*, 8(16), 325–335. [https://doi.org/10.52058/3041-1572-2025-8\(16\)-325-335](https://doi.org/10.52058/3041-1572-2025-8(16)-325-335)
10. Юрченко, А. О., Семеніхіна, О. В., Хворостіна, Ю. В., & Удовиченко, О. М. (2019). Навчання програмувати в старшій школі крізь призму чинних навчальних програм. *Фізико-математична освіта*, 2(20), Ч.2, 47–54.

#### References

1. Coşkunserçe, O. (2023). Comparing the use of block-based and robot programming in introductory programming education: Effects on perceptions of programming self-efficacy. *Computer Applications in Engineering Education*, 31(5), 1234–1255. <https://doi.org/10.1002/cae.22637>
2. Espinal, A., Vieira, C., & Guerrero-Bequis, V. (2022). Student ability and difficulties with transfer from a block-based programming language into other programming languages: A case study in Colombia. *Computer Science Education*, 33(4), 567–599. <https://doi.org/10.1080/08993408.2022.2079867>
3. Khvorostina, Y., Shamonina, V., & Semenikhina, O. (2025). The connection between the study of mathematics and programming through the prism of scientific and pedagogical research. *Visnyk nauky ta osvity*, 4(34), 932–945. [https://doi.org/10.52058/2786-6165-2025-4\(34\)-932-945](https://doi.org/10.52058/2786-6165-2025-4(34)-932-945)
4. Rudenko, Y., Drushlyak, M., Osmuk, N., Shvets, O., Kolyshkin, O., & Semenikhina, O. (2022). Problems of teaching pupils of non-specialized classes to program and ways to overcome them: Local study. *International Journal of Computer Science and Network Security*, 22(1), 105–112. <https://doi.org/10.22937/IJCSNS.2022.22.1.16>
5. Semenikhina, O. V., & Rudenko, Y. O. (2018). Problemy navchannia prohramuvaty uchniv starshykh klasiv ta shliakhy yikh podolannia. *Informatsiini tekhnolohii i zasoby navchannia*, 66(4), 54–64.
6. Dehtiarova, N., Petrenko, S., & Udovychenko, O. (2023). Robota z hrafichnymy vidzhetamy pry vyvchenni movy prohramuvannia Python v zakladakh zahalnoi serednoi osvity. *Osvita. Innovatyka. Praktyka*, 11(4), 26–34. <https://doi.org/10.31110/2616-650X-vol11i4-004>
7. Kobyllyk, T., Kohut, U., & Zhydyk, V. (2021). Metodychni aspekty vyvchennia osnov alhorytmizatsii i prohramuvannia movoiu Python u shkylnomu kursu informatyky u starshykh klasakh. *Fizyko-matematychna osvita*, 31(5), 36–44. <https://doi.org/10.31110/2413-1571-2021-031-5-006>
8. Penko, V., & Penko, O. (2023). Vykorystannia vizualizatsii na riznykh etapakh vyvchennia dystsypliny «Prohramuvannia». *Osvita. Innovatyka. Praktyka*, 11(2), 31–39. <https://doi.org/10.31110/2616-650X-vol11i2-005>
9. Yurchenko, A.O., Bepalyi, V.R., & Semenikhina, O.V. (2025). Navchannia prohramuvanniu v tsyfrovu epokhu: rol MOOC i vizualizatsii u formuvanni alhorytmichnoi hramotnosti. *Suspilstvo ta natsionalni interesy*, 8(16), 325–335. [https://doi.org/10.52058/3041-1572-2025-8\(16\)-325-335](https://doi.org/10.52058/3041-1572-2025-8(16)-325-335)
10. Yurchenko, A. O., Semenikhina, O. V., Khvorostina, Yu. V., & Udovychenko, O. M. (2019). Navchannia prohramuvaty v starshii shkoli kriz pryizmu chynnykh navchalnykh program. *Fizyko-matematychna osvita*, 2(20), Ch.2, 47–54.

| Матеріал надійшов до редакції: 16.06.2025 р. | Прийнято до друку: 19.07.2025 р. | Опубліковано: 30.09.2025 р. |



This work is licensed under a Creative Commons License Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0).