

 <https://doi.org/10.31651/2524-2660-2025-1-172-178>

 <https://orcid.org/0000-0002-7005-2602>

ВЕЛИКДАН Юлія

старша викладачка катедри теорії і методики технологічної освіти та комп'ютерної графіки,

Університет Григорія Сковороди в Переяславі

e-mail: jucilenda@gmail.com

УДК 37.091.3:004(045)

МОЖЛИВОСТІ SCRATCH У ФОРМУВАННІ АЛГОРИТМІЧНОГО І ЛОГІЧНОГО МИСЛЕННЯ УЧНІВ БАЗОВОЇ ШКОЛИ

У статті досліджено освітній потенціал платформи Scratch для формування алгоритмічного та логічного мислення в учнів, які опановують основи програмування. Scratch, розроблений Массачусетським технологічним інститутом, є візуальною платформою, що дозволяє створювати інтерактивні ігри, анімації та інші творчі проекти. Інструментарій платформи сприяє розвитку когнітивних здібностей учнів через залучення до проектної діяльності, яка потребує системного планування, аналізу та застосування ключових принципів програмування.

Особлива увага приділена алгоритмічному мисленню, що формується завдяки використанню циклів, умовних операторів, змінних і функцій. Учні навчаються розбивати складні задачі на дрібні підзадачі, декомпонуючи їх для ефективного виконання. Процес налагодження проектів дозволяє покращити критичне мислення через аналіз допущених помилок і пошук шляхів їх усунення. Scratch також допомагає формувати стратегічне мислення завдяки інтеграції творчого підходу до вирішення завдань.

У статті описано приклад створення гри «Лабіринт», яка демонструє ефективність Scratch у навчанні. Учні проходять етапи планування, реалізації та тестування гри, використовуючи алгоритми, структурування дій та контроль їх виконання. Цей процес сприяє розвитку просторового мислення, навичок вирішення проблем і творчості.

Дослідження підтверджує, що Scratch є ефективним інструментом не лише для навчання основам програмування, але й для комплексного розвитку логічного мислення, когнітивних та соціальних компетентностей. Онлайн-спільнота Scratch надає учням можливість співпраці, обміну ідеями та отримання зворотного зв'язку, що сприяє їхньому подальшому професійному та особистісному зростанню. Використання платформи у сучасному навчальному процесі створює умови для інтерактивного навчання та всебічного розвитку учнів.

Ключові слова: програмування; алгоритм; інформатика; технології; критичне мислення; середовище програмування Scratch; ігрові Scratch-проекти; Нова українська школа.

Постановка та обґрунтування актуальності проблеми. Сучасна освіта все більше орієнтується на розвиток когнітивних здібностей учнів, які є основою для їхньої подальшої професійної самореалізації. У світі, що швидко змінюється, здатність до алгоритмічного та логічного мислення стає

необхідною складовою для вирішення складних завдань, прийняття обґрунтованих рішень і адаптації до нових технологічних викликів.

Одна з актуальних проблем сучасної педагогіки – інтеграція інноваційних підходів до навчання, які не лише сприяють засвоєнню базових знань, але й формують навички аналізу, планування та творчого мислення. У цьому контексті програмна платформа Scratch пропонує ефективні інструменти для поєднання технічного навчання з розвитком когнітивних здібностей. Вона дозволяє учням в інтерактивній формі опановувати основи програмування, одночасно формуючи ключові компетентності, що відповідають сучасним освітнім стандартам.

Особливе значення має впровадження Scratch у шкільну програму для розвитку алгоритмічного мислення через проектну діяльність, яка включає декомпозицію задач, використання умовних операторів і циклів, а також налагодження створених програм. Такий підхід не лише підвищує зацікавленість учнів, але й сприяє їхній підготовці до роботи у високотехнологічному середовищі.

Зважаючи на необхідність формування нового покоління, здатного до інноваційної діяльності, проблема використання платформ типу Scratch у навчанні є не лише актуальною, але й стратегічно важливою для розвитку системи освіти в цілому.

Аналіз останніх досліджень та публікацій. Упродовж останніх років світ побачила низка наукових публікацій, у яких підіймалися питання, які присвячені дослідженню середовища Scratch як засіб навчання учнів основ програмування. З-поміж дослідників, які зробили значний внесок у вивчення Scratch, можна виділити Т.Є. Сорочіну, Н.В. Морзе, І.Б. Василюка, Т.М. Барболіну, С.О. Шляпчак, В.О. Корецьку, Л.В. Костерну та інших. Їхні праці сприяють глибшому розумінню ролі Scratch у сучасній освіті та його впливу на розвиток алгоритмічного та логічного мислення учнів.

Мета статті – розкрити освітні можливості Scratch у формуванні навичок програмування, логічного, алгоритмічного мислення

та проектного підходу до вирішення завдань у навчальному процесі.

Виклад основного матеріалу дослідження. У сучасному світі, де технології проникають у всі сфери життя, розвиток алгоритмічного та логічного мислення набуває особливої значущості. Ці навички необхідні не лише для програмістів, а й для всіх, хто прагне ефективно аналізувати інформацію, ухвалювати обґрунтовані рішення та знаходити оптимальні способи розв'язання завдань. Алгоритмічне мислення дозволяє структурувати процес пошуку рішень, формувати логічну послідовність дій та оцінювати їхню ефективність. Логічне мислення, своєю чергою, сприяє глибшому розумінню закономірностей, встановленню зв'язків між подіями та розробці аргументованих висновків.

Алгоритмічне та логічне мислення є ключовими навичками в сучасному світі, особливо в інформаційній епосі, де технології відіграють важливу роль у багатьох сферах життя. Ці навички є необхідними для ефективного розв'язання проблем, розробки програм та алгоритмів, а також для розуміння складних систем. Вони сприяють розвитку аналітичного підходу до вирішення завдань, формуванню структурованого способу мислення та здатності до прогнозування наслідків прийнятих рішень. У контексті цифрової трансформації логічне мислення допомагає адаптуватися до швидких змін, оптимізувати процеси та знаходити інноваційні рішення в різних сферах – від інженерії до бізнес-аналітики. Крім того, розвиток цих навичок позитивно впливає на когнітивні здібності, підвищує гнучкість мислення та сприяє креативності, що є важливим фактором успіху в умовах сучасного конкурентного середовища.

Алгоритмічне мислення базується на здатності людини до аналізу, прогнозування результатів та створення чітких інструкцій для досягнення певної мети. Людина, яка володіє розвиненим алгоритмічним мисленням, вміє розбивати складні завдання на простіші складові, визначати найоптимальніший порядок їх виконання та передбачати можливі труднощі. Ця навичка є надзвичайно важливою в умовах швидких змін, оскільки дозволяє швидко адаптуватися до нових обставин і знаходити нестандартні рішення. До основних аспектів алгоритмічного мислення можна віднести:

1. Розуміння проблеми. Здатність аналізувати складні завдання та розбивати їх на менші, більш керовані компоненти.

2. Розробка алгоритму. Здатність розробляти послідовність конкретних кроків, які потрібно виконати для досягнення мети.

3. Оцінка та вдосконалення. Здатність оцінювати та вдосконалювати алгоритм на основі його ефективності та результатів (Кошелєв, Пасічник, 2017).

Логічне мислення передбачає здатність правильно оперувати інформацією, аналізувати її, виокремлювати ключові аспекти та робити висновки на основі отриманих даних. Воно допомагає уникати хаотичних рішень, сприяє розвитку критичного погляду на поставлені задачі та дозволяє оцінювати їх з різних точок зору. У процесі навчання програмуванню логічне мислення сприяє правильному розумінню принципів роботи алгоритмів, дозволяє виявляти та виправляти помилки, а також оптимізувати створені програми. Основні аспекти логічного мислення включають:

1. Аналіз інформації. Здатність аналізувати, оцінювати та зрозуміти різноманітну інформацію.

2. Виділення залежностей. Здатність визначати зв'язки та залежності між різними фактами або подіями.

3. Розв'язання проблем. Здатність застосовувати логічні правила для ефективного розв'язання проблем і прийняття обґрунтованих рішень (Шнайдер, Назаренко, 2008).

Формування алгоритмічного та логічного мислення є важливим завданням сучасної освіти. Особливої актуальності воно набуває у вивченні програмування, адже створення програм потребує не лише знань мов кодування, а й умінь мислити логічно, вибудовувати правильну послідовність дій та передбачати їхні наслідки. Одним із ефективних засобів розвитку цих навичок є використання навчального середовища Scratch, яке дозволяє учням у доступній формі опановувати основи алгоритмізації та програмування. Цю особливість навчального середовища Scratch було використано Є. Лодатком, О. Мочулою і Г. Яріш – авторами підручника з інформатики для 4 класу – для ознайомлення учнів з можливістю створення проєктів, що не передбачають «написання текстів програм на формалізованих мовах програмування, [а на основі] ... графічних засобів для зображення даних і структур управління» (Лодатко, Мочула, Яріш, 2021, с. 89–90).

Scratch є відкритою програмною платформою, розробленою Массачусетським технологічним інститутом для навчання дітей основам програмування, дозволяючи їм створювати інтерактивні історії, ігри та анімації. Ця платформа особливо цінна в освітньому контексті, адже сприяє розвитку алгоритмічного та логічного мислення серед учнів.

Scratch – це засіб візуального програмування, що впроваджує парадигму об'єктно-орієнтованого програмування та дозволяє створювати анімовані мультимедійні інтерак-

ктивні продукти у вигляді історій, ігор, моделей та іншого. Scratch є навчальною програмою, яку можна завантажити з офіційного сайту за адресою <http://scratch.mit.edu>, або створювати проекти в онлайн-режимі. Важливо відзначити, що у Scratch інтуїтивно зрозумілий інтерфейс, що передбачає вибір мови з кількох десятків доступних, тому програмування може здійснюватися мовою користувача (Дудка, Власій, 2017).

О. Дудка, О. Власій, Н. Магомета зазначають, що Scratch – середовище відкриває можливості для створення інтерактивних засобів, які можуть використовуватися для демонстрації різноманітних природних явищ, історичних подій, а також для створення вітальних листівок, виховних презентацій та інших матеріалів (Дудка, Власій, Магомета, 2018). Це не лише розширює можливості навчального процесу, але й сприяє інтеграції різних дисциплін та використанню компетентнісного підходу до навчання.

Scratch допомагає формувати логічне мислення учнів, оскільки змушує їх структурувати свої думки та плани перед тим, як перетворити їх на функціонуючий код. Учні навчаються визначати послідовності дій для рішення завдань, розбиваючи складні задачі на менші, більш керовані частини. Цей процес, відомий як декомпозиція, є критично важливим вмінням у програмуванні. Scratch також включає у себе використання умовних операторів і циклів, які дозволяють учням використовувати логічне міркування для контролю потоку програми.

Розвиток логічного мислення через Scratch є надзвичайно важливим аспектом у навчанні дітей, оскільки воно сприяє не лише здобуттю технічних навичок, а й формуванню фундаментальних розумових здібностей. Розглянемо детальніше, як це відбувається на практиці.

Одним з основних компонентів логічного мислення є здатність аналізувати задачу та визначати необхідні кроки для її вирішення. Наприклад, у Scratch, якщо учень хоче *створити просту анімацію персонажа*, який ходить, він мусить спочатку розуміти, які рухи повинен виконувати персонаж, та як ці рухи програмувати за допомогою блоків. Учень мусить вирішити, коли персонаж починає рух, як швидко він рухається, та коли зупиняється, що вимагає логічного планування та послідовності кроків (Лодатко, Мочула, Яріш, 2021, с. 106).

Декомпозиція допомагає учням розбивати складні завдання на менші, більш керовані частини. У Scratch, при створенні гри, учень може почати з основної структури – спочатку створити фон, потім додати персонажів, і нарешті, інтегрувати правила гри та різні рівні складності. Цей підхід дозволяє

учням систематично підходити до розробки проекту, зосереджуючись на одному аспекті за раз.

Умовні оператори в Scratch дозволяють учням впроваджувати різні сценарії залежно від певних умов. Наприклад, учень може програмувати персонажа так, що якщо персонаж торкається об'єкта, він отримує очки або втрачає життя. Це спонукає до використання логічного міркування при виборі правильних умов та визначенні наслідків для кожної з них (Дудка, Власій, Магомета, 2018).

Цикли є іншим критичним елементом програмування, який сприяє логічному мисленню. У Scratch, цикли можуть бути використані для повторення певної дії кілька разів. Наприклад, якщо учень хоче, щоб персонаж пройшов певну дистанцію, він може використати цикл з умовою завершення, що спростить код і зробить його більш ефективним.

Scratch також сприяє розвитку критичного мислення через процес налагодження. Учні навчаються аналізувати та коректувати свої проекти, виявляючи та виправляючи помилки. Цей процес допомагає їм краще розуміти логіку програмування і важливість точності у передачі інструкцій. Окрім того, робота з помилками формує навички самостійного пошуку рішень та вчить експериментувати з різними підходами до вирішення завдань. Завдяки цьому діти розвивають терпіння, наполегливість і здатність мислити стратегічно, що є важливими навичками не лише в програмуванні, а й у повсякденному житті.

Планування проектів у Scratch вимагає стратегічного мислення. Учні мають вирішувати, які функції включити, як вони вплинуть на загальний досвід користувача і які ресурси потрібні для реалізації проекту. Це виховує здатність продумувати потреби та наслідки своїх рішень.

Завдяки можливості спілкування в онлайн-спільноті Scratch, учні також розвивають соціальне та колаборативне мислення. Вони обговорюють ідеї, отримують зворотній зв'язок та вчаться від інших, що сприяє глибшому розумінню процесів та різноманітних підходів до розв'язання проблем.

Ці аспекти Scratch роблять його не тільки інструментом для навчання програмуванню, а й платформою для комплексного розвитку когнітивних навичок, які будуть корисними учням у майбутньому.

Один із захопливих прикладів вправи, яку можна створити за допомогою Scratch для учнів 7 класу, це гра «Лабіринт». Цей проект не тільки веселий, але й ефективно розвиває логічне мислення, просторову орієнтацію та навички програмування.

Кроки для створення гри «Лабіринт» у Scratch:

1. Підготовка сцени:
 - Створіть новий проект у Scratch.
 - Виберіть або створіть фон, який буде виглядати як лабіринт. Можна використати інструменти редагування зображення в Scratch для малювання стін лабіринту або завантажити готове зображення.
2. Додавання персонажу:
 - Виберіть спрайт з бібліотеки Scratch або створіть власний. Цей спрайт буде персонажем, яким гравець буде керувати, щоб проходити лабіринт.
3. Програмування руху персонажу:
 - Додайте блоки коду, що дозволяють персонажу рухатись вгору, вниз, вліво та вправо при натисканні відповідних клавіш.
 - Використайте блоки умов, щоб забезпечити зупинку персонажа при зіткненні зі стінами лабіринту.
4. Додавання завдань та перешкод:
 - Введіть додаткові елементи, такі як ключі для відкриття замків або часові обмеження, що додають виклик гри.
 - Можете додати «ворогів» або рухомі об'єкти, які будуть ускладнювати проходження лабіринту.
5. Визначення кінцевої мети гри:
 - Встановіть місце, куди має дістатися персонаж для виграшу. Наприклад, це може бути інший край лабіринту або спеціальна зона.
 - Додайте ефекти та анімації для підвищення захоплюючого ефекту при досягненні цілі.
6. Тестування і налагодження:
 - Запрошуйте інших учнів або вчителів спробувати гру і зберіть зворотний зв'язок. Виправляйте помилки, оптимізуйте код та вносьте необхідні зміни, щоб гра стала більш гладкою та захопливою.
7. Поділіться та обговоріть проект:
 - Опублікуйте гру в онлайн-спільноті Scratch для отримання ширшого зворотного зв'язку.
 - Обговоріть у класі, які стратегії використовувалися, які проблеми виникали під час створення гри, і як вони були вирішені.

Ця вправа не тільки стимулює розвиток логічного мислення, але й навчає учнів основам програмування через практичний досвід. Розробка гри «Лабіринт» (рис. 1) дозволяє учням краще зрозуміти, як алгоритми та програмні структури впливають на поведінку ігрових елементів та їх взаємодію в рамках розв'язання проблем. Крім того,

процес планування та реалізації проекту виховує навички критичного мислення та творчості, а також сприяє розвитку вміння працювати в команді та адаптуватися до змін.



Рис. 1. Ілюстрація для гри «Лабіринт» на Scratch

Ілюстрація, що зображена на рис. 1, відображає лабіринт, персонажа, що намагається його пройти, та різні елементи всередині, такі як ключі та скарбниці. Ця картинка може бути використана для візуалізації гри або як частина навчальних матеріалів.

Створення ігор у Scratch – це чудовий спосіб залучити учнів до активного і змістовного навчання, заохочуючи їх до самостійного дослідження та творчого розв'язання задач (Дудка, Власій, Магомета, 2018).

Алгоритмічне мислення є фундаментальним аспектом навчання програмуванню, і платформа Scratch ідеально підходить для його розвитку у учнів. Ось як Scratch сприяє формуванню алгоритмічного мислення через різноманітні аспекти та активності.

Scratch використовує візуальні блоки для представлення коду, що дозволяє учням легше засвоювати, як програми працюють. Візуальні блоки можна легко перетягувати та з'єднувати, що допомагає учням зрозуміти логічний зв'язок між різними частинами програми. Наприклад, учень може створити простий скрипт, який змушує спрайт пересуватися вперед, коли користувач натискає стрілку вправо. Таке вправління демонструє учням поняття події та реакції у програмуванні (Яценко, 2017).

Алгоритмічне мислення також включає здатність розробляти кроки для вирішення проблеми. У Scratch, діти можуть використовувати алгоритми для керування складними завданнями, такими як навігація спрайта через лабіринт. Учень повинен визначити, в якому порядку розміщувати блоки для перевірки зіткнень, руху та взаємодії з об'єктами, що вимагає логічного планування та аналізу.

Цикли та умовні оператори є ключовими концепціями в алгоритмічному мисленні. У Scratch, учні можуть використовувати блоки

«повторювати», щоб змусити дії відбуватися багато разів. Наприклад, для анімації ходьби спрайта можна використати цикл, який повторює команду руху кілька разів. Умовні оператори, такі як «якщо», дозволяють учням виконувати дії на основі специфічних умов, наприклад, змінювати напрямок руху спрайта при зіткненні зі стіною.

Під час створення власних проектів у Scratch, учні застосовують алгоритмічне мислення для планування та реалізації своїх ідей. Вони дізнаються, як розбивати великі завдання на менші, більш керовані частини, як розв'язувати проблеми та оптимізувати свої рішення. Це не тільки покращує їхні програмувальні навички, але й розвиває важливі навички критичного мислення та креативності (Яценко, 2017).

Змінні та функції є ще двома важливими аспектами програмування, які Scratch допомагає освоїти. Учні вчаться використовувати змінні для зберігання даних, що можуть змінюватись, наприклад, для підрахунку очок у грі. Функції в Scratch дозволяють учням групувати код у повторювані блоки, що є корисним для зменшення повторення коду та збільшення його читабельності.

На закінчення, Scratch забезпечує платформу для обміну проектами та спільної роботи, що дозволяє учням вчитися від інших та розвивати навички співпраці. Обмін алгоритмами та рішеннями сприяє глибшому розумінню алгоритмічних принципів та покращує навички розв'язання проблем.

Як можемо бачити, проектна діяльність у Scratch сприяє не тільки логічному та алгоритмічному мисленню, але й розвитку критичного мислення. Учні навчаються аналізувати помилки та несподівані результати своїх програм, вдосконалюючи їх для досягнення кращих результатів. Цей процес, відомий як налагодження, вчить учнів терпінню та наполегливості, а також спонукає до глибшого розуміння того, як програми і алгоритми функціонують.

Scratch має активну онлайн-спільноту, де учні можуть ділитися своїми проектами, отримувати зворотній зв'язок та вчитися на прикладах одне одного. Це сприяє не тільки технічному, але й соціальному навчанню. Участь у спільноті також вчить учнів співпраці, дозволяючи їм працювати разом над проектами, обмінюватися ідеями та розв'язувати спільні проблеми (Корецька, Шлянчак, 2017). Це означає, що Scratch — це не просто інструмент для програмування наодинці. Це цілий онлайн-світ, де діти з різних куточків планети можуть показати іншим, що вони створили — свої ігри, історії чи мультики. Дивлячись на роботи інших, учні надихаються, знаходять нові цікаві ідеї та розуміють, як можна втілити той чи ін-

ший задум. Вони можуть навіть взяти чийсь проект за основу і додати щось своє, щоб зробити його ще кращим або зовсім іншим. Інші учасники спільноти можуть залишати коментарі до проектів: написати, що їм сподобалося, або дати дружню пораду, як можна щось покращити. Такий обмін думками допомагає авторам побачити свої роботи збоку, вчитися на відгуках та вдосконалювати свої навички. Крім того, діти вчаться працювати разом. Вони можуть об'єднувати зусилля для створення спільних проектів, ділитися завданнями та разом шукати найкращі рішення, так само як це роблять дорослі у командній роботі. Коли учень бачить, що його робота подобається іншим, що її коментують чи використовують для створення нових проектів, це дуже підбадьорює та надихає. Це дає відчуття успіху і бажання творити далі, пробувати щось нове та складніше. Таким чином, ця онлайн-спільнота робить навчання програмуванню більш живим, цікавим та соціальним, допомагаючи дітям розвивати не лише технічні вміння, але й навички спілкування та взаємодії.

Ефективність використання Scratch як освітнього інструменту значною мірою залежить від обраного педагогічного підходу та методичної грамотності вчителя. Важливо не просто показати учням, як користуватися блоками, а створити середовище, де їхнє мислення активно розвивається через проектну діяльність. Для цього вчителям варто зосередитись на проектному підході, заохочуючи учнів створювати власні значущі ігри, анімації чи історії, що суттєво підвищує мотивацію порівняно з виконанням механічних вправ. Водночас необхідно враховувати різний темп засвоєння матеріалу та рівень підготовки учнів, пропонуючи завдання різної складності або додаткові виклики — тобто застосовуючи диференціацію.

Дуже корисною є організація співпраці між учнями, наприклад, у форматі парного програмування або роботи в малих групах. Це не лише сприяє обміну ідеями та взаємонавчанню, але й розвиває комунікативні навички, вміння пояснювати свою логіку та спільно шукати рішення. При цьому важливо робити акцент не лише на фінальному результаті, а й на самому процесі створення проекту: обговорювати з учнями етапи планування, вибір алгоритмів, труднощі, що виникали, та шляхи їх подолання.

Оцінювання в Scratch також має виходити за межі простої перевірки працездатності програми. Щоб оцінити саме розвиток мислення, варто аналізувати *логіку* та алгоритмічну структуру створеного коду, навички вирішення проблем, які учень демонстрував під час роботи, його креативність та оригінальність ідеї, а також здатність пояснити

власний код і прийняті рішення. Для такого комплексного оцінювання вчитель може використовувати спостереження за роботою учнів, детальний аналіз їхніх проектів, усні опитування, методи взаємооцінювання або розроблені рубрики оцінювання, що містять критерії, пов'язані з мисленнєвими навичками.

Крім того, Scratch відкриває широкі можливості для міжпредметної інтеграції, демонструючи універсальність алгоритмічного підходу. Наприклад, на уроках математики учні можуть створювати програми для візуалізації геометричних фігур чи розв'язання задач; на природничих науках – моделювати фізичні явища чи біологічні процеси; на уроках мови та літератури – розробляти інтерактивні оповідання чи вікторини; а на мистецтві – експериментувати з генеративним мистецтвом чи анімацією. Така інтеграція не лише поглиблює знання з різних предметів, але й показує практичне застосування програмування у різноманітних сферах.

Застосування цих методично виважених підходів до викладання дозволить максимально реалізувати освітній потенціал платформи Scratch для формування логічного та алгоритмічного мислення учнів, готуючи їх до викликів сучасного технологічного світу.

Таким чином, Scratch є чудовим інструментом для розвитку алгоритмічного та логічного мислення у школярів. Його візуальний та інтуїтивно зрозумілий інтерфейс, активна спільнота та зосередженість на проектній діяльності роблять його ідеальним вибором для введення у світ технологій та програмування.

Висновки і перспективи подальших досліджень. Використання середовища Scratch у навчальному процесі є надзвичайно ефективним інструментом для розвитку алгоритмічного та логічного мислення учнів. Його візуальний інтерфейс, доступність і багатофункціональність сприяють засвоєнню базових понять програмування, формуванню критичного мислення, креативності та навичок вирішення проблем. Важливим аспектом є також розвиток соціальних і комунікативних навичок завдяки інтеграції в онлайн-спільноту Scratch.

Перспективи подальших досліджень у цій галузі пов'язані з розробкою нових методичних підходів до інтеграції Scratch у навчальні програми, дослідженням його впливу на академічну успішність учнів і розвиток їхніх когнітивних здібностей. Особливо актуальними є дослідження, спрямовані на адаптацію платформи для роботи з учнями різного віку та рівня підготовки. У подальшому важливо також дослідити можливості поєд-

нання Scratch із іншими технологічними інструментами для створення комплексних навчальних середовищ.

Подальші дослідження також можуть бути зосереджені на розробці методичних рекомендацій для вчителів, які сприятимуть більш ефективному впровадженню Scratch у шкільну програму. Аналіз навчальних досягнень учнів, які працюють із цією платформою, дозволить виявити сильні та слабкі сторони її використання, а також визначити шляхи покращення навчального процесу. З огляду на швидкий розвиток технологій, важливим є також вивчення можливостей поєднання Scratch із сучасними інструментами штучного інтелекту, що відкриє нові горизонти для розвитку алгоритмічного та логічного мислення в освіті.

Список бібліографічних посилань.

- Дудка, Власій, 2017 – Дудка, О.М., Власій, О.О. (2017). Особливості вивчення програмування на Scratch. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво. Науковий журнал*, 26: 81–87.
- Дудка, Власій, Магомета, 2018 – Дудка, О.М., Власій, Н.М., Магомета, Н.М. (2018). Реалізація компетентнісного підходу до вивчення програмування на Scratch. *Відкрите освітнє е-середовище сучасного університету*, 5: 88–96.
- Корецька, Шлянчак, 2017 – Корецька, В., Шлянчак, С. (2017). Використання середовища Scratch у процесі підготовки майбутніх учителів. *Vzdelávanie a spoločnosť*, II: 161–173.
- Кошелев, Пасічник, 2017 – Кошелев, О. Пасічник, Н. (2017). Теоретико-методичні основи розвитку алгоритмічного мислення молодших школярів. *Молодь і ринок*, 8: 60–64.
- Лодатко, Мочула, Яріш, 2021 – Лодатко, Є.О., Мочула, О.В., Яріш, Г.П. (2021). Інформатика: підручник для 4 кл. закладів загальної середньої освіти. Тернопіль: Навчальна книга – Богдан. 160 с.
- Шнайдер, Назаренко, 2008 – Шнайдер, Л.І., Назаренко, С.Г. (2008). Розвиток логічного мислення учнів. *Математика в школах України*, 14/15: 2–10.
- Яценко, 2017 – Яценко, О.І. (2017). Середовище програмування «Scratch»: аналіз можливостей використання з метою формування інформативних компетентностей вчителя початкової школи. *Актуальні питання сучасної інформатики*, 5: 276–278.

References

- Dudka O.M., Vlasii O.O. (2017). Features of learning programming in Scratch. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво. Науковий журнал*, 26: 81–87. [in Ukr.].
- Dudka O.M., Vlasii N.M., Mahometa N.M. (2018). Implementation of the competence-based approach to learning programming in Scratch. *Open Educational Environment of the Modern University*, 5: 88–96. [in Ukr.].
- Koretska V., Shlianchak S. (2017). Using the Scratch environment in the process of training future teachers. *Education and Society*, II: 161–173. [in Ukr.].
- Koshelev O., Pasichnyk N. (2017). Theoretical and methodological foundations of the development of algorithmic thinking in younger students. *Youth and the Market*, 8: 60–64. [in Ukr.].
- Lodatkо, E.O., Mochula, O.V., Yarish, G.P. (2021). Informatics: a textbook for 4th grade of general secondary education institutions. Ternopil: Textbook – Bohdan. 160 p.

Shnaider L.I., Nazarenko S.H. (2008). Development of students' logical thinking. *Mathematics in Schools of Ukraine*, 14/15: 2–10. [in Ukr.].

Yatsenko O.I. (2017). The Scratch programming environment: an analysis of its use for forming the informa-

tiveness competencies of a primary school teacher. *Current Issues in Modern Informatics*, 5: 276–278. [in Ukr.].

VELYKDAN Yuliya

senior lecturer at the Department of Theory and Methodology of Technological Education and Computer Graphics, Hryhorii Skovoroda University in Pereiaslav

SCRATCH'S POSSIBILITIES IN FORMING ALGORITHMIC AND LOGICAL THINKING IN ELEMENTARY SCHOOL STUDENTS

Summary. The article explores the educational potential of the Scratch platform for forming algorithmic and logical thinking in students who are mastering the basics of programming. Scratch, developed by the Massachusetts Institute of Technology, is a visual platform that allows you to create interactive games, animations, and other creative projects. The platform's tools contribute to the development of students' cognitive abilities through involvement in project activities that require systematic planning, analysis, and application of key programming principles.

Particular attention is paid to algorithmic thinking, which is formed through the use of loops, conditional operators, variables, and functions. Students learn to break complex tasks into small subtasks, decomposing them for effective execution. The process of debugging projects allows you to improve critical thinking by analyzing mistakes made and finding ways to eliminate them. Scratch also helps to form strategic thinking by integrating a creative approach to problem solving.

The article describes an example of creating a game "Labyrinth", which demonstrates the effectiveness of Scratch in learning. Students go through the stages of plan-

ning, implementing and testing the game, using algorithms, structuring actions and controlling their execution. This process contributes to the development of spatial thinking, problem-solving skills and creativity.

The study confirms that Scratch is an effective tool not only for teaching the basics of programming, but also for the comprehensive development of logical thinking, cognitive and social competencies. The Scratch online community provides students with the opportunity to collaborate, exchange ideas, and receive feedback, which contributes to their further professional and personal growth. Using the platform in the modern educational process creates conditions for interactive learning and comprehensive development of students.

Keywords: programming; algorithm; computer science; technology; critical thinking; Scratch programming environment; Scratch game projects; New Ukrainian School.

Одержано редакцією 02.02.2025
Прийнято до публікації 18.02.2025