

Сумський державний педагогічний університет імені А. С. Макаренка

Фізико-математичний факультет

Кафедра інформатики

Плющик Володимир Володимирович

**АЛГОРИТМІЧНЕ МИСЛЕННЯ ТА ЙОГО РОЗВИТОК НА УРОКАХ
ІНФОРМАТИКИ**

Спеціальність: 014 Середня освіта (Інформатика)

Галузь знань: 01 Освіта/Педагогіка

Кваліфікаційна робота
на здобуття освітнього ступеня магістра

Науковий керівник:

_____ В.Г.Шамоня

кандидат фізико-математичних наук,
доцент

«___» _____ 20__ року

Виконавець:

_____ В.В.Плющик

«___» _____ 20__ року

Суми 2025

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. Теоретико-методичні засади формування алгоритмічного мислення учнів закладу загальної середньої освіти	8
1.1. Алгоритмічне мислення як ключ до пізнання цифрового світу: сутність, зміст	8
1.2. Психолого-педагогічні умови розвитку алгоритмічного мислення в процесі вивчення інформатики	15
1.3. Методичні підходи до розвитку алгоритмічного мислення в шкільному курсі інформатики	24
РОЗДІЛ 2. Проєктування освітнього середовища для розвитку алгоритмічного мислення: досвід і перспективи	33
2.1. Умови організації навчальної діяльності здобувачів освіти для розвитку алгоритмічного мислення	33
2.2. Інформаційно-освітні інструменти і ресурси для формування алгоритмічного мислення учнів	44
2.3. Практична реалізація фрагментів уроків з формування алгоритмічного мислення здобувачів освіти 9 класів.....	50
ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62
ДОДАТКИ.....	71

СПИСОК УМОВНИХ СКОРОЧЕНЬ

ЗЗСО – заклад загальної середньої освіти.

НУШ – Нова українська школа.

fizmat@sspu.edu.ua
суворо дотримуйтесь
правил академічної
добросовісності

ВСТУП

Актуальність теми. Цифрові технології розвиваються дуже швидко, саме тому змінюється і сам навчальний процес: здобувачі освіти мають не лише вміти працювати з інформацією, а й розвивати здатність мислити алгоритмічно. У XXI столітті ключове завдання сучасного закладу загальної середньої освіти (ЗЗСО) – це не тільки засвоєння знань, а й формування здатності до логічного аналізу, структурованого мислення та вміння будувати послідовно дії для вирішення завдань різної складності. Алгоритмічне мислення – головна і основна складова цифрової грамотності. Вона спонукає учнів орієнтуватися у різних інформаційних середовищах і приймати ефективні рішення.

Зміни у змісті шкільної освіти, зокрема в інформатичній предметній галузі, сприяють створенню нових педагогічних підходів і впровадженню інноваційних форм і методів у навчальний процес. Під час реформування Нової української школи (НУШ) зростає потреба у створенні особливих умов навчання, які стимулюють учнів до самостійного пошуку, експериментування, планування дій та оцінювання власної діяльності. Саме алгоритмічне мислення забезпечує ці складні процеси, виступаючи вагомим інструментом як для навчання, так і для життя в умовах цифрового та сучасного суспільства.

Враховуючи вищезазначені особливості, можна стверджувати, що актуальності набуває педагогічна розробка уроків, які спрямовані на розвиток алгоритмічного мислення здобувачів освіти. Необхідно врахувати психолого-педагогічні чинники, можливість використання цифрових ресурсів і створення навчального середовища, яке мотивує, для формування ключових компетентностей учнів та підготовки їх до свідомого існування у цифровому просторі. Це зумовлено необхідністю підготовки молодого покоління до життя в умовах цифрової трансформації суспільства, де здатність до логічного аналізу, структурованого мислення та розв'язання проблем набувають ключового значення у повсякденні.

Науковці звертають увагу на важливості розвитку алгоритмічного мислення у школярів. Наприклад, у дисертаційному дослідженні Нікітенко О. В. зазначено, що алгоритмічне мислення – основа для формування алгоритмічної культури, яка сприяє розвитку пізнавальних здібностей учнів та їхньої здатності до самостійного навчання [1].

А у статті Зіятдінова Р. та Муси С. описано можливості розвитку алгоритмічного мислення у початковій освіті. На їхню думку, воно передбачає виконання послідовних логічних дій чи операцій та формування в учнів здатності до планування, аналізу й розв'язку задач різної складності за допомогою алгоритмів. Його формування базується на використанні простих логічних операціях, що мають чітку послідовність, подібно до міні-алгоритмів. Таке мислення не лише полегшує засвоєння різних понять, а й закладає основи для подальшого вивчення інформатики в шкільному середовищі [2].

Крім того, такі науковці, як Крупа А. та Мар'єнко В., вважають, що алгоритмічне мислення впливає на формування поглядів учнів, оскільки інформація в сучасному світі створюється на основі алгоритмів, і розуміння цих процесів є необхідним для критичного сприйняття дійсності [3].

Таким чином, розвиток алгоритмічного мислення та його впровадження у освітній процес є не лише актуальним процесом, але й необхідною умовою підготовки учнів до ефективної діяльності в інформаційному суспільстві. Формування такої здатності сприяє адаптації школярів до вимог цифрової епохи, створює основу для розвитку критичного мислення, вирішення нестандартних задач і самостійного прийняття рішень у невизначених ситуаціях.

Мета роботи: дослідити теоретичні засади, психолого-педагогічні умови та методичні підходи до формування алгоритмічного мислення в учнів закладу загальної середньої освіти на уроках інформатики.

Мета роботи передбачає виконання наступних **завдань**:

1. Розкрити сутність поняття «алгоритмічне мислення» та обґрунтувати його значення в умовах цифрової трансформації освіти;

2. Проаналізувати психолого-педагогічні умови розвитку алгоритмічного мислення у школярів;
3. Охарактеризувати методичні підходи та сучасні освітні технології, що сприяють формуванню алгоритмічного мислення на уроках інформатики;
4. Визначити особливості проектування освітнього середовища для розвитку алгоритмічного мислення учнів;
5. Розробити дидактичну модель фрагментів уроків з використанням технологій тривимірного моделювання у середовищі Blender, спрямованих на формування алгоритмічного мислення здобувачів освіти 9 класів закладу загальної середньої освіти.

Об'єкт дослідження: формування алгоритмічного мислення учнів у ЗЗСО.

Предмет дослідження: особливості формування алгоритмічного мислення учнів у процесі вивчення інформатики в ЗЗСО.

Методи дослідження: теоретичні методи досліджень (історичний метод, сходження від абстрактного до конкретного, узагальнення), методи емпіричного дослідження (вивчення документації), тестування, економіко-статистичні (графічні методи).

Теоретична значущість полягає в поглибленні наукових уявлень про сутність, структуру та умови формування алгоритмічного мислення здобувачів освіти в процесі вивчення інформатики, а також в обґрунтуванні методичних підходів до його розвитку в умовах цифровізації освітнього середовища.

Практичне значення кваліфікаційної роботи полягає в тому, що її теоретичні положення, психолого-педагогічні висновки та методичні розробки можуть бути використані для вдосконалення освітнього процесу в закладах загальної середньої освіти. Запропоновані підходи до формування алгоритмічного мислення, зокрема через інтеграцію тривимірного моделювання у середовищі Blender, сприяють підвищенню мотивації учнів, розвитку їхньої здатності до логічного аналізу, планування та критичного мислення. Реалізація розроблених фрагментів уроків та освітніх моделей

створює умови для ефективного впровадження інноваційних технологій у навчання інформатики, що відповідає вимогам цифрової трансформації освіти та компетентнісного підходу Нової української школи.

Гіпотеза. Якщо освітній процес включає системну роботу з тривимірним моделюванням у середовищі Blender, побудовану на поетапному формуванні алгоритмічного мислення через практичні завдання, рефлексію та міжпредметну інтеграцію, то учні демонструють не лише зростання рівня алгоритмічного мислення, а й підвищення мотивації, здатності до самостійного планування та критичного аналізу.

Апробація результатів роботи. Основний зміст дипломної роботи відображено у науковій статті та тезах:

- 1. ОРГАНІЗАЦІЯ НАВЧАЛЬНОЇ ДІЯЛЬНОСТІ УЧНІВ СУЧАСНОГО ЗАКЛАДУ ЗАГАЛЬНОЇ СЕРЕДНЬОЇ ОСВІТИ В ЦИФРОВОМУ СЕРЕДОВИЩІ ДЛЯ РОЗВИТКУ АЛГОРИТМІЧНОГО МИСЛЕННЯ**
- 2. ФОРМУВАННЯ АЛГОРИТМІЧНОГО МИСЛЕННЯ УЧНІВ ЗАКЛАДУ ЗАГАЛЬНОЇ СЕРЕДНЬОЇ ОСВІТИ ЗАСОБАМИ ТРИВИМІРНОГО МОДЕЛЮВАННЯ У СЕРЕДОВИЩІ BLENDER**

Структура роботи. Магістерська робота складається зі вступу, двох розділів, списку використаних джерел, додатків А – Д. Загальний обсяг роботи – 92 сторінки, з них – 61 сторінка – основний текст, список використаних джерел – на 9 сторінках.

РОЗДІЛ 1

ТЕОРЕТИКО-МЕТОДИЧНІ ЗАСАДИ ФОРМУВАННЯ АЛГОРИТМІЧНОГО МИСЛЕННЯ УЧНІВ ЗАКЛАДУ ЗАГАЛЬНОЇ СЕРЕДНЬОЇ ОСВІТИ

1.1. Алгоритмічне мислення як ключ до пізнання цифрового світу: сутність, зміст

У другій половині XX століття зростання обчислювальної техніки створило основу для нового виду мислення – алгоритмічного, яке стало важливою складовою поняття «комп'ютерне мислення». Саме цей термін популяризувала дослідниця Вінг Дж. у 2006 році, використовуючи його як спосіб вирішення проблем та задач, проєктування систем різної складності та розуміння поведінки, що є основою інформатики. Вона стверджує, що обчислювальне мислення передбачає розв'язання проблем, проєктування систем і розуміння людської поведінки, спираючись на принципи, що є фундаментальними для інформатики [4].

Сучасні інформаційні технології відкривають перед людством величезний спектр можливостей. Та на практиці часто такі потенціали залишаються нереалізованими через недостатню готовність людського мислення сприймати принципи функціонування цифрового середовища. Зокрема, не кожна людина здатна інтуїтивно мислити алгоритмічно, що є основою для повноцінної взаємодії з різноманітними технологіями.

Саме тому основне завдання освіти – це цілеспрямоване формування алгоритмічного мислення як основи цифрової грамотності людства. Його розвиток передбачає побудову та формування логічних структур мислення – здатності аналізувати, узагальнювати, планувати послідовно дії. За даними психологів, найсприятливіший період для закладення таких розумових здібностей є молодший шкільний вік. Це пов'язано з тим, що у дітей починають активно розвиватися пізнавальні процеси та здатність до системного мислення.

Останні десятиліття спостерігається стрімкий розвиток електронно-обчислювальної техніки, що призводить до виникнення нового поняття – «інформаційна грамотність». Цей термін охоплює не лише здатність працювати з комп'ютером, а й уміння розуміти суть поданої інформації, правильно її інтерпретувати, ефективно організовувати й планувати свою діяльність у руслі інформаційного потоку. У такому значенні інформатика постає як міждисциплінарна наука, яка вивчає структуру, властивості, джерела та способи використання інформації, а також методи її обробки, збереження чи її передачі [5].

Інформація, з наукової точки зору, є сукупністю сигналів чи повідомлень, що надходять до певної системи ззовні (вхідна інформація), зберігаються в її межах (внутрішня інформація) або передаються назовні (вихідна інформація). Уміння працювати з такою інформацією є ключовою необхідністю сучасної освітньої парадигми. Вона формує у здобувачів освіти здатність приймати правильні рішення в умовах складного інформаційного потоку.

Фундаментальні знання інформаційної грамотності нерозривно пов'язані з поняттям «алгоритм». Історично цей термін виник давно. Цей процес припав на активний розвиток арифметики та цифрових обчислень, але вже сьогодні його значення значно розширилося. Поняття «алгоритм» вже прийнято розглядати як універсальний засіб будь-якої діяльності, де чітко визначені інструкції чи кроки ведуть до досягнення конкретного та визначеного результату. Саме алгоритмічне мислення, тобто здатність логічно структурувати дії, шукати ефективні шляхи досягнення мети, стає необхідним умінням людини для успішної адаптації в сучасному цифровому суспільстві [6].

Проблему формування алгоритмічного мислення активно вивчають як теоретики, так і практики освітнього шкільного процесу. Зокрема, значний внесок зробили українські науковці Гладун М. А. та Морзе Н. В., які акцентують увагу на педагогічних умовах розвитку логічних структур мислення [7]. У свою чергу, доцент Волошинов С. А. аналізує інформаційні технології як основу формування сучасного типу мислення учнів [8].

Вчені трактують алгоритмічне мислення як впорядковану сукупність дій, прийомів і стратегій, що дають змогу знаходити рішення як у теоретичних, так і практичних ситуаціях. Результатом такої розумової діяльності є алгоритм – структурований шлях до досягнення конкретного результату [9].

Алгоритмічне мислення, на думку Копаєва О., визначається як організована система послідовних розумових дій, що забезпечує ефективне вирішення різноманітних задач – від абстрактних до конкретних. Головною ознакою такого типу мислення є здатність людини логічно формулювати послідовність дії у вигляді алгоритму, що передбачає не лише раціональне структурування проблеми, а й оптимізацію шляхів її розв’язання.

Це мислення дозволяє спроектувати універсальні рішення, які застосовуються до різних видів ситуацій, що робить його важливим компонентом не лише для освіти в галузі інформатики, а й загальної освітньої підготовки. Саме алгоритмічний підхід сприяє формуванню аналітичних навичок, навчає учнів мислити кроками, передбачати можливий результат і шукати найефективніші шляхи дій чи розв’язку проблем [10].

У цифрову епоху алгоритмічне мислення виконує не лише освітню, а й вагомую специфічну соціокультурну функцію. Воно дозволяє розуміти, яким чином працюють алгоритми в соціальних мережах, електронній комерції або під час прийняття рішень на державному рівні. Завдяки цьому учні та користувачі можуть критично оцінювати будь-яку інформацію, формувати усвідомлену позицію і не залишатися пасивними споживачами, а носіями активного вибору та відповідального прийняття рішень [11].

Важливо, що алгоритмічне мислення виходить за межі STEM-предметів і набуває значення навіть під час вивчення гуманітарних дисциплін. Воно стає інструментом аналізу мовних структур, соціальних моделей, філософських дилем – допомагаючи виокремлювати логічні компоненти, осмислювати контексти та їх наслідки. Українські дослідники визначають «культуру алгоритмічного мислення» як здатність застосовувати математичні моделі та

логіку в гуманітарному аналізі, що відкриває можливості для нових підходів в сфері освіти.

Оскільки алгоритмічне мислення є фундаментальним компонентом комп'ютерної грамотності та ключовою складовою розвитку навичок ХХІ століття, воно не зводиться лише до написання програмного коду, а передбачає розвиток логіки, послідовного планування та системного підходу до розв'язання проблем. В основі алгоритмічного мислення лежить чотири основні компоненти: декомпозиція, виявлення шаблонів, абстракція та створення алгоритмів. Кожен із цих елементів забезпечує формування цілісної картини мислення, необхідної для роботи з інформаційними технологіями та складними системами.

Декомпозиція – це процес розбиття складної задачі або проблеми на менші блоки чи завдання. Завдяки цьому етапу учні навчаються вирішувати проблему поетапно, що полегшує її розв'язання. Цей підхід особливо корисний для молодших школярів, які ще не можуть керувати великим обсягом інформації одночасно.

Декомпозиція дає змогу краще зрозуміти, які дії необхідно виконати, і в якому порядку. Наприклад, якщо необхідно створити простий комп'ютерний анімаційний сюжет, то його можна розбити на частини: вибір фону, створення персонажів, написання сценарію руху.

Українські дослідники наголошують на важливості цього компонента при вивченні наук інформатичного циклу, оскільки він сприяє розвитку послідовного мислення і структурованого підходу до розв'язання задач будь-якої складності [12].

Другий важливий елемент – це здатність розпізнавати повторювані структури, шаблони або закономірності в інформації чи процесах. Це зменшує складність завдань і сприяє ефективному пошуку рішень у схожих завданнях.

Учні, розвиваючи цю навичку, можуть бачити подібності між завданнями, що значно спрощує навчання та робить його цікавим. Наприклад, діти можуть виділити схему повторення дій у грі та оформити її як функцію або

процес, що оптимізує подальший код. Саме цю здатність покладено в основу структурного підходу до програмування.

Українські освітні програми також включають розвиток цієї навички в під час вивчення предметів початкової школи. Наприклад, «Модельна навчальна програма інформатики для 5–6 класів (НУШ)» акцентує увагу здобувачів освіти на виявленні шаблонів – схожих елементах або особливостях, що дозволяє прогнозувати результати або знаходити однакові рішення для різних задач [13].

Аналізуючи інформацію з платформи Dystosvita можна стверджувати, що знаходження шаблонів є критично важливим етапом для побудови алгоритмів, особливо під час вивчення інформатики. Наприклад, учні ідентифікують повторювані закономірності в поведінці об'єктів у середовищі програмування Scratch або при вивченні цифрових моделей геометричних фігур, що дозволяє їм рішення завдань автоматизувати та ефективніше оперувати повторюваними діями [14].

Абстракція – це процес виділення суттєвих характеристик об'єкта або проблеми, у ході якого ігноруються несуттєві деталі. Завдяки абстракції учні можуть концентруватися на головному, не відволікаючись на зайву інформацію.

Наприклад, у програмуванні це означає створення загальних структур даних або шаблонів, які можуть застосовуватися до різних задач. Абстракція допомагає учням не просто запам'ятовувати кроки, а розуміти їхню логіку та значення.

Українські науковці зазначають, що абстрактне мислення є однією з найскладніших навичок, яку можна розвивати через алгоритмічне мислення, і саме цей компонент стимулює креативність [14].

Заключний компонент структури алгоритмічного мислення – це розробка послідовності інструкцій або правил, які дозволяють автоматизувати розв'язання певної задачі. Алгоритми ж можуть бути виражені у вигляді тексту, блок-схем, псевдокоду чи програмного коду.

На цьому етапі учні навчаються формалізувати свої ідеї та втілювати їх у зрозумілі, чіткі інструкції, які може виконати як людина, так і машина. Це сприяє розвитку логіки, точності та системності.

Створення алгоритмів на уроках інформатики часто відбувається з використанням сучасних освітніх платформ, зокрема серед молодших школярів певне місце займає середовище Scratch або уроки unplugged (без комп'ютера), що дозволяють навчатися через гру та моделювання [15].

Алгоритмічне мислення є основою цифрової грамотності, що дозволяє учням розвивати навички аналізу, структурування та вирішення проблем. Його формування відбувається завдяки різним науковим підходам, які включають емпіричні дослідження, педагогічні методики та технологічні інструменти.

Одне із важливих досліджень у цій сфері — оцінка алгоритмічного мислення через задачу Cross-Array Task, яка була проведена Адорні Г. та Піатті А. у 2024 році. Це дослідження аналізує здатність учнів розробляти алгоритми для вирішення задач, використовуючи як традиційні, так і цифрові методи. Впровадження цифрових платформ, таких як віртуальний Cross Array Task, дозволяє автоматизувати оцінку навичок та адаптувати навчальні матеріали відповідно до рівня підготовки учнів [16].

Ще одне значне дослідження стосовно розвитку навичок алгоритмічного мислення в освіті було проведено цими ж науковцями разом з колегою Лутц Е. У ньому розглядається ефективність різних підходів до навчання, включаючи використання жестових методів та блокового програмування [17].

Незважаючи на значний прогрес у дослідженнях алгоритмічного мислення, його впровадження в освітній процес України стикається з великою кількістю викликів. Одним із ключових аспектів є недостатня підготовка педагогів до інтеграції алгоритмічного мислення у навчальні програми. Дослідження показують, що ефективне використання таких платформ, як Scratch, потребує проходження спеціалізованих тренінгів чи курсів для вчителів, які дозволяють їм адаптувати методики викладання до сучасних цифрових реалій [18].

Ще одним бар'єром є обмеженість матеріально-технічної бази у багатьох навчальних закладах. Впровадження алгоритмічного мислення передбачає доступ до комп'ютерів, програмного забезпечення та стабільного інтернет-з'єднання, що не завжди є можливим у школах.

Важливим аспектом є також відсутність єдиної освітньої стратегії щодо розвитку алгоритмічного мислення. Хоча в Україні існують окремі напрямки, такі як інтеграція STEM-освіти та цифрових технологій у навчальний процес, але системний підхід до формування алгоритмічного мислення ще потребує доопрацювання [19].

Отже, алгоритмічне мислення виступає основою для розуміння цифрових процесів. Саме воно допомагає людині структурувати реальність, виявляти закономірності, створювати моделі та передбачати наслідки дій у технологічному світі. Через декомпозицію, абстракцію та створення алгоритмів мислення стає більш впорядкованим, цілеспрямованим, а отже – здатним до ефективного пізнання цифрових явищ і систем.

Сутність алгоритмічного мислення полягає в умінні структурувати дії, виявляти закономірності, абстрагувати несуттєве та формалізувати розв'язання задач у вигляді чітких послідовностей. Воно формує новий тип інтелектуальної культури – таку, що дозволяє діяти впевнено у світі цифрових викликів. Без цієї навички неможливо повноцінно взаємодіяти з сучасними технологіями, оскільки вони самі побудовані на алгоритмах. Саме тому розвиток алгоритмічного мислення – ключ до успішної цифрової трансформації суспільства.

Алгоритмічне мислення набуває міждисциплінарного значення, стаючи ключем до опанування логіки сучасного світу – світу, що функціонує за алгоритмами, керується даними і потребує раціонального мислення як основи прийняття обґрунтованих рішень.

Попри виклики, дослідження свідчать про позитивний вплив алгоритмічного мислення на розвиток критичного мислення та навичок вирішення проблем. Його інтеграція у навчальні програми може сприяти

підготовці учнів до сучасного цифрового суспільства, що зможе підвищити їхню конкурентоспроможність на ринку праці.

1.2. Психолого-педагогічні умови розвитку алгоритмічного мислення в процесі вивчення інформатики

Одне із головних завдань шкільної освіти – всебічний і збалансований розвиток дитини як індивідуальності. У цьому контексті ключову роль відіграє формування основ психологічної діяльності – різних типів мислення, а також уваги й пам'яті. Ці когнітивні складові процесу мислення забезпечують успішне засвоєння знань під час вивчення предметів шкільного циклу. Здатність мислити дозволяє дитині встановити психічний баланс. Вона навчається міркувати, оцінювати різні ситуації, приймати нестандартні рішення й знаходити оригінальні підходи до вирішення завдань чи проблем, що позитивно впливає на її інтелектуальний та особистісний ріст.

У початкових класах відбувається природний перехід від конкретно-образного до логічно-понятійного мислення. Така трансформація є основою для здобуття навичок аналітичного підходу до реальних і навчальних проблем. Серед когнітивних процесів особливої уваги заслуговує алгоритмічне мислення. Його розвиток сприяє чіткому й ефективному виконанню дій у типових або повторюваних обставинах, формує вміння дотримуватися послідовності, передбачати наслідки та організовувати власну діяльність за логічною структурою.

Розвиток алгоритмічного мислення – ключове завдання НУШ. Уміння створювати й виконувати алгоритми є важливим метапредметним результатом навчання. Учні мають навчитися планувати, утримувати план у пам'яті та діяти відповідно до нього для досягнення поставленої мети [20].

Питання формування в учнів загальних підходів до мислення виходить за межі суто методичної площини – це також ключова дидактична та психологічна задача. Її вирішення безпосередньо впливає на розвиток інтелектуальних

здібностей, які є важливими для опанування будь-якої шкільної дисципліни та підготовки до майбутньої професійної діяльності чи особистісного росту.

Побудова розумових структур, які забезпечують вирішення типових завдань, базується на поступовому закріпленні правил дій у мисленнєвій діяльності. Цей процес передбачає послідовне оволодіння знанням про об'єкт, усвідомлення суті задачі, виявлення ключових властивостей, що впливають на способи розв'язання, засвоєння стратегій та побудову власних шляхів, які ведуть до досягнення мети. Згодом ці стратегії переходять у стабільні когнітивні схеми. У результаті чого в нервовій системі вибудовується стійкий механізм реагування на певні чинники, який дозволяє опрацьовувати нову інформацію майже автоматично. Раніше набуті зв'язки, після надходження відповідного запиту чи умов задачі, активуються миттєво, і людина виконує необхідні логічні операції без спеціального осмислення кожної дії. Подібні механізми, що ґрунтуються на асоціативному узагальненні, утворюють функціональну основу для вирішення окремих категорій завдань. Сукупність таких закріплених мисленнєвих дій утворює операційні структури, які забезпечують цілісність та ефективність процесу мислення [21].

На думку Рибалко О., основу операційного типу мислення складають специфічні уміння та навички, що забезпечують ефективне розв'язання навчальних і практичних завдань. Серед них: здатність планувати чітку послідовність дій, необхідних для досягнення мети; уміння аналізувати об'єкти шляхом виявлення їхніх характеристик; організація процесу пошуку актуальних відомостей; дотримання визначених правил при роботі з об'єктами; а також вільне володіння клавіатурним введенням на цифрових пристроях [22].

Схожу точку зору висуває й Воробей О., підкреслюючи важливість розвитку здатності до алгоритмічного мислення – оформлення складної діяльності у вигляді чітко структурованих кроків. Такий підхід дозволяє здобувачам освіти не лише ефективніше виконувати завдання, але й закладати основи системного та логічного підходу до навчання [23].

Мисленнєві операції – це сформовані та автоматизовані способи пізнання, що застосовуються в процесі розумової діяльності. У педагогіці останнім часом помітна тенденція до практичного втілення цієї ідеї через концепції алгоритмізації освітнього процесу [24], [25]. У галузі психології та педагогіки термін «алгоритмічний підхід» передбачає створення та опанування учнями чітких схем розв'язання типових навчальних завдань, а також структурування самого процесу викладання, коли педагог дотримується певного алгоритму дій для досягнення навчальної мети [26].

У сучасній методиці недостатньо опрацьоване питання формування понять, які лежать в основі розпізнавання об'єктів і розв'язання задач. Недостатньо також описано, як на основі цих понять створювати оптимальні алгоритми та навчати учнів ефективно їх застосовувати. Науковці відзначають, що існують різні шляхи навчання алгоритмізації. Один із них – надання готових алгоритмів учням, що може бути доречним у певних випадках. Проте навіть у таких умовах важливо, щоб здобувачі освіти розуміли зміст матеріалу, з яким працюють, знали властивості явищ і могли усвідомлено застосовувати алгоритм.

З педагогічної точки зору значно ефективніше, коли учень самостійно відкриває або за допомогою вчителя створює алгоритм. Такий підхід сприяє глибшому розумінню і креативності не лише в математиці, а й в інших навчальних дисциплінах. Варто розпочинати не з заучування послідовності дій, а з вивчення суті явища, його особливостей і взаємозв'язків із іншими явищами.

Одне з актуальних питань методики – чи не стане навчання алгоритмам причиною шаблонності мислення і не стримуватиме творчість? Багато праць дослідників зважають на цю небезпеку, вважаючи, що алгоритмічне мислення не повинно обмежуватись готовими схемами. Навпаки – воно має стимулювати побудову власних алгоритмів: підтримувати логіку і послідовність дій, а ще й розвивати творчі здібності.

Правильно організоване навчання алгоритмам передбачає розвиток умінь самостійно їх створювати, а не лише відтворювати готові. Алгоритм виступає

водночас як структурна схема дій і як засіб для розвитку креативності. Опанування алгоритмічних дій дозволяє уникати помилок у процесі розв'язання задач. Якщо учень оволодіє цими діями, він зможе ефективно і логічно мислити. Без цього ж вміння він не зможе досягати одразу результату, а тільки шляхом спроб і помилок [27].

На схоже питання відповідає одна з найвідоміших психологічних теорій інтелектуального становлення – концепція поетапного розвитку мисленнєвих дій, заснована на напрацюваннях Гальперіна П. та Талізінної Н. Згідно з їхніми твердженнями, формування вміння вирішувати завдання розпочинається з того, що дитина практично опановує визначення понять, речей або явищ. У процесі навчання властивості та зв'язки, які це визначення містить, починають відігравати роль орієнтирів для правильного виконання дій.

Формування розумових операцій, на думку Гальперіна П., проходить кілька послідовних стадій: спочатку створюється загальна орієнтаційна схема дії, далі вона відображається у матеріалізованій формі. Потім ця дія переходить у зовнішнє мовлення, згодом – у внутрішню мовленнєву форму. Завершується процес повним автоматизмом – дія здійснюється подумки, без зовнішніх впливів.

Згідно з теорією Махмутова М., який досліджував зв'язок між методами наукового пізнання та навчання, пізнавальний процес у проблемному навчанні проходить через наступні етапи: виникнення проблемної ситуації, її аналіз, пошук шляхів розв'язання, включаючи як вже відомі, так і нові способи – через висунення гіпотез, інтуїтивні здогадки та логічні підтвердження. Проблеми можуть бути різними за характером невідомого, серед них – алгоритмічні, що передбачають застосування вже засвоєних рішень. Сам алгоритмічний процес полягає у поетапній обробці та трансформації інформації для досягнення поставленої мети. Повна побудова алгоритму включає формулювання задачі, створення моделі, розробку й реалізацію алгоритму, його аналіз, відпрацювання, обробку результатів та відповідне документування [21].

Основним напрямком формування алгоритмічного мислення в закладі освіти є уроки інформатики, зокрема теми, пов'язані зі створенням алгоритмів і програмуванням. Водночас варто підкреслити, що засвоєння алгоритмічних навичок відбувається не лише на уроках інформатики, а й у процесі вивчення інших предметів – української мови, математики, технологій, а також інтегрованого курсу STEM чи інших дисциплін.

Під час впровадження інформатики в початкових класах основна мета була визначена як розвиток алгоритмічного мислення та навичок створення алгоритмів у здобувачів освіти. Зокрема, у 2013 році курс «Сходи до інформатики» містив розділ «Алгоритми і виконавці», що знайомив молодших школярів із поняттям алгоритму, командами виконавця та прикладами застосування алгоритмів у повсякденному житті. Крім цього, курс передбачав навчання основам програмування в середовищі Scratch [28].

Відповідно до положень реформи НУШ одним із завдань освітньої сфери інформатики є стимулювання формування алгоритмічного мислення у учнів початкової та середньої школи. Здобувачі освіти мають навчитися чітко виконувати задані алгоритми для реалізації різних завдань – наприклад, виготовлення виробів, малювання фігур на папері або під час роботи у цифровому середовищі. Вони повинні засвоїти значення терміну «алгоритм», а також вміти розрізняти типи алгоритмічних структур: послідовність дій (лінійна), розгалуження та цикли. Окрім того, учні мають розуміти роль виконавця алгоритму, його набір команд та основні властивості алгоритмів. Також школярі повинні мати навички створення і запису алгоритмів, як у текстовому, так і у графічному вигляді, зокрема для побутових операцій або для виконання дій в програмному середовищі Scratch [29].

Для ефективного розвитку алгоритмічного мислення в учнів початкових класів доцільно використовувати інтерактивні онлайн-ресурси. Наприклад, середовище Blockly, яке пропонує низку ігор, що передбачають поетапне знайомство з базовими командами, побудову алгоритмів та досягнення конкретних цілей через вправи на логіку, послідовність і повторення.

Іншим онлайн-ресурсом є Pilas Bloques, який містить набір інтерактивних завдань для дітей, знайомих з основами програмування. У ньому вправи поділені за рівнями складності й орієнтовані на поступове засвоєння логіки дій та формування стійких навичок побудови алгоритмів.

З метою удосконалення відповідних умінь у мережі Інтернет розроблено завдання для учнів 2 – 4 класів. Серед них – розуміння поняття алгоритму й виконавця, виявлення помилок у лінійних алгоритмах, розгалуженнях, а також вправи на створення власних алгоритмічних рішень. До таких завдань віднесено графічні диктанти, доповнення відсутніх етапів у послідовностях дій, визначення правильного результату виконання алгоритму тощо [20].

Із моменту впровадження предмета «Інформатика» до початкової школи одним із провідних напрямів навчання стало формування в учнів базових понять, пов'язаних із алгоритмами, розвиток вмінь будувати й аналізувати алгоритмічні послідовності. У програмі НУШ чітко окреслено основні етапи формування алгоритмічного мислення, що включають створення команд, побудову алгоритмів за прикладами, виявлення неточностей у заданих послідовностях дій, ознайомлення з основними типами структурування алгоритмів – такими як лінійна послідовність, повторення, розгалуження. Окрім того, передбачено засвоєння принципів виконання алгоритмів у спеціалізованих середовищах та розпізнавання істинності або хибності тверджень.

Під час занять з інформатики в початковій школі учні можуть використовувати різні способи подання алгоритмів. Серед них виділяють графічний варіант, опис алгоритму словами, а також програмний запис. Програмний варіант вимагає глибшого розуміння структури алгоритму, тож його вивчення розпочинається із третього року навчання. У другому класі ще не вводиться саме поняття «алгоритм», однак завдяки послідовним діям і опису кроків учні інтуїтивно знайомляться з принципами, які згодом допоможуть у повноцінному сприйнятті алгоритмічного мислення.

Починаючи з третього класу, поняття «алгоритм» і «команда» набувають чіткого змісту. Здобувачам освіти пропонуються приклади команд у формі спонукальних речень, після чого вчитель визначає, хто виконує зазначені дії – так формується розуміння ролі виконавця. У цей період учні засвоюють зміст понять, пов'язаних із алгоритмами, знайомляться з базовими елементами логіки. Здобувачі освіти добре розуміють, що виконавцями команд можуть бути як живі істоти, так і неживі об'єкти – побутові пристрої, фантастичні персонажі, рослини, інструменти чи навіть тварини [30].

Методика, яку використовують у межах інформатичної галузі початкової освіти, орієнтована на впровадження алгоритмічного підходу не лише у спеціальній темі, присвяченій алгоритмам, але й у багатьох інших розділах курсу. Наприклад, під час опанування навичок створення презентацій, учням пропонують виконати завдання з алгоритмічним характером: як змінити порядок слайдів, видалити слайд чи налаштувати його властивості. У такий спосіб поняття алгоритму стає звичним і закріплюється у активному словниковому запасі молодших школярів.

У третьому класі школярі знайомляться зі структурою різних алгоритмічних послідовностей за наступним принципом: спершу вони аналізують приклади алгоритмів, які демонструють певну логіку побудови, далі визначають ключові слова і графічні позначення для запису цих алгоритмів у вигляді блок-схем. Наступним кроком є самостійне складання алгоритмів, їх практична реалізація, а також виявлення та виправлення логічних помилок [32].

Продовження роботи з алгоритмами відбувається у четвертому класі, де значна увага приділяється побудові блок-схем і створенню власних програм для виконавців. У цьому класі вводяться поняття розгалужених алгоритмів і алгоритмів із повторенням.

Поступово учні знайомляться з термінами як логічними одиницями – вони дізнаються про істинність і хибність тверджень. Під час вивчення алгоритмів з розгалуженням школярам пропонують розглянути життєві ситуації як основу для побудови умовних дій. Наприклад, дитині можуть

запропонувати змодельовати ситуацію з побуту, в якій потрібно прийняти рішення залежно від поставленої умови. Такі завдання розвивають логічне і послідовне мислення, допомагають краще зрозуміти, як функціонує алгоритм у реальному житті [33].

На етапі базової середньої освіти, починаючи з 5 класу, вивчення алгоритмів і процес алгоритмізації стають поступово складнішими, набуваючи більшої глибини і практичного спрямування. У п'ятому класі школярі починають знайомство з основними алгоритмічними структурами: слідуванням, розгалуженням і циклами. У навчальному процесі активно використовується середовище Scratch, яке дозволяє учням у візуальній формі оновлювати комбінування простих команд, створювати перші алгоритми з кількох етапів, тестувати їхню логіку й бачити результат у реальному часі.

У шостому класі до змісту додаються поняття умовного виконання дій та розширені алгоритмічні структури, що включають цикли з передумовою та післяумовою. Значну увагу приділяють практичним завданням на реалізацію вибору дій залежно від заданої умови. Паралельно відбувається знайомство з текстовими мовами програмування, серед яких відводиться значна увага Python. Це зручна та зрозуміла мова, яка добре підходить для початкового засвоєння принципів програмування. Завдяки цьому учні починають усвідомлювати логіку роботи програмного коду та закріплюють поняття алгоритмічного мислення на практиці.

У 7 – 8 класах учні переходять до глибшого аналізу алгоритмів. Вивчаються складніші конструкції, такі як вкладені цикли, логічні оператори, рекурсія, функції та параметри. Алгоритмізація на цьому етапі тісно поєднується з міжпредметними зв'язками, зокрема з предметами математики, фізики та технологій. Учні реалізують проєктну діяльність, створюють моделі, автоматизовані системи, симуляції реальних процесів. Окремі завдання включають реалізацію алгоритмів сортування, пошуку елементів, а також формують уявлення про ефективність та складність алгоритмів. Такий підхід сприяє формуванню здатності до самостійної розробки невеликих програмних

продуктів, де учень виступає повноцінним автором – від ідеї до реалізації [34, 35].

Отже, розвиток алгоритмічного мислення під час вивчення інформатики – це важлива частина формування розумових навичок учнів. Це мислення допомагає формувати здатність до логічного аналізу, прогнозування та структурованого вирішення навчальних задач. В освітньому процесі алгоритмічний підхід забезпечує систематизацію знань, розвиток самостійності та критичного мислення. У початкових класах школярі поступово переходять від наочно-образного до абстрактно-логічного мислення, що створює передумови для ефективного опанування процесів алгоритмізації. Важливим компонентом цього розвитку є послідовне закріплення розумових стратегій, що дозволяє учням швидко адаптуватися до нових завдань і автоматизувати процес обробки інформації.

Психолого-педагогічні умови, що сприяють формуванню алгоритмічного мислення, включають створення навчальних ситуацій, які потребують чітких послідовностей дій, аналізу та порівняння даних. Значну роль відіграє методика викладання даного предмету для учнів. Вона має на меті – спрямування здобувачів освіти не лише на використання готових алгоритмів під час вирішення проблемних ситуацій у школі, а й на їх самостійне створення. Це сприяє розкриттю креативності та усвідомленому застосуванню алгоритмічного підходу у навчальній діяльності. З педагогічної точки зору важливо не обмежувати учнів механічними процесами із запам'ятовування алгоритмів, а навпаки, мотивувати їх до самостійного пошуку оптимальних шляхів розв'язання задач, що є основою для формування стійкої здатності мислити та розвитку аналітичних навичок.

Саме тому алгоритмічне мислення виступає не лише важливим елементом навчання інформатики, а й загальним інструментом для розвитку інтелектуальних здібностей учнів. Його формування позитивно впливає на засвоєння інших навчальних дисциплін, сприяє розвитку системного підходу до аналізу явищ, а також забезпечує здатність до осмисленого застосування знань

у практичних ситуаціях. Це особливо важливо для підготовки учнів до подальшого їх навчання, професійної діяльності та інтеграції в інформаційне суспільство.

1.3. Методичні підходи до розвитку алгоритмічного мислення в шкільному курсі інформатики

Сучасна освіта, в основі якої лежить компетентнісний підхід навчання, має на меті не лише теоретичне засвоєння знань, а й формування в учнів певних умінь і навичок. Саме їх вони зможуть успішно використовувати в реальному житті. Головна складова цього підходу – це інформаційна компетентність. Її розглядають як інтегровану якість особистості, що охоплює здатність цілеспрямовано добирати, аналізувати, осмислювати, трансформувати та продукувати інформацію. Суть інформаційної компетентності полягає ще й у вмінні приймати оптимальні рішення в залежності від різноманітних життєвих ситуацій [35].

У рамках реформи освіти НУШ, яка відбулася нещодавно, викладання інформатики у закладах освіти ґрунтується на компетентнісному підході. Він передбачає розвиток ключових умінь учнів для комфортного життя в цифровому суспільстві. Особлива увага приділяється формуванню цифрової грамотності, до складу якої входить алгоритмічне мислення. На уроках інформатики здобувачі освіти не лише отримують теоретичні знання, а й навчаються застосовувати їх на практиці. Зазвичай вони працюють з візуальним (Scratch, Blockly) і текстовим програмуванням (Python), створюють інтерактивні проєкти, аналізують задачі та формують алгоритми розв'язання певних задач. Завдяки цьому учні навчаються мислити послідовно, приймати зважені рішення, оцінювати результати своєї роботи, шукати помилки та використовувати технології як інструмент для досягнення власних цілей [36].

Компетентнісний підхід реалізується через інтеграцію інформаційних технологій у навчальний процес. Бо учень не просто вивчає інформатику, а

застосовує її можливості для розв'язання прикладних завдань. Саме алгоритмічне мислення виступає критичною складовою цифрової компетентності. Воно дозволяє скласти послідовність дій, прогнозувати результати й аналізувати свої рішення [37].

Викладання інформатики в освітньому закладі має на меті всебічний розвиток учня як активного учасника цифрового суспільства. Зміст цього предмета спрямований на формування інформаційної та алгоритмічної культури, що є основою сучасної освіченості. Учні поступово, крок за кроком, опановують поняття інформації, алгоритму, моделі, навчаються розрізняти їхні властивості та застосовувати під час виконання практичних завдань. Основна увага зосереджена на розвитку алгоритмічного мислення. Саме такий тип мислення необхідний як у повсякденному житті, так і в професійній діяльності. Засвоюючи поняття алгоритмічних конструкцій, логічних операцій та основ мов програмування, здобувачі освіти навчаються складати власні алгоритми для різних виконавців, а ще й намагаються реалізовувати їх у програмному середовищі. Водночас у них формується вміння структурувати дані, обирати найефективніші способи їх представлення (схеми, таблиці, графіки) та використовувати дані цифрові інструменти для обробки будь-якої інформації. Особливу увагу приділено цифровій безпеці, етиці спілкування в Інтернеті, а також усвідомленому й правовому використанню інформаційних ресурсів [38].

Формування алгоритмічного мислення забезпечується активним залученням учня до пізнавального процесу. Саме тому важливою складовою сучасного уроку інформатики є ще й діяльнісний підхід. Він забезпечує засвоєння знань через практичну активність, дослідження та створення цифрових продуктів.

Традиційні підходи до організації освітнього процесу втрачають сенс, бо недостатньо сприяють розвитку колективної форми навчання. Вони в більшій мірі не передбачають взаємну відповідальність учнів та не націлені на спільному досягненні мети. Така форма роботи не формує справжньої взаємодії

в учнівському середовищі та не створює умов, за яких контроль за виконанням завдань здійснюється самими учасниками навчання.

Одним із ефективних засобів подолання цієї проблеми прийнято вважати – застосування діяльнісного підходу. Зокрема, активні методи навчання – такі як робота над проєктами, вирішення проблемних ситуацій, рольові моделювання, мозкові штурми – дають змогу значно підвищити мотивацію до навчання, активізувати процеси мислення, розвивати самостійність у прийнятті рішень і вміння взаємодіяти в команді.

Під час вивчення інформатики та завдяки застосуванню цих методів формуються навички аналізу, пошуку нестандартних рішень, критичного мислення та сприйняття нової інформації. Наприклад, використання проблемного навчання дає змогу організувати пізнавальну діяльність навколо штучно створених ситуацій. Вони викликають в здобувачів освіти інтелектуальне напруження та бажання знайти нові підходи до розв'язання поставленого завдання. Перед учнем постає навчальна проблема, яку не можна розв'язати відомими способами. Саме це стимулює пошук нових рішень та сприяє кращому засвоєнню знань. Це і є головною метою діяльнісного підходу під час вивчення предмету інформатики [39].

Під час використання цього підходу варто визначити основну одиницю – не інформацію, а активну дію учня, яка спрямована на розв'язання реальних задач. У ході розгляду основних її характеристик учень виступає як активний дослідник: аналізує, моделює, приймає рішення та оцінює результати своєї роботи. У інформатиці це означає, що алгоритмічне мислення формується не в теорії, а через практичну діяльність з програмування чи моделювання проєктів.

Алгоритмічне мислення ґрунтується на здатності людини логічно аналізувати ситуацію, передбачати наслідки дій і створювати точні покрокові інструкції для досягнення певної мети. Люди з розвиненим алгоритмічним мисленням вміють розкладати складні задачі на прості частини, визначати ефективну послідовність дій і прогнозувати можливі труднощі на кожному етапі. Ця здатність особливо цінна в умовах змін, адже дозволяє швидко

орієнтуватися в нових ситуаціях і знаходити нестандартні шляхи до розв'язання проблем. Серед основних елементів алгоритмічного мислення варто виділити наступні.

1. Аналіз завдання – це уміння глибоко осмислювати проблему, ділити її на складові частини та формулювати чітке бачення мети.
2. Створення алгоритму – це побудова логічної та впорядкованої послідовності дій чи кроків для досягнення мети.
3. Оцінка та оптимізація рішень – це перевірка ефективності створеного алгоритму, подальше його вдосконалення та коригування на основі виявлених недоліків або за участю нових умов [40].

Алгоритмізація у даному контексті розкривається через виконання практичних завдань. Основою уроків стають практичні вправи: створення алгоритмів у різних застосунках, створення та розв'язок ігрових завдань (лабораторії, квести, пізнавальні блоки).

Наприклад, одним із ефективних інструментів у даному випадку є середовище Scratch, яке допомагає учням у простій та зрозумілій формі пізнати основи алгоритмізації та програмування. Основи роботи у середовищі Scratch використали у своїх працях Лодатко Є., Яріш Г. та Мочула О. Ці автори підручника з інформатики для 4 класу, знайомлять школярів із створенням проєктів, які не вимагають написання коду на мовах програмування, а базуються на графічних інструментах для відображення даних і структур керування [41].

Scratch – це навчальна платформа, розроблена Массачусетським технологічним інститутом. Вона допомагає учням здобути базові навички програмування, створюючи інтерактивні історії, ігри чи анімації. Ця система вкрай корисна в навчальному процесі, оскільки стимулює розвиток алгоритмічного й логічного мислення серед учнів.

Дана платформа є засобом візуального програмування, що підтримує принципи об'єктно-орієнтованого підходу. Вона дає змогу створювати анімовані мультимедійні продукти у формі історій, ігор, моделей чи інших

проектів. Інтерфейс Scratch має зрозумілий та простий дизайн та підтримує багато мов, що надає можливості до програмування рідною мовою користувача.

Як зауважують дослідники Дудка О., Магомета Н. та Власій О., середовище Scratch відкриває можливості для створення інтерактивних матеріалів. Його можна застосовувати для демонстрації, наприклад, природних явищ, історичних подій, а також для виготовлення вітальних листівок чи різного роду презентацій. Це значно розширює можливості навчального процесу, сприяє міжпредметній інтеграції та впровадженню компетентнісного підходу.

Користуючись Scratch, учні розвивають алгоритмічне та логічне мислення. Це пов'язано з тим, що вони змушені чітко планувати свої дії перед тим, як перетворити їх у робочий код. Школярі навчаються складати послідовність кроків для розв'язання задач, розділяючи складні проблеми на менші, керовані частинки. Останні – носять назву процесу, відомого як декомпозиція. Також платформа підтримує використання умовних операторів і циклів, що допомагає школярам застосовувати логіку для контролю виконання програми.

Декомпозиція допомагає учням розбивати складні завдання на простіші, більш контрольовані етапи. Наприклад, створюючи гру в Scratch, спочатку можна оформити фон, потім додати персонажів, а далі – прописати правила гри та встановити різні рівні складності. Такий поетапний підхід дає змогу учням поступово та за складеним планом працювати над проектом, поступово опрацьовуючи кожен його елемент.

У Scratch умовні оператори надають можливість програмувати поведінку персонажів відповідно до певних умов. Наприклад, якщо герой доторкається до певного об'єкта, він може отримати бонусні бали або, навпаки, втратити життя. Це сприяє розвитку логічного мислення, бо учні навчаються обирати правильні умови та передбачати наслідки кожної дії [42].

Алгоритмічне мислення – основа при вивченні мови програмування. Ця платформа Scratch чудово підходить для його розвитку серед учнів. Вона допомагає формувати це мислення через різні активності та функції.

Середовище Scratch використовує візуальні блоки, які представлені кодом. Це значно полегшує процес засвоєння принципів роботи програм. Учні можуть просто перетягувати й об'єднувати ці блоки, а це сприяє кращому розумінню логічних зв'язків між елементами коду. Наприклад, можна створити простий скрипт, який рухає уперед при натисканні на праву стрілку, що наочно ілюструє поняття подій і реакцій у програмуванні [43].

Алгоритмічне мислення створює здатність створювати послідовні дії для вирішення поставлених завдань. У середовищі Scratch учні мають змогу застосовувати алгоритми для виконання складних завдань, наприклад, переміщення об'єкта через лабіринт. Для цього потрібно правильно визначити порядок блоків, що відповідають за виявлення перешкод, рух та взаємодію з іншими об'єктами, що розвиває навички логічного аналізу та планування.

Цикли та умовні конструкції – це невід'ємні складові частини алгоритмічного мислення. У Scratch користувачі можуть застосовувати блоки «повторити», що змушують певні дії виконуватись багаторазово. Наприклад, щоб створити анімацію руху персонажа, використовується цикл для багаторазового повторення однієї дії. Умовні блоки типу «якщо» дозволяють програмі реагувати на конкретні ситуації. Наприклад, змінювати напрямок у разі зіткнення зі стіною.

Крім того, середовище Scratch є платформою для обміну розробками. Це сприяє колективному навчанню та розвитку вміння працювати в групі. Спільне обговорення алгоритмів поглиблює розуміння логіки програмування і допомагає покращити навички вирішення складних задач.

Проектна діяльність у Scratch позитивно впливає на розвиток як логічного, так і критичного мислення. Учні аналізують недоліки власних прийнятих рішень. Потім вони виправляють помилки та вдосконалюють код.

Процес налагодження сприяє формуванню витримки, наполегливості й глибшому розумінню внутрішньої логіки програм.

Середовище Scratch має власну онлайн-спільноту, де учні публікують проекти, отримують коментарі й надихаються ідеями інших. Це формує не лише технічні, а й соціальні їх вміння. Спільні розробки сприяють розвитку співпраці: діти об'єднують зусилля, обговорюють ідеї, діляться функціями та спільно шукають оптимальні рішення. Отриманий досвід формує у школярів навички командної роботи, яка необхідна в сучасному світі. Отримуючи позитивні відгуки, діти прагнуть створювати щось нове, складніше, цікавіше, а ще це мотивує їх до подальшого навчання.

Ефективність використання середовища Scratch залежить від обраних педагогом педагогічних стратегій. Учителю важливо не просто пояснити, як працюють блоки, а створити умови для активного мислення учнів через проєктну діяльність. Йому варто підбирати проєкти на теми, які цікаві самим школярам. Бо вони мотивують більше, ніж механічне повторення певних дій. Успішне навчання також передбачає врахування індивідуального темпу та рівня знань, тому важливо застосовувати диференційований підхід навчання.

Окреме місце відведене співпраці учителя та учнів: у парах чи групах вони навчаються пояснювати свої ідеї, аргументувати логіку рішень, слухати одне одного та спільно долати труднощі. Увагу слід приділяти не лише результату, але й процесу. Варто відводити час на обговорення етапів розробки алгоритму, можливих перешкод чи труднощів, які можуть виникнути. А ще варто розглянути способи їх подолання. Такий підхід формує у школярів усвідомлене ставлення до навчання та забезпечує глибоке опанування принципами алгоритмічного мислення [44].

У сучасній освіті важливим завданням є формування універсальних навичок, зокрема алгоритмічного мислення. Воно лежить в основі ефективного опрацювання інформації, аналізу й вирішення практичних задач. Цей вид мислення не обмежується лише інформатикою – його потенціал успішно реалізується на уроках математики, природознавства, трудового навчання,

STEM та інших дисциплін. Завдяки міжпредметній інтеграції алгоритмічний підхід – це дієвий засіб навчання. Так учні навчаються будувати послідовні інструкції, шукати закономірності, планувати дії та передбачати наслідки. Такий підхід значно підвищує якість засвоєння знань і розвиває ключові компетентності.

Наприклад, у межах сучасного підходу до інтегрованої освіти дослідниці Jessica F. Cantlon, Katherine T. Becker та Caroline M. DeLong розглянули, як взаємодія інформатики з природничими дисциплінами сприяє розвитку алгоритмічного мислення у молодших школярів. Їхнє дослідження було короткотривале. Під час його проведення здобувачі освіти створювали цифрові ігри для аналізу поведінки приматів у зоопарку.

Учасники проєкту мали змогу застосовувати програму Scratch, щоб розробити інтерфейс гри. А вона, у свою чергу, заснована на справжніх наукових спостереженнях. Це дозволило їм поєднати програмування із вивченням тваринної поведінки – галуззю, що належить до природничих наук. Учні не лише використовували логічні конструкції (умови, цикли, послідовності), а й проєктували справжні стратегії спостереження, висували гіпотези та аналізували отримані результати.

Таким чином, у процесі виконання завдань з алгоритмізації школярі розвивали базові компетентності з інформатики. До них належали: послідовне планування дій, передбачення наслідків алгоритмів, корегування дій на основі отриманого досвіду. Це відбувалося в ході інтегрованого дослідження, що підсилювало мотивацію та критичне мислення. Спрямованість активностей на вивчення явищ природи дозволила забезпечити міжпредметну взаємодію в межах STEM-моделі.

Експериментальні результати досліджень вказують на ефективність коротких інтеграційних STEM-проєктів у формуванні ключових когнітивних навичок, зокрема здатності декомпонувати складні задачі, перевіряти їхню структуру та оптимізувати алгоритми дій у відповідь на зміну зовнішніх умов [45].

Отже, розвиток алгоритмічного мислення під час вивчення предмету інформатики є невід'ємною складовою формування цифрової та інформаційної компетентностей учнів. Ефективне впровадження методичних підходів до формування цього виду мислення ґрунтується в поєднанні компетентнісного та діяльнісного підходів. Це передбачає активну участь учнів у створенні цифрових продуктів, вирішенні практичних задач, участі у проектах та моделюванні життєвих ситуацій.

Одним із ключових методичних засобів є використання середовища візуального програмування Scratch. Воно забезпечує доступність, наочність і мотивацію до навчання. Через проєктну діяльність, створення ігор, анімацій, симуляцій, учні навчаються декомпозиції завдань, формуванню послідовних алгоритмів, використанню умов, циклів, змінних, що формує в них цілісне уявлення про структуру програмного мислення.

Особливу ефективність демонструють інтеграційні підходи. Завдяки їм алгоритмічне мислення розвивається не лише в межах предмету інформатики, а й у поєднанні з природничими науками, математикою, технологіями та STEM-напрямом. Такі міжпредметні зв'язки стимулюють учнів до використання алгоритмічного підходу у вирішенні реальних завдань. Завдяки йому спостерігається глибше розуміння навчального матеріалу. Тому методика навчання інформатики має бути динамічною, гнучкою та орієнтованою на активну пізнавальну діяльність. Це дозволить формувати в учнів універсальні навички XXI століття. А розвиток алгоритмічного мислення має бути системним, поетапним і тісно пов'язаним з реальним досвідом учня.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ОСВІТНЬОГО СЕРЕДОВИЩА ДЛЯ РОЗВИТКУ АЛГОРИТМІЧНОГО МИСЛЕННЯ: ДОСВІД І ПЕРСПЕКТИВИ

2.1 Умови організації навчальної діяльності здобувачів освіти для розвитку алгоритмічного мислення

Молодший шкільний вік – це період, коли закладаються основи мислення, здатність до послідовного аналізу та вміння планувати свої дії. Саме в цей час найкраще розвиваються рефлексія, внутрішній план дій та навички впорядкування власних кроків. Це створює основу для формування алгоритмічного стилю мислення. Варто зауважити, що такі якості не з'являються самостійно, їх становлення потребує спеціально організованої роботи як педагога так і систематичної роботи здобувачів освіти. У такому випадку необхідно ретельно продумувати структуру навчального процесу у кожному класі ЗЗСО. Лише цілеспрямоване поєднання завдань, вправ і ігрових методик забезпечує стійкий розвиток алгоритмічних умінь [28].

Подальший розгляд дослідження варто розглядати у контексті аналізу умов, що визначають результативність організації навчальної діяльності здобувачів освіти. Саме ці умови складають змістове наповнення, методичні прийоми, способи контролю та стимулювання пізнавальної активності учнів. Їхня взаємодія найкраще відображається у структурованій формі. Саме тому

нами створено таблицю, яка узагальнює ключові чинники розвитку алгоритмічного мислення школярів (Додаток А).

З неї можна зробити висновок, що розвиток алгоритмічного мислення потребує створення умов, що поєднують когнітивну стимуляцію, доступ до цифрових ресурсів та педагогічну підтримку. Особливо важливо забезпечити середовище, де учні можуть експериментувати, помилятися і шукати рішення.

Мотивацію більшість дослідників розглядають як психолого-педагогічну умову. Це пов'язано з тим, що вона зумовлена її вирішальною роллю у формуванні стійкого інтересу до навчання та розвитку алгоритмічного мислення. Саме мотивація визначає готовність учнів активно діяти, приймати інтелектуальні виклики й послідовно досягати поставлених цілей.

Наприклад, щоб якісно закріпити здобуті знання, педагогу варто наводити реальні приклади з життя, включати в заняття ігрові елементи, пропонувати спільне обговорення результатів. Такі форми роботи сприяють активній праці здобувачів освіти у вільній формі, без напруження. Такий підхід створює атмосферу успіху, у якій діти відчувають власну значущість і прагнуть розвивати навички планування та логічного мислення.

Створення практичних завдань, що відображають життєві потреби учасників навчального процесу, забезпечує сталість інтересу та підсилює внутрішню мотивацію. Учні із задоволенням виконують кроки алгоритму, аналізують дії та шукають ефективні рішення. Це важливо, бо вони бачать їх користь у реальному житті. Це дозволяє не лише формувати алгоритмічний стиль мислення, а й розвивати наполегливість, самостійність і готовність до подальшого навчання.

Індивідуальний підхід і врахування вікової готовності є ключовими чинниками розвитку алгоритмічного мислення школярів. На цьому етапі важливо звертати увагу на рівень сформованості пам'яті, концентрацію уваги учнів, здатність до логічних операцій та здатності до самоконтролю, оскільки саме вони визначають готовність дитини до побудови внутрішнього плану дій. Ефективна організація навчання потребує гнучкої адаптації змісту, темпу та

методів, що дозволяє підтримувати інтерес, розкривати потенціал кожного учня і поступово формувати навички послідовного мислення.

Соціальна взаємодія – це важлива умова формування алгоритмічного мислення, бо спільна діяльність створює можливість для обміну ідеями, формує умови для порівняння підходів і колективного пошуку рішень. Організація роботи в парах чи малих групах стимулює учнів до обґрунтування власних дій, планування послідовності кроків і критичного аналізу результатів. Це поглиблює розуміння навчального матеріалу та розвиває здатність до рефлексії [45].

Ігрові та проблемні методики створюють природне середовище для розвитку алгоритмічного мислення, оскільки поєднують цікавість із необхідністю логічних дій. Розв'язання нестандартних завдань, участь у навчальних іграх чи моделювання ситуацій спонукають дітей аналізувати умови, виділяти суттєві етапи та вибудовувати послідовність кроків для досягнення поставленої мети. Такий підхід підвищує мотивацію, формує навички планування та сприяє глибшому засвоєнню знань через активну практику й самостійне прийняття рішень.

Щоб отримати довготривалий ефект, важливо цілеспрямовано поєднувати розвиток алгоритмічного мислення з усіма етапами навчального процесу. Найбільш результативним виявляється інтегрування такого підходу у викладання наступних предметів: математики, інформатики, STEM та технологій. Ці дисципліни гарно поєднують логічні операції, мають чітку послідовність дій та здійснюють практичну перевірку здобутих знань.

Методична інтеграція алгоритмічного підходу формує не тільки цифрову грамотність, а й сприяє розвитку аналітичного мислення, планування та вміння узагальнювати. Завдяки цьому учні навчаються структурувати завдання, передбачати результати та застосовувати здобуті навички у різних життєвих і навчальних ситуаціях. Саме такий підхід робить освіту більш сучасною та практико орієнтованою.

У статті науковиці О. С. Лісневської [46] досліджуються різні види роботи на уроках математики та технологій. Вони сприяють формуванню просторової уяви та творчого мислення молодших школярів. Запропоновані вправи допомагають розвивати математичні здібності, підвищують інтерес учнів до навчання, спонукають їх до процесів порівняння, аналізу, а ще навчають співставляти відомості й розвивають вміння виділяти найважливіше у завданнях.

Особлива увага приділяється інтеграції уроків математики та технологій для розвитку мислення і просторової уяви. Виконання завдань з серії «Уяви, що вийде» дозволяє учням подумки передбачати результати і аналізувати наслідки прийнятих рішень. На заняттях пропонується виконати ряд вправ. Наприклад, поступове «розгортання» серветок, щоб діти бачили, як один розріз створює кілька відображень фігур. Далі учні самостійно створюють подібні завдання, виконують їх у парах або групах, спостерігають один за одним і роблять висновки про симетрію та повторюваність дій.

Нею представлена робота з конструктором «Танграм», де школярі створюють геометричні фігури, поступово ускладнюють завдання і переходять до самостійного створення форм. Крім того, проводиться робота з сірниковими коробками: діти рахують грані, порівнюють розміри, обклеюють їх кольоровим папером і об'єднують у макети предметів довкілля. Такий підхід допомагає розвивати просторове сприймання, логіку, навички планування та аналізу результатів. Кожен етап передбачає активну участь учнів, а ще розвиває вміння шукати помилки, узагальнювати досвід та поступово формувати творче і алгоритмічне мислення.

Алгоритмічне мислення є основою успішного засвоєння знань у багатьох галузях – від математики та інформатики до природничих дисциплін і навіть повсякденних життєвих ситуацій. Питання його формування постійно розглядається та досліджується науковцями, а також педагогами упродовж усього розвитку шкільного курсу інформатики. Це поняття можна визначити як особливий тип мислення, що дає змогу розбивати складні завдання на прості,

логічно обґрунтовані кроки, які ведуть до очікуваного результату. Воно охоплює здатність скласти послідовність дій за допомогою зрозумілих та покрокових інструкцій.

Одним із найефективніших засобів формування алгоритмічного стилю мислення в школярів є навчання основам програмування. Під час занять учні навчаються точно висловлювати власні думки та ідеї, розбивати завдання на окремі етапи, прогнозувати можливі помилки та перевіряти різні варіанти їх розв'язання. Вчителю варто створювати проблемні ситуації. Вони спонукають дітей до самостійного пошуку рішень та формування нових підходів до розв'язання поставлених задач. Для стійкої підтримки інтересу та закріплення навичок доцільно використовувати різноманітні дидактичні ігри й логічні головоломки. У їх основу закладено принципи алгоритмізації у захопливій і доступній формі. Важливим елементом є й використання наочності: схеми, діаграми, графіки та інші візуальні матеріали. Вони допомагають школярам краще зрозуміти структуру алгоритмів та простежити їхню послідовність виконання [47].

Розвиток алгоритмічного мислення в умовах сучасної школи неможливий без належної матеріально-технічної бази. Адже технічне забезпечення створює простір, де учні отримують не тільки теоретичні знання, а й навчаються практично їх застосовувати. Вони моделюють різні ситуації, проводять експерименти та аналізують результати власної роботи.

Основними компонентами такого забезпечення є:

- комп'ютерна техніка – персональні комп'ютери, ноутбуки чи планшети, які надають учням можливість працювати з навчальними програмами, створювати й відразу перевіряти власні алгоритми.

- стабільний доступ до мережі Інтернет, що відкриває шлях до онлайн-платформ, симуляторів і середовищ для візуального програмування (Scratch, Code.org, Blockly та ін.), де школярі навчаються будувати логічні послідовності та знаходити помилки.

- інтерактивні ресурси й цифрові інструменти, які допомагають організовувати заняття у формі гри, командних проєктів або змагань, підсилюючи інтерес і мотивацію до навчання.

- мультимедійне обладнання – інтерактивні дошки, проектори, аудіо- та відеозасоби, що дають змогу наочно демонструвати схеми алгоритмів, логічні зв'язки та інші абстрактні поняття.

Поєднання цих технічних компонентів забезпечує створення навчального середовища, у якому школярі можуть повноцінно розвивати навички алгоритмічного мислення через практику, експерименти та творчий пошук рішень [28].

Державні стандарти початкової освіти визначають вимоги до рівня загальноосвітньої підготовки учнів. Вони мають узгоджуватися зі структурою та змістом предметних компетентностей і відображатися у результатах кожної освітньої галузі. У переліку ключових завдань навчального курсу «Інформатика» в початковій школі важливе місце займає формування логічного, критичного й алгоритмічного мислення [57]. Розвиток алгоритмічного мислення забезпечує не лише засвоєння самого поняття «алгоритм», а й розуміння його структур та принципів. Уміння виконувати готові алгоритми, аналізувати зміст завдань, переносити здобуті знання та навички у практичну діяльність і повсякденні ситуації – це основа процесу алгоритмізації [48].

Підсумком сформованого алгоритмічного мислення у здобувачів освіти стає здатність створювати й представляти алгоритми у різних формах: графічній (схеми, блок-схеми), мовою програмування або вербально – у вигляді словесного опису дій [49]. При цьому алгоритмічні підходи застосовуються не лише на уроках інформатики. Їх елементи можна простежити на заняттях з інших навчальних предметів. Наприклад, під час уроків української мови. У такому випадку алгоритм проявляється у поетапному вивченні правил написання слів або виконанні звуко-буквеного аналізу. На заняттях курсу «Я досліджую світ» алгоритмічність простежується у проведенні спостережень,

виконанні практичних робіт чи досліджень, які здійснюються відповідно до заздалегідь визначеного плану.

Алгоритм як категорія мислення пронизує всі напрями загальної середньої освіти, незалежно від навчальної галузі чи предмету [28]. У кожній освітній сфері – мовно-літературній, математичній, природничій, технологічній, соціальній, громадянській, мистецькій та фізкультурній – наявні завдання, що передбачають послідовне виконання дій, планування, моделювання та перевірку результатів. Державний стандарт початкової освіти чітко визначає вимоги до результатів навчання, акцентуючи на необхідності розвитку алгоритмічного мислення в усіх без винятку предметах, а не лише в курсі інформатики [28; 50].

Наприклад, у мовно-літературній галузі алгоритмічні навички проявляються під час створення таблиць і схем, переказуванні текстів за складеним планом, аналізом та порівнянням змісту, визначенні головного й другорядного, складанні плану чи моделі на основі тексту, а також у перевірці правопису й пошуку орфографічних помилок [4].

Математична освітня галузь формує вміння розробляти послідовність дій для розв'язання задач будь-якої складності, а ще й прогнозувати результати, застосовувати власний досвід у процесі критичного аналізу отриманих даних.

Природнича галузь сприяє алгоритмічному мисленню через спостереження, планування та проведення дослідів, моделювання явищ, прогнозування наслідків досліджень, систематизацію даних і виявлення взаємозв'язків у навколишньому середовищі [28].

У технологічній сфері алгоритмізація реалізується під час складання планів виготовлення виробів, розрахунку витрат матеріалів та проектування готових продуктів.

Соціальна і здоров'язбережувальна галузь включає послідовність дій у надзвичайних ситуаціях, прогнозування наслідків рішень, організацію навчання та відпочинку.

У громадянській та історичній складових алгоритмічне мислення формується під час критичного аналізу джерел, визначення хронології подій, встановлення причинно-наслідкових зв'язків та аргументуванні власної позиції.

Мистецька галузь розвиває здатність планувати й оцінювати етапи творчої діяльності, вчить визначати послідовність дій для вдосконалення власних робіт.

У фізкультурній галузі алгоритми проявляються під час виконання рухових вправ за визначеними правилами, підборі елементів для гімнастики та дотриманні чіткої послідовності під час ігор і змагань [50].

Таким чином, алгоритмізація в ЗЗСО є наскрізною вимогою сучасної освіти: вона забезпечує не лише оволодіння конкретними предметними знаннями, а й розвиток універсальних умінь планування, аналізу та передбачення, що становлять основу подальшого інтелектуального зростання здобувачів освіти [28; 50].

На уроках інформатики у середніх класах учні поступово переходять від простого виконання готових інструкцій до самостійного створення власних алгоритмів. Для цього доцільно добирати завдання, пов'язані з повсякденними ситуаціями, які добре знайомі дітям. Такий підхід допомагає школярам швидко розвивати всі елементи алгоритмічного мислення та природно сприймати навіть складні ситуації. Важливо, щоб запропоновані вправи відповідали рівню попередніх знань і досвіду новачків, адже надто складні або абстрактні приклади можуть знизити інтерес і впевненість у власних силах.

Найбільш ефективними вважаються завдання, які можна роз'язати кількома способами. Це стимулює пошук власних рішень і розуміння принципів роботи алгоритмів. На початковому етапі для учнів важливіше навчитися бачити логіку та закономірності, ніж запам'ятовувати готові схеми. Основна мета – сформувати усвідомлення, що алгоритм є чіткою послідовністю дій, яка враховує всі можливі ситуації та сценарії, а не лише типові випадки.

Попри це, багато здобувачів освіти сприймають тему алгоритмів як складну та нудну, зосереджуючись на механічному вивченні окремих

послідовних алгоритмів, які вважаються необхідними для практики чи навчальної програми. Щоб змінити таке ставлення, завдання варто подавати практично значущими, різноманітними та цікавими. А ще необхідно підкреслювати їх зв'язок із реальним життям, включаючи можливість творчого підходу до пошуку рішень.

Засвоювати принципи та концепції алгоритмів можна у більш доступній і захопливій формі, якщо перетворити навчання на гру. Суть підходу полягає в тому, щоб залучити якомога більше учнів до практичної діяльності, де вони не лише виконують готові інструкції, а й самостійно створюють та вдосконалюють власні алгоритми. Завдання вчителя – правильно сформулювати проблему, поставити правильні запитання та спонукати школярів до пошуку дієвих рішень, які допомагають виконати поставлену задачу. Педагог також мотивує дітей до аналізу та оптимізації алгоритмів, маючи на меті зробити їх більш ефективними.

Такі уроки можна порівняти з театральною постановкою: учні виступають у ролі виконавців та творців ідей, а вчитель – режисера процесу. Наприклад, завдання на визначення максимального значення серед набору чисел може стати основою для дослідницької гри, де школярі експериментують з різними способами пошуку відповіді, поступово знаходячи нові підходи [53].

Алгоритмічне мислення є однією з ключових компетентностей, яких учні набувають під час вивчення інформатики. Воно охоплює широкий спектр інтелектуальних умінь і розвивається на різних рівнях, формуючи фундамент для подальшого навчання програмування. Однак саме розуміння суті алгоритмів часто викликає труднощі на початку вивчення процесу кодування. Тому важливо довести школярам, що алгоритми – це не сухі інструкції, а потужні інструменти для творчого пошуку. Вони відкривають простір для експериментів і дозволяють досягати значного прогресу.

Під час таких «ігрових» занять діти не лише винаходять алгоритми, а й перевіряють їх на практиці, виявляючи сильні та слабкі сторони прийнятих

рішень. Це пробуджує інтерес до подальших досліджень, стимулює пошук нових ідей та мотивує до поглибленого вивчення інформатики.

Проте робота лише з готовими алгоритмами, що подаються у підручниках, не створює достатніх умов для формування глибоких алгоритмічних умінь. Часто вчитель не створює ситуацій, які б сприяли розвитку цього типу мислення. Натомість сучасні форми та методи викладання інформатики мають не лише навчати виконанню інструкцій і правил, а й формувати здатність логічно міркувати, користуватися відповідною термінологією, будувати та перевіряти висловлювання, робити аргументовані висновки.

Ефективні методики розвитку алгоритмічного мислення учнів 5 – 9 класів передбачають навчити їх розбивати складні завдання на послідовні прості кроки. Вони мають бути взаємопов'язані між собою, а ще й розвивати уміння прогнозувати різні варіанти рішень і передбачати умови, за яких виконується певна послідовність дій. Такий підхід стимулює дітей самостійно висувати припущення, доводити власні твердження, робити обґрунтовані висновки та здобувати нові знання.

Тісний зв'язок алгоритмічного мислення з іншими видами розумової діяльності підкреслює значення інформаційних, рефлексивних і дослідницьких умінь для розвитку загальних закономірностей мислення та інтелектуальної сфери особистості. У цьому контексті роль інформатики є надважливою, оскільки саме цей предмет поєднує високий рівень абстракції з практичною спрямованістю. Він дозволяє природно переходити від абстрактних понять до конкретних прикладів, що робить навчання глибоким і водночас природним чи практичним.

Формування алгоритмічного мислення для старшокласників розглянуто у праці Дорошенко Ю. О. та Осіпої Л. В. [51]. Він доводить, що у ході дидактичного моделювання формування алгоритмічної культури учнів під час розв'язування обчислювальних задач, було досліджено характеристики навчального процесу, визначено його структуру, встановлено причинно-

наслідкові зв'язки. За допомогою використання інструментальних програмних засобів описано умови функціонування дидактичної системи та вплив зовнішніх чинників.

Дидактична модель розглядається як схематичне подання всіх компонентів навчального процесу. Це забезпечує його результативність, групування, встановлює зв'язки та послідовність застосування. Цілеспрямоване формування алгоритмічної культури вимагає відповідного організаційно-методичного забезпечення. Тому нами визначено дидактичні умови підвищення ефективності навчання:

- створення позитивної мотивації старшокласників до діяльності, спрямованої на розвиток алгоритмічних умінь;
- реалізація міжпредметних зв'язків інформатики з природничо-математичними дисциплінами;
- практична спрямованість змісту навчання та збільшення частки самостійної роботи;
- зворотний зв'язок і формування в учнів здатності до рефлексії.

Для практичної реалізації моделі авторами Дорошенко Ю. О. та Осіпою Л. В. розроблено авторський курс за вибором «Розв'язування обчислювальних задач з використанням інструментальних програмних засобів». Він є змістовим ядром і реалізатором визначених вище дидактичних умов, спрямованих на розвиток інтелектуальних здібностей, логічного й алгоритмічного мислення. Під час розв'язування спеціально дібраних задач створюються спеціальні калькулятори для автоматизованого розв'язання завдань певного типу.

Основою даного курсу стала система обчислювальних задач предметного змісту. Вона побудована за двома підходами – модельним та алгоритмічним. Модельний передбачає побудову математичної моделі як результату формалізації умови. Алгоритмічний підхід ґрунтується на структурній реалізації алгоритму. Задачі добираються з математики, фізики, астрономії, географії, біології тощо залежно від профілю школи та рівня підготовки учнів.

Їх послідовне розв'язання сприяє формуванню алгоритмічних знань, умінь і навичок, а також формує особистий досвід алгоритмічної діяльності [52].

Отже, нами проведено порівняльний аналіз змісту освіти та методичних підходів до розвитку алгоритмічного мислення на різних щаблях шкільного навчання – у ЗЗСО. З'ясовано що кожний етап має свої специфічні умови, які слід враховувати під час організації освітнього процесу.

У початкових класах формування алгоритмічного мислення відбувається переважно через ігрову діяльність, виконання вправ на логіку та послідовність дій. Навчальний матеріал подається у доступній для дітей формі, з використанням наочних моделей, вправ із простими умовами та наслідками, а також елементів програмування у візуальних середовищах.

У середній школі розумова діяльність учнів переходить на абстрактніший рівень. Діти навчаються працювати з алгоритмами, логічними операціями, циклами та умовами. Шкільні програми з інформатики, математики та природничих дисциплін мають бути спрямовані на розвиток уміння аналізувати, планувати й оптимізувати дії. Значну роль відіграє використання цифрових інструментів, середовищ програмування з використанням Scratch чи Blockly.

У старших класах алгоритмічне мислення стає системним. Учні вже здатні самостійно створювати алгоритми, розв'язувати складні завдання, працювати з реальними кейсами та мовами програмування. В навчальні програми варто включати проєктну діяльність, оптимізацію процесів і розвиток алгоритмічної культури як складника цифрової грамотності.

Тому, розвиток алгоритмічного мислення є поступовим процесом, що вимагає узгодженості змісту навчання, методик і засобів на всіх рівнях закладу освіти. А також необхідно врахувати вікові та когнітивні особливості учнів.

2.2. Інформаційно-освітні інструменти і ресурси для формування алгоритмічного мислення учнів

Інформатика, що вивчає інформацію та інформаційні процеси, має важливе значення для формування цілісного світогляду та системно-інформаційної картини світу. Ця наука динамічно розвивається та постійно розширює сфери практичного застосування. Одним із головних завдань шкільного курсу «Інформатика» є розвиток алгоритмічного мислення. Під час опанування предмета учні навчаються порівнювати, аналізувати, узагальнювати, абстрагувати, встановлювати структурні, ієрархічні та причинно-наслідкові зв'язки.

Науковці визначають алгоритмічне мислення як систему прийомів, способів і стратегій розв'язання теоретичних і практичних задач, результатом яких є алгоритми – особливі продукти людської діяльності. У процесі його формування школярі поступово набувають уявлень про алгоритм, його властивості, виконавців, різні форми подання та навичок створення й покрокового виконання алгоритмів, а також структурування власної діяльності. Такі вміння закріплюються через систему вправ, що виконуються протягом усього періоду навчання в початковій школі [54].

Виконання цих завдань забезпечується завдяки різноманітним ресурсам та сервісам: Scratch, Blockly, Kodable, CheckiO, BeetleBlocks, Lightbot, ScratchJr, Tynker, Code Land: Coding for Kids, CodeMonkey, CargoBot та Bee-Bot та інших.

У дослідженні Шкуренко О. В. представлені результати опитування, яке показало беззаперечну першість середовища Scratch, з яким знайомі всі опитані учні (100%). Такий результат підтверджує провідну роль цієї платформи у початковому формуванні алгоритмічного мислення. Інтуїтивний інтерфейс Scratch надає змогу створювати програми у вигляді блокових конструкцій, що полегшує сприйняття складних понять та поступово привчає дітей до логіки послідовностей, умов і циклів. Завдяки поєднанню ігрових елементів і творчості Scratch не лише навчає баз програмування, а й формує здатність мислити алгоритмічно та структурувати власні дії.

Водночас інші ресурси, такі як LightBot (26%), Pilas Bloques (22%), Blockly Games (16%), Tynker та CodeMonkey (по 10%) – хоч і мають меншу популярність, суттєво доповнюють освітній процес [55].

Для розвитку алгоритмічного мислення учнів можна використовувати різні онлайн-платформи, що поєднують навчання з елементами гри. Одним із прикладів є середовище Blockly, яке пропонує широкий вибір інтерактивних завдань і вправ. Він знайомить із принципами складання алгоритмів, близьких до реального коду. Blockly містить вісім рівнів поступового навчання: головоломки, лабіринти, керування тваринками, створення анімацій, музичних фрагментів та мініпрограм. На кожному етапі учні отримують серію завдань, виконання яких допомагає крок за кроком засвоювати основні команди, будувати прості та складні алгоритми і досягати визначеної мети.

Ще одним ефективним інструментом є середовище Pilas Bloques, розроблене для школярів, які вже мають базові знання програмування. Воно складається з двох блоків завдань: перший орієнтований на молодших або середніх учнів, другий – на тих, хто володіє початковими навичками алгоритмізації. Наприклад, у сценарії «Нагодуй котика Дубу» передбачено шість рівнів, де потрібно прокласти маршрут до їжі, використовуючи блоки руху та команду «з'їсти м'ясо». Інший сценарій «Котик починає малювати» має сім рівнів, де гравець завершує малюнок, поступово додаючи блоки малювання та перестрибування [20].

Використання таких сервісів допомагає дітям поступово переходити від візуального конструювання до текстового програмування, тренуючи аналітичне та структурне мислення, уміння оптимізувати рішення. Низький рівень їх упізнаваності підкреслює необхідність ширшого впровадження цих платформ у шкільну практику для поглиблення алгоритмічної компетентності [55].

Отже, Scratch залишається ключовою стартовою платформою, яку варто оглянути детальніше. Серед багатьох сучасних мов і середовищ програмування особливо вирізняється Scratch. Ця платформа поєднує в собі не лише технічні можливості, а й значний психолого-педагогічний та методичний потенціал, що

робить її цінним інструментом для навчання дітей і підлітків. Scratch відкриває широке поле для навчальної діяльності: від опанування базових принципів алгоритмізації до знайомства з основами об'єктно-орієнтованого, покрокового та навіть паралельного програмування [57].

Важливою перевагою цієї системи є можливість моделювання різноманітних об'єктів, процесів і явищ, що сприяє розвитку логічного мислення та вміння будувати причинно-наслідкові зв'язки. Scratch підтримує як індивідуальні, так і групові проекти, стимулює науково-пізнавальну активність і забезпечує міжпредметні інтеграції, дозволяючи поєднувати знання з інформатики, математики, мистецтва, природничих наук. Завдяки цьому платформа ефективно використовується як у навчальному процесі, так і в гуртковій або позашкільній роботі, зокрема в напрямі художньої та творчої діяльності.

Поєднуючи в собі елементи програмування та дизайну інтерфейсів, Scratch робить перші кроки у світ кодування цікавими та доступними. Візуальне середовище з блоками, що легко з'єднуються, знімає бар'єри складності й мотивує учнів експериментувати, створювати власні проекти та розвивати алгоритмічне мислення.

Основні особливості Scratch роблять його унікальним серед середовищ програмування, орієнтованих на навчання дітей та підлітків. Його головна перевага – блочне програмування, яке дозволяє створювати програми шляхом з'єднання яскравих графічних блоків, що «приєднуються» один до одного лише у правильних комбінаціях. Такий підхід практично усуває синтаксичні помилки та дає можливість вільно експериментувати, навіть під час виконання програми.

Ще одна важлива риса – широкі можливості роботи з даними. Scratch дозволяє керувати графікою, анімацією, звуками, музикою та іншими мультимедійними елементами, що особливо актуально в сучасній цифровій культурі. Учні можуть поєднувати різні типи інформації, створювати динамічні сцени й ігрові проекти, тренуючи логіку та творче мислення.

Scratch також вирізняється потужною культурою спільної роботи. Офіційний сайт надає відкритий доступ до тисяч готових проєктів: користувачі можуть переглядати роботи інших, редагувати їхні скрипти, додавати власні ідеї та ділитися результатами. Це не лише розвиває комунікаційні навички, а й формує відчуття належності до глобальної творчої спільноти.

Завдяки поєднанню різних підходів програмування Scratch навчає базових принципів процедурної логіки (послідовність дій, розгалуження, цикли, змінні, списки, робота з випадковими числами), знайомить з елементами об'єктно-орієнтованого програмування (створення об'єктів, їхні поля, методи, обробка подій) та паралельного виконання (запуск процесів у кількох потоках із синхронізацією). Крім того, він дає змогу створювати прості та зручні інтерфейси користувача, що наближає роботу до реального програмування [56].

Scratch сприяє розвитку логічного мислення учнів, адже вимагає від них чітко структурувати власні ідеї та плани перед тим, як перетворити їх на працюючий код. Під час роботи з проєктами школярі навчаються визначати послідовність дій для розв'язання завдань, розбиваючи складні проблеми на менші та більш керовані частини. Такий підхід, відомий як декомпозиція, є одним із ключових умінь у програмуванні й допомагає формувати навички аналізу та системного мислення.

Крім того, середовище Scratch передбачає використання умовних операторів і циклів, що вчить дітей застосовувати логічні міркування для управління перебігом програми. Це дозволяє не лише будувати алгоритми, а й передбачати різні сценарії їх виконання. Розвиток логічного мислення через роботу зі Scratch важливий не лише для здобуття технічних знань, а й для формування базових інтелектуальних здібностей: уміння планувати, передбачати наслідки та знаходити оптимальні рішення.

Наприклад, створюючи просту анімацію персонажа, учень повинен спочатку проаналізувати, які рухи має виконувати герой і як правильно запрограмувати їх за допомогою блоків. Учень мусить вирішити, коли персонаж починає рух, як швидко він рухається та коли зупиняється, що

вимагає логічного планування та алгоритмічної послідовності дій, що поступово формує навички, необхідні для подальшого вивчення складніших мов програмування.

Scratch не лише навчає основ програмування, а й активно формує алгоритмічне та критичне мислення. Під час налагодження проєктів учні аналізують власні алгоритми, знаходять і виправляють помилки, що допомагає глибше розуміти логіку виконання команд та значення точної послідовності дій. Планування проєктів вимагає продуманого добору функцій і ресурсів, розробки чітких кроків та передбачення результатів, що безпосередньо розвиває навички алгоритмізації [15].

Крім того, робота у спільноті Scratch стимулює обмін ідеями та спільний пошук ефективних рішень, що сприяє гнучкому мисленню й умінню адаптувати алгоритми під різні умови. Таким чином, Scratch стає не просто середовищем для створення програм, а й потужним інструментом комплексного розвитку – від логіки та стратегічного планування до творчого й соціального досвіду.

Scratch активно розвиває алгоритмічне мислення учнів завдяки своїй наочності та простоті. Програми створюються шляхом з'єднання графічних блоків, що допомагає легко бачити логічні зв'язки між елементами та розуміти принцип роботи коду. Такий підхід спрощує планування послідовності дій і дозволяє поступово засвоювати основи програмування.

Працюючи в Scratch, школярі складають алгоритми для розв'язання різних завдань. Щоб досягти мети, вони визначають порядок розміщення блоків, перевіряють зіткнення та рух, що розвиває навички логічного планування й аналізу. Особливе значення мають цикли та умовні оператори: здобувачі освіти застосовують команди повторення або перевірки умов, навчаючись оптимізувати свої рішення [44].

Усе це перетворює Scratch на інструмент, який не лише навчає алгоритмічному мисленню, а й стимулює креативність, самостійність і вміння перетворювати ідеї на готові програмні продукти. А завдяки візуальному

інтерфейсу та доступності Scratch стає потужним інструментом для формування базових понять програмування, логічного й критичного мислення, творчості та вмінь вирішувати практичні проблеми.

У сучасній школі розвиток алгоритмічного мислення учнів вимагає не лише чіткої методики, а й ретельно підібраних цифрових освітніх ресурсів. Представлена нами таблиця (Додаток Б) створена з метою систематизації найефективніших платформ, які сприяють формуванню навичок алгоритмізації на різних етапах навчання. Вона охоплює ключові характеристики середовищ – тип інтерфейсу, вікову орієнтацію, форми представлення алгоритмів, а також освітній потенціал кожного інструменту.

Такий формат дозволяє педагогам, методистам і розробникам програм швидко зіставити можливості платформ, обрати оптимальні засоби для конкретної вікової групи та навчальної мети. Узагальнення цифрових рішень у контексті алгоритмізації сприяє підвищенню якості навчання інформатики, забезпечує поступовий перехід від візуального до текстового програмування та формує у школярів логічне, структурне й проєктне мислення.

Отже, нами висвітлено основні напрями розвитку алгоритмічного мислення здобувачів освіти у межах сучасної освітньої системи. Розглянуто низку цифрових інструментів, які сприяють формуванню логічного мислення, навичок структурованого аналізу та базових умінь програмування. Акцент зроблено на таких платформах, як Scratch, Blockly та інших, що забезпечують плавний перехід від візуального моделювання до роботи з текстовим кодом.

Інтеграція цифрових освітніх середовищ у навчальні практики не лише активізує пізнавальну діяльність учнів, а й сприяє впровадженню компетентнісного підходу, орієнтованого на розвиток аналітичного мислення, здатності до самостійного створення алгоритмів та вирішення задач. У результаті чого алгоритмізація стає важливою складовою цифрової грамотності, формуючи інтелектуальну гнучкість і готовність школярів до викликів інформаційного суспільства.

2.3. Практична реалізація фрагментів уроків з формування алгоритмічного мислення здобувачів освіти 9 класів

Для обґрунтування практичної частини роботи важливо показати, як сучасні цифрові технології можуть сприяти розвитку алгоритмічного мислення учнів.

Тривимірне моделювання (3D-моделювання) – це процес формування цифрових об'єктів у тривимірному просторі за допомогою спеціальних комп'ютерних програм, які застосовуються для створення візуалізацій, симуляцій, анімацій і різноманітного виробництва в архітектурі, комп'ютерних іграх, рекламних проєктах та медичних розробках. Алгоритмізація при цьому виступає ключовим компонентом, адже вона задає чітку послідовність дій і команд, що дає змогу системі опрацьовувати інформацію, будувати й змінювати тривимірні моделі, формувати геометрію, керувати анімацією, здійснювати рендеринг та виконувати інші етапи роботи у відповідному програмному середовищі.

Сучасні 3D-технології надають можливість створювати наочні, реалістичні та інтерактивні моделі реальних предметів і процесів, що значно полегшує їх сприйняття та опрацювання учнями. Завдяки інтерактивності школярі можуть безпосередньо взаємодіяти з навчальними об'єктами, відпрацьовувати послідовність дій і спостерігати результат, що стимулює практичне мислення та формує навички алгоритмізації. Такі цифрові засоби підвищують інтерес до уроків і мотивують учнів до дослідження нових тем [58].

Останні десятиліття відзначаються стрімким зростанням ролі моделювання в науці, освіті та технологіях. Як зазначає Теплицький І. О., моделювання – це форма дослідження, за якої для отримання нових знань вивчають не сам об'єкт, а його спрощене відображення – модель. Такий підхід дозволяє безпечно, економно та ефективно аналізувати складні системи, прогнозувати результати й шукати оптимальні рішення.

Моделювання поступово стало невід'ємним елементом освітнього процесу, зокрема в галузі інформатики. Особливої актуальності воно набуває у контексті впровадження технологій тривимірного (3D) моделювання, яке активно використовується в архітектурі, промисловості, медицині, кінематографі, дизайні та освітніх практиках. Знайомство учнів із 3D-моделюванням не лише розвиває просторове мислення й технологічну грамотність, а й формує сучасний світогляд, готує до викликів цифрової епохи та майбутніх професій.

Для ефективного впровадження методики навчання 3D-моделювання важливо визначити, як ця тема представлена в освітніх системах різних країн, які існують підходи до організації занять і які технологічні інструменти використовуються. У багатьох країнах (зокрема Польщі, Фінляндії, Китаї, Латвії) елементи 3D-моделювання інтегруються в навчальні програми як окремі модулі або факультативи, часто в межах STEM-проектів. В інших – таких як Норвегія, Данія чи Франція – ця діяльність реалізується через позакласні ініціативи, гуртки або міждисциплінарні проекти.

В Україні розвиток 3D-моделювання у школах здебільшого ініціюється самими педагогами, які впроваджують нові технології на базі доступних ресурсів. Державний стандарт базової середньої освіти вже містить теми, пов'язані з тривимірною графікою, а також основи 3D-друку. На цій основі створено модельні програми, де окремі модулі присвячені створенню тривимірних об'єктів, редагуванню форм, візуалізації, анімації та рендерингу.

Такі навчальні програми сприяють формуванню в учнів умінь аналізувати об'єкти, передбачати наслідки рішень, застосовувати логіку та творчий підхід до виконання завдань. Крім того, 3D-моделювання розвиває критичне мислення, інформаційну культуру та здатність працювати з сучасними цифровими інструментами – навички, які визначають конкурентоспроможність учнів у майбутньому [59].

Отже, інтеграція 3D-моделювання в освітній процес Нової української школи є не лише сучасною тенденцією, а й необхідністю. Вона відкриває нові

можливості для реалізації творчого потенціалу учнів, робить навчання більш наочним, практичним і спрямованим на підготовку молоді до реальних потреб майбутнього.

Під час розробки 3D-моделей для курсу інформатики доцільно використовувати приклади, які не лише візуалізують абстрактні поняття, а й демонструють логіку виконання певних алгоритмів. Створюючи моделі, учні проходять етапи, схожі на побудову програмного коду: визначають послідовність операцій, перевіряють їх правильність, удосконалюють кроки. Наприклад, під час моделювання алгоритму можна показати, як елементи масиву змінюють позиції, а сам процес відображається у вигляді тривимірної анімації. Такий підхід допомагає зрозуміти не лише принцип роботи алгоритму, а й значення правильної логічної структури.

Використання 3D-графіки та моделювання відкриває нові перспективи для розвитку просторового та алгоритмічного мислення. Здатність уявляти об'єкти в тривимірному просторі, визначати їх форму, розміри, взаємне розташування та рух поєднується з умінням будувати чіткі алгоритми дій. Це сприяє розвитку креативності, навичок просторового аналізу й логічного планування, що особливо важливо під час вивчення не тільки інформатики, а й геометрії, фізики, STEM та інших предметів.

У межах практичної частини роботи розглянемо програму Blender як основний інструмент для створення тривимірних моделей з метою з'ясування та встановлення впливу процесу алгоритмізації на здобувачів освіти. Саме через роботу у цьому середовищі можна наочно простежити, як побудова послідовності дій, планування та виконання команд сприяють формуванню алгоритмічного мислення в учнів.

Використання платформи Blender у навчанні тривимірного моделювання є сучасним кроком у впровадженні інноваційних технологій, які відкривають широкі можливості для розвитку творчих і технічних навичок. Програма є безкоштовною та має відкритий вихідний код, що робить її доступною для будь-якого навчального закладу та кожного учня без додаткових фінансових

витрат. Вона поєднує в собі потужні засоби для моделювання, анімації, створення графічних текстур, рендерингу й інших процесів, необхідних для повноцінної роботи з тривимірними об'єктами. Завдяки цьому учні отримують змогу не лише будувати власні проєкти, а й практикувати алгоритмічний підхід: планувати етапи моделювання, розподіляти завдання, передбачати результат кожного кроку та виправляти помилки.

Blender має активну міжнародну спільноту, де можна знайти приклади готових робіт, навчальні матеріали й поради для початківців. Це значно полегшує процес навчання та дає можливість швидко отримувати підтримку або нові ідеї для проєктів. Разом з тим інтерфейс програми досить насичений, що може ускладнювати перші кроки для новачків і потребує певного часу на опанування. У порівнянні з комерційними аналогами кількість шкільних методичних посібників може бути меншою, а окремі складні функції вимагають потужнішого обладнання.

Попри ці труднощі, Blender залишається надзвичайно корисним інструментом для освітнього процесу. Його використання дає змогу учням поєднувати візуальне моделювання з чітким алгоритмічним мисленням, розвивати креативність, навички просторового аналізу та планування, що є важливими компетенціями для сучасних напрямів навчання та майбутніх професій.

Для цього ми провели вхідне тестування, яке дозволило визначити початковий рівень інтересу учнів до 3D-моделювання, їхнє розуміння алгоритмів та здатність послідовно планувати власні дії (Додаток В).

Перші завдання спрямовувалися на визначення інтересу до роботи з тривимірними об'єктами, адже саме внутрішня мотивація є важливою умовою залучення учнів до алгоритмічних дій і виконання складних завдань. Наступні питання перевіряли розуміння того, що таке алгоритм у простих діях, уміння описувати послідовність кроків під час створення 3D-моделей, а також уважність до порядку виконання команд. Такі показники дозволяють оцінити

початковий рівень логічного та структурованого мислення, без якого неможливо формувати навички алгоритмізації.

Окремий блок запитань був спрямований на виявлення аналітичного підходу учнів: їхню здатність пояснювати причини змін у поведінці об'єктів, помічати помилки й виправляти їх, що є основою для розвитку навичок дебагінгу та самокорекції. Не менш важливими були твердження, які показували, чи можуть здобувачі освіти переносити практичний досвід роботи у Blender на загальне розуміння принципів програмування, а також чи відзначають вони власний прогрес у логічному мисленні.

Завершальні питання анкети дозволяли визначити довгострокову мотивацію учнів та їхню готовність до самостійного алгоритмічного планування – від створення простої сцени до продуманих 3D-проектів.

Результати відповідей здобувачів освіти 9-А класу І групи Комунальна установа Сумська загальноосвітня школа І-ІІІ ступенів №27, м. Суми, Сумської області представлені нами у вигляді схем (Додаток Г). Аналіз отриманих результатів засвідчив неоднорідність рівня сформованості алгоритмічних умінь. Більшість учнів мають початкову мотивацію до роботи з алгоритмічними структурами, але потребують системного розвитку базових понять, практичних навичок та рефлексивних умінь.

Нами запропонований блок завдань, які здобувачі освіти виконували протягом циклу уроків на основі платформи Blender, мета яких зазначена у додатку (Додаток Д).

Після завершення цього етапу нами проведено вихідне тестування, де додалися запитання про подальше використання алгоритмічних структур та визначення інтересу здобувачів освіти до даної тематики.

У результаті аналізу вихідного анкетування учнів простежується позитивна динаміка у формуванні алгоритмічного мислення. Порівняно з початковими результатами. Здобувачі освіти демонструють зростання інтересу до роботи з тривимірними об'єктами, краще розуміють сутність алгоритму та виявляють готовність до самостійного планування дій. Це свідчить про

поступове засвоєння базових понять і розвиток навичок, пов'язаних із логічним мисленням, аналізом та рефлексією.

Значне покращення спостерігається у здатності пояснювати поведінку об'єктів, помічати послідовність дій та описувати алгоритми у контексті 3D-моделювання. Учні почали активніше висловлювати бажання продовжувати роботу з цифровими середовищами, що може бути наслідком практичної спрямованості занять, використання візуальних інструментів і залучення до творчих завдань.

Водночас окремі показники залишаються на рівні часткової сформованості: хоча більшість учнів виявляють інтерес до пошуку помилок та їх виправлення, але не всі готові до глибокої рефлексії або самостійного планування складних сцен. Це може бути пов'язано з недостатнім досвідом, обмеженим часом на практику або потребою в додатковій підтримці з боку педагога.

Загалом результати свідчать про ефективність обраних методичних підходів, зокрема використання ігрових форм, візуального моделювання та міжпредметної інтеграції. Формування алгоритмічного мислення відбувається поступово, і його успішність залежить від системності навчання, доступності інструментів та мотиваційного середовища, яке стимулює учнів до самостійного пошуку рішень і осмислення власних дій.

Отже, узагальнюючи результати проведеного дослідження, можна зробити висновок, що використання технологій тривимірного моделювання у процесі навчання інформатики має значний потенціал для розвитку алгоритмічного мислення здобувачів освіти. Інтеграція 3D-моделювання у навчальний процес сприяє формуванню в учнів уміння мислити послідовно, логічно та системно, планувати власну діяльність і усвідомлювати взаємозв'язок між етапами виконання завдань. Це не лише підсилює інтелектуальну активність, а й формує стійку мотивацію до пізнання, творчості та самостійного пошуку шляхів розв'язання поставлених проблем.

Практичне використання середовища Blender дало змогу поєднати візуальне моделювання з аналітичними процесами, необхідними для створення та відлагодження алгоритмів. Учні, працюючи над побудовою тривимірних сцен, фактично проходили ті самі етапи, що й під час програмування: аналіз умови, планування дій, реалізація та перевірка результатів. Такий підхід перетворює абстрактні поняття на зрозумілі, конкретні дії, формує вміння прогнозувати наслідки, оцінювати правильність виконаних кроків і виявляти помилки. Таким чином, діяльність у середовищі 3D-моделювання виступає ефективним засобом розвитку навичок дебагінгу, логічного аналізу та критичного мислення.

Результати вхідного та вихідного анкетування засвідчили помітне зростання інтересу учнів до роботи з тривимірними об'єктами, покращення розуміння сутності алгоритмів і підвищення здатності до самостійного планування дій. Якщо на початковому етапі більшість школярів демонстрували лише часткову обізнаність у послідовності алгоритмічних дій, то після циклу уроків на базі Blender простежується стійка позитивна динаміка: учні почали усвідомлювати логіку процесів, коректніше описувати етапи створення моделей, активніше виявляти ініціативу та творчість.

Особливо важливим результатом є те, що здобувачі освіти не лише навчилися користуватися інструментами тривимірного моделювання, а й почали сприймати власну діяльність як послідовний процес, у якому кожен крок має значення. Це свідчить про розвиток структурованого мислення, необхідного для подальшого вивчення програмування, математики та природничих дисциплін. Крім того, використання інтерактивних цифрових технологій сприяло підвищенню пізнавальної активності, формуванню комунікативних навичок під час обговорення проєктів, розвитку вміння працювати в команді та здійснювати самооцінку результатів.

Отримані дані підтверджують, що тривимірне моделювання може стати дієвим методичним інструментом для формування алгоритмічного мислення, особливо якщо воно супроводжується системною організацією навчального

процесу, інтеграцією в різні предмети та підтримується належною технічною базою. Позитивні зміни у рівні розуміння алгоритмічних структур і зростання навчальної мотивації демонструють ефективність поєднання візуальних засобів з алгоритмічним підходом до вирішення задач.

Підсумовуючи, можна стверджувати, що впровадження 3D-моделювання у шкільний курс інформатики є доцільним і перспективним напрямом, який відповідає вимогам сучасної освіти та викликам цифрової епохи. Воно забезпечує розвиток ключових компетентностей, формує готовність учнів до інноваційної діяльності, а головне — сприяє усвідомленому формуванню алгоритмічного мислення як фундаментальної складової майбутньої професійної підготовки.

ВИСНОВКИ

1. За результатами проведеного дослідження розкрито сутність поняття «алгоритмічне мислення» як системи логічних дій, спрямованих на розв'язання задач у цифровому середовищі. Обґрунтовано його значення в умовах трансформації освіти, де здатність до структурованого аналізу, планування та оптимізації рішень стає ключовою компетентністю здобувачів освіти.

Алгоритмічне мислення визначено як фундаментальний компонент цифрової грамотності, що забезпечує ефективну взаємодію з інформаційними технологіями. У роботі доведено, що його розвиток можливий через інтеграцію практичних завдань, рефлексивного аналізу та міжпредметних зв'язків, зокрема у процесі тривимірного моделювання в середовищі Blender.

Практична реалізація поетапного формування алгоритмічного мислення сприяє підвищенню мотивації учнів, розвитку їхньої здатності до самостійного планування, критичного осмислення та прийняття рішень. Впровадження таких підходів у навчальний процес створює умови для формування інноваційного освітнього середовища, яке відповідає викликам цифрової епохи та забезпечує якісну підготовку здобувачів освіти до життя в інформаційному суспільстві.

2. У результаті аналізу психолого-педагогічних умов розвитку алгоритмічного мислення у школярів встановлено, що ефективне формування цієї здатності потребує врахування вікових особливостей учнів, рівня їхньої когнітивної готовності та мотиваційної сфери. Найсприятливішим періодом для закладення логічних структур мислення є молодший шкільний вік, коли активно розвиваються пізнавальні процеси, здатність до системного аналізу та послідовного планування.

До ключових умов належать: створення навчального середовища, яке стимулює пошук, експериментування та рефлексію; використання завдань, що передбачають декомпозицію, виявлення шаблонів, абстрагування та побудову алгоритмів; застосування візуальних ігрових платформ, які забезпечують поступовий перехід до текстового програмування. Важливу роль відіграє

підтримка самостійності учнів, розвиток їхньої здатності до критичного осмислення та прийняття рішень у нестандартних ситуаціях.

Таким чином, психолого-педагогічні умови формування алгоритмічного мислення охоплюють не лише дидактичні аспекти, а й емоційно-мотиваційні чинники, що забезпечують цілісне включення здобувача освіти в процес пізнання, аналізу та конструювання алгоритмів у контексті цифрової трансформації освіти.

3. У ході дослідження охарактеризовано методичні підходи та сучасні освітні технології, що сприяють формуванню алгоритмічного мислення на уроках інформатики. Встановлено, що ефективність навчального процесу забезпечується поєднанням поетапного формування навичок алгоритмізації, використанням візуальних середовищ програмування та інтеграцією міжпредметних зв'язків.

До провідних методичних рішень належать: застосування задач з декомпозицією, виявленням шаблонів і абстрагуванням; використання платформ Scratch, Blockly, Blender для моделювання алгоритмів у графічному та тривимірному форматах; організація навчання через практичні проєкти, що стимулюють самостійність і рефлексію. Освітні технології, орієнтовані на розвиток алгоритмічного мислення, дозволяють учням не лише засвоїти базові поняття інформатики, а й сформувати здатність до логічного планування, аналізу та оптимізації рішень у цифровому середовищі.

4. У процесі дослідження визначено ключові особливості проєктування освітнього середовища, спрямованого на розвиток алгоритмічного мислення учнів. Таке середовище має бути динамічним, технологічно насиченим і методично обґрунтованим. Його основою є поєднання цифрових ресурсів, інструментів візуального та тривимірного моделювання, зокрема середовища Blender, а також створення умов для практичного застосування знань.

До важливих характеристик освітнього середовища належать: наявність доступу до сучасних програмних засобів; організація навчальної діяльності

через проєкти, що передбачають декомпозицію, абстрагування та побудову алгоритмів; підтримка індивідуального темпу навчання та рефлексії. Особливу увагу приділено міжпредметній інтеграції, яка дозволяє учням застосовувати алгоритмічне мислення в різних навчальних контекстах.

Таким чином, ефективне освітнє середовище для розвитку алгоритмічного мислення – це не лише технічна інфраструктура, а й педагогічно продумана система, що стимулює учнів до логічного аналізу, самостійного планування та пошуку оптимальних рішень у цифровому просторі.

5. У межах практичної частини дослідження розроблено дидактичну модель фрагментів уроків інформатики для учнів 9 класу, що передбачає використання технологій тривимірного моделювання у середовищі Blender. Модель побудована на основі поетапного формування алгоритмічного мислення, з урахуванням вікових особливостей здобувачів освіти, їхньої мотивації та рівня когнітивної готовності до роботи з цифровими інструментами.

Запропоновані фрагменти уроків включали такі ключові компоненти:

- вступне мотиваційне завдання, що активізує інтерес до теми через візуальні приклади 3D-моделей;
- інструктивно-практичний блок, де учні знайомляться з базовими функціями Blender, навчаються створювати об'єкти, задавати координати, використовувати трансформації та логічні послідовності дій;
- рефлексивний етап, на якому учні аналізують власні алгоритми, виявляють помилки, оптимізують рішення та обговорюють варіанти покращення моделей;
- міжпредметна інтеграція, що дозволяє поєднувати знання з інформатики, математики, геометрії та мистецтва, формуючи цілісне бачення задачі.

Особливу увагу приділено декомпозиції завдань: учні навчаються розбивати складні проєкти на окремі етапи – створення сцени, моделювання

об'єктів, налаштування руху, додавання логіки. Це сприяє розвитку структурного мислення та здатності планувати дії послідовно. У процесі роботи з Blender учні також опановують елементи об'єктно-орієнтованого мислення, адже кожен об'єкт має властивості, методи та взаємодіє з іншими елементами сцени.

Подані нами практичні завдання моделі включають:

- створення простих 3D-об'єктів (куб, сфера, персонаж);
- побудову алгоритму руху об'єкта за заданими координатами;
- моделювання сцени з умовною логікою (наприклад, об'єкт рухається лише при виконанні певної умови);
- оптимізацію сценарію через використання циклів, умовних операторів та функцій.

Результати впровадження моделі показали, що учні демонструють зростання рівня алгоритмічного мислення, підвищення мотивації до навчання, розвиток рефлексивних навичок та здатності до самостійного планування. Робота з Blender виявилася ефективним засобом для формування логіки, послідовності, абстрагування та візуального представлення алгоритмів.

Таким чином, розроблена дидактична модель є прикладом інтеграції сучасних освітніх технологій у шкільний курс інформатики, що відповідає вимогам цифрової трансформації освіти та сприяє формуванню ключових компетентностей здобувачів освіти у сфері алгоритмізації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Нікітенко В. О. Взаємодія освіти, культури, туризму та їх вплив на розвиток креативного потенціалу особистості в умовах інноваційно-інформаційного суспільства : автореф. дис. на здобуття наук. ступеня доктора філософ. наук спец. 09.00.03 «Соціальна філософія та філософія історії» / О. В. Нікітенко. – Запоріжжя, 2020. – 39 с.
2. Ziatdinov R., Musa S. Rapid mental computation system as a tool for algorithmic thinking of elementary school students development / R. Ziatdinov, S. Musa // European Researcher. – 2012. – Vol. 25, № 7. – P. 1105–1110.
3. Крупа А. Г., Мар'єнко В. Ю. Вплив інформаційно-комунікаційних технологій (ІКТ) на цифрову трансформацію суспільства: соціально-філософський аналіз / А. Г. Крупа, В. Ю. Мар'єнко // Освітній дискурс : зб. наук. праць. – 2025. – Вип. 52(1–2). – С. 53–60
4. Звіт про семінар на тему «Обсяг та природа обчислювального мислення» / [Електронний ресурс]. – Режим доступу: <https://nap.nationalacademies.org/read/12840/chapter/1>. – Назва з екрану.
5. Ольхова Н. В. Формування алгоритмічного мислення здобувачів початкової освіти під час вивчення інформатики / Н. В. Ольхова // Педагогічний процес: теорія і практика. – 2022. – Т. 8, № 2. – С. 45–52.
6. Пасічник О. В. Розвиток алгоритмічного мислення на уроках інформатики / О. В. Пасічник // Комп'ютер у школі та сім'ї : науково-методичний журнал. – 2014. – № 7/8. – С. 13–18.
7. Гладун М. А., Морзе Н. В. Система вправ з інформатики для формування алгоритмічного мислення в учнів молодших класів / М. А. Гладун, Н. В. Морзе // Інформатика та інформаційні технології в навчальних закладах. – 2013. – № 4(46). – С. 41–49.
8. Волошинов С. А. Алгоритмічна підготовка майбутніх судноводіїв системою візуальної підтримки в умовах інформаційно-комунікаційного

педагогічного середовища : дис. ... канд. пед. наук : 13.00.04 / Волошинов Сергій Анатолійович. – Херсон, 2012. – 244 с.

9. Барболіна Т. М. Розвиток алгоритмічного й операційного мислення у процесі вивчення прикладного програмного забезпечення / Т. М. Барболіна. – К. : Генезис, 2010. – С. 19–22.

10. Копаєв О. В. Алгоритм як модель алгоритмічного процесу // Комп'ютерно-орієнтовані системи навчання : зб. наук. праць. – К. : НПУ ім. М. П. Драгоманова, 2003. – Вип. 6. – С. 206–213.

11. Культура споживання інформації: збірник матеріалів молодіжного наукового форуму «DIGITAL THINK: МОЛОДІЖНИЙ НАУКОВИЙ ФОРУМ» : зб. наук. праць. – Суми : СумДПУ ім. А. С. Макаренка, 2025. – Суми-Ужгород. – 100 с.

12. Медведєва М. О., Жмурко О. І., Криворучко І. І., Ковтанюк М. С. Елементи підготовки майбутніх учителів інформатики до застосування технології формування Computational Thinking // Фізико-математична освіта. – Суми : СумДПУ ім. А. С. Макаренка, 2021. – Вип. 1(27). – С. 67–75.

13. Пасічник О. В., Чернікова Л. А. Модельна навчальна програма «Інформатика. 5–6 класи» для закладів загальної середньої освіти. Рекомендовано МОН України (наказ від 12.07.2021 № 795) / [Електронний ресурс]. – Режим доступу: https://osvita.ua/doc/files/news/831/83155/Inform_5-6-kl_Pasichnyk_Chernikova_14_07_21.pdf. – Назва з екрану.

14. Byrka M., Sushchenko A., Luchko V., Perun G., Luchko V. Algorithmic thinking in higher education: determining observable and measurable content / M. Byrka, A. Sushchenko, V. Luchko, G. Perun, V. Luchko // Information Technologies and Learning Tools. – 2024. – Vol. 104, №6 – P. 1–13.

15. Великдан Ю. Можливості Scratch у формуванні алгоритмічного і логічного мислення учнів базової школи / Ю. Великдан // Вісник Черкаського національного університету імені Богдана Хмельницького. Серія «Педагогічні науки». – 2025. – № 1. – С. 172–178.

16. Adorni G., Piatti A. Designing the virtual CAT: A digital tool for algorithmic thinking assessment in compulsory education [Електронний ресурс] / arXiv preprint. – 2024. – № arXiv:2408.01263v3. – Режим доступу: <https://arxiv.org/abs/2408.01263v3>. – Назва з екрану.

17. Wang Y., Lopez-Pernas S., Seidel S., Perez-Sanagustin M. Development of algorithmic thinking skills in K-12 education: A comparative study of unplugged and digital assessment instruments [Електронний ресурс] / Y. Wang, S. Lopez-Pernas, S. Seidel, M. Perez-Sanagustin // ResearchGate. – 2024. – Режим доступу: <https://www.researchgate.net/publication/383470038>. – Назва з екрану.

18. Порохняк Д. Р., Романишина О. Я. Формування алгоритмічного мислення під час розв'язування олімпіадних задач з інформаційних технологій у офісних програмах / Д. Р. Порохняк, О. Я. Романишина // Сучасні цифрові технології та інноваційні методики навчання: досвід, тенденції, перспективи : матеріали Всеукраїнської науково-практичної конференції, 10 квітня 2025 р. – Вип. 15. – С. 96–98.

19. Освіта.UA. ІШ та STEM: як технології змінюють навчання в Україні / [Електронний ресурс]. – Режим доступу: <https://osvita.ua/vnz/reform/94290/>. – Назва з екрану.

20. Баранов С., Олефіренко Н. Шляхи розвитку алгоритмічного мислення в учнів початкової школи / С Баранов., Н. Олефіренко // Науково-дослідна робота студентів як чинник удосконалення професійної підготовки майбутнього вчителя: зб. наук. праць. – Харків, 2024. Вип. 23. – С. 15–20.

21. Сарієнко В. Дидактична функція формування алгоритмічного мислення // Професіоналізм педагога: теоретичні й методичні аспекти. – Слов'янськ : СДПУ, 2018. – Вип. 8. – С. 91–99.

22. Рибалко О. О. Алгоритми та математика у початковій школі / О. О. Рибалко // Комп'ютер у школі та сім'ї. – 2014. – № 3. – С. 26–29.

23. Воробей О. О. Викладання основ алгоритмізації у 5-му класі / О. О. Воробей // Комп'ютер у школі та сім'ї. – 2013. – № 2. – С. 7–9.

24. Chojak M. Algorithms in the educational process – opportunities and limitations / Chojak M. // *Rocznik Pedagogiczny*. – 2024. – Т. 43, з. 4. – Р. 75–89.

25. Таранченко О. М. Алгоритмізація організації навчального процесу в інклюзивному класі / О. М. Таранченко // *Спеціальна та інклюзивна освіта: теорія, методика, практика : матеріали 7-ої Всеукраїнської науково-практичної конференції (25 березня 2021 р.)*. – Умань: Візаві, 2021. – Вип. 7. – С. 169–174.

26. Algorithm: Psychology Definition, History & Examples [Електронний ресурс]. – Режим доступу: <https://www.zimbardo.com/algorithm-psychology-definition-history-examples/>. – Назва з екрану.

27. Алгоритмічний метод [Електронний ресурс]. – Режим доступу: https://stud.com.ua/131883/pedagogika/algoritmichniy_metod. – Назва з екрану.

28. Вдовенко В. В. Формування алгоритмічного мислення молодших школярів на уроках інформатики / В. В. Вдовенко // *Наукові записки. – Кропивницький*. – 2017. – Вип. 11. – С. 23–27.

29. Пасічник О. В., Чернікова Л. А. Навчальна програма «Інформатика» 5-6 класи: розроблена на основі модельної навчальної програми «Інформатика (5-6 класи)» для закладів загальної середньої освіти [Електронний ресурс]. – Режим доступу: <https://bit.ly/3ZYW4Eu>. – Назва з екрану.

30. Навчальні програми для 1–4 класів. Програми Нової української школи [Електронний ресурс]. – Режим доступу: <https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/osvitni-programi/navchalni-programi-dlya-1-4-klasiv>. – Назва з екрану.

31. Коршунова О. В. Сходінки до інформатики: [підручник для 3-го класу загальноосвітніх навчальних закладів] / О. В. Коршунова. – К. : Генеза, 2014. – 176 с.

32. Волощенко О. В., Козак О. П., Остапенко Г. С. Я досліджую світ: [підручник для 3 класу закладів загальної середньої освіти] / О. В. Волощенко, О. П. Козак, Г. С. Остапенко. – К. : Світич, 2020. – 143 с.

33. Інформатика: [підручник для 4 кл. закладів загальної середньої освіти / [Т. Г. Гільберг, О. В. Суховірський, Л. В. Грубіян та ін.]. – К. : Генеза, 2021. – 128 с.

34. Ривкінд Й. Я., Лисенко Т. І., Чернікова Л. А. Модельна навчальна програма «Інформатика. 5–6 класи» для закладів загальної середньої освіти [Електронний ресурс]. – Режим доступу: <https://drive.google.com/file/d/12IcMma6w2VBltwxGWBbDI9TkoSQ9SDQt/view> – Назва з екрану.

35. Самойленко Н. І. Компетентнісний підхід до навчання інформатики в основній школі / Н. І. Самойленко // Тези доповідей III Міжнародної науково-практичної конференції «Інформаційні технології в освіті, науці і техніці», 15–17 травня 2013 р., м. Черкаси. – Черкаси : ЧНУ ім. Б. Хмельницького, 2013. – С. 64–65.

36. Методика викладання інформатики у 5–7 класах НУШ // Всеосвіта. – 2024. – Режим доступу: <https://vsimosvita.com/metodyka-vykladannya-informatyky-u-5-7-klasaх-nush> – Назва з екрана.

37. Семко Л. П. Вимоги до змісту навчання інформатики в 7–9 класах закладів загальної середньої освіти / Л. П. Семко // Цифрові інструменти в роботі сучасного педагога: оптимізація та інновації : матеріали Всеукраїнської наук.-практ. інтернет-конф. – 2024. – С. 177–179.

38. Завадський І. Еволюція концепції компетентнісно-орієнтованого навчання в курсі інформатики в українській школі / І. Завадський // Матеріали міжнародної науково-практичної конференції «Інноваційні освітні технології в умовах цифрової трансформації освіти» (Київ, 29 березня 2021 р.). – Київ, 2021. – С. 74–76.

39. Самойленко Н. І., Семко Л. П. Методичні підходи до вивчення інформатики в основній школі / Н. І. Самойленко, Л. П. Семко // Педагогічні науки: теорія, історія, інновації. – 2015. – Вип. 1 (13). – С. 68–74.

40. Кошелєв О., Пасічник Н. Особливості формування алгоритмічного мислення молодших школярів на уроках інформатики / О. Кошелєв, Н. Пасічник / Молодь і ринок. – 2021. – 90–94.

41. Лодатко Є. О., Мочула О. В., Яріш Г. П. Інформатика: [підручник для 4 класу закладів загальної середньої освіти] / Є. О. Лодатко, О. В. Мочула, Г. П. Яріш. – Тернопіль : Навчальна книга – Богдан, 2021. – 160 с.

42. Дудка О. М., Власій Н. М., Магомета Н. М. Реалізація компетентнісного підходу до вивчення програмування на Scratch / О. М. Дудка, Н. М. Власій, Н. М. Магомета // Відкрите освітнє е-середовище сучасного університету. – 2018. – Вип. 5. – С. 88–96.

43. Яценко О. І. Середовище програмування «Scratch»: аналіз можливостей використання з метою формування інформативних компетентностей вчителя початкової школи / О. І. Яценко // Актуальні питання сучасної інформатики. – 2017. – Вип. 5. – С. 276–278.

44. Cantlon J. F., Becker K. T., DeLong C. M. Computational thinking during a short, authentic, interdisciplinary STEM experience for elementary students / J. F. Cantlon, K. T. Becker, C. M. DeLong // Journal for STEM Education Research. – 2024. – Vol. 7. – P. 425–443.

45. Husanova S. H. Psychological and pedagogical foundations for the formation of computative thinking / Husanova S. H. // Pedagogical Cluster – Journal of Pedagogical Developments (PCJPD). – 2024. – Vol. 2, Issue 10. – P. 37–42.

46. Лісневська О. С. Інтеграція уроків математики та трудового навчання як засіб розвитку творчого мислення та просторової уяви молодших школярів [Електронний ресурс] / О. С. Лісневська // Портал «Studentam.net.ua». – Режим доступу: <https://studentam.net.ua/content/view/7544/97/>. – Назва з екрану.

47. Малахов А., Остапенко Л. Алгоритмічне мислення та шляхи його формування в учнів / Малахов А., Остапенко Л. // Інноваційні педагогічні технології в цифровій школі : зб. тез доп. учасників VI Міжнародної наук.-практ. конф. молодих учених (м. Харків, 15–16 трав. 2024 р.). – Харків : ХНПУ ім. Г. С. Сковороди, 2024. – С. 121–122.

48. Початкова освіта: методичні рекомендації щодо використання в освітньому процесі Типової освітньої програми для 1 класів закладів загальної середньої освіти; типова освітня програма для закладів загальної середньої освіти (колективу авторів під керівництвом О. Я. Савченко) : методичні рекомендації провідних науковців Інституту педагогіки НАПН України щодо впровадження ідей Нової української школи в початковій освіті. – Київ : УОВЦ «Оріон», 2018. – 160 с.

49. Саган О. В. Методика вивчення алгоритмів у початкових класах / О. В. Саган // Початкова школа. – 2017. – № 6. – С. 26–33.

50. Типові освітні програми для закладів загальної середньої освіти: 1–2 та 3–4 класи. – Київ : Видавництво «Світоч», 2019. – 336 с.

51. Дорошенко Ю. О., Осіпа Л. В. Дидактичне моделювання формування алгоритмічної культури старшокласників [Електронний ресурс] / Ю. О. Дорошенко, Л. В. Осіпа // Інститут педагогіки НАПН України. – Режим доступу: <https://lib.iitta.gov.ua/id/eprint/9252/1/%D0%94%D0%BE%D1%80%D0%BE%D1%88%D0%B5%D0%BD%D0%BA%D0%BE%D1%81%D1%82%D0%B0%D1%82%D1%82%D1%8F.pdf>. – Назва з екрана.

52. Осіпа Л. В. Формування алгоритмічної культури старшокласників у процесі розв'язування обчислювальних задач з використанням інструментальних програмних засобів [Електронний ресурс] : автореф. дис. ... канд. пед. наук : 13.00.09 – теорія навчання / Л. В. Осіпа. – Київ, 2014. – 22 с. – Режим доступу: https://lib.iitta.gov.ua/id/eprint/9257/1/a%D0%B2%D1%82%D0%BE%D1%80%D0%B5%D1%84_Osipa.pdf. – Назва з екрана.

53. Петрик Л. Ф. Розвиток алгоритмічного мислення на уроках інформатики [Електронний ресурс] / Л. Ф. Петрик // ВсімОсвіта. – Режим доступу: <https://vsimosvita.com/rozvytok-algorytmichnogo-myslennya-na-urokah-informatyky/>. – Назва з екрана.

54. Гнатенко А. В. Формування алгоритмічного мислення учнів початкової школи засобами інформаційно-комунікаційних технологій [Електронний ресурс] / А. В. Гнатенко // Матеріали студентської науково-практичної конференції. – Кременчуцький національний університет імені Михайла Остроградського, 2019. – Режим доступу: <https://conf.ztu.edu.ua/wp-content/uploads/2019/12/116-3.pdf>. – Назва з екрана.

55. Шкуренко О. В., Смик А. О. Методичні особливості вивчення теми «Алгоритми і виконавці» у початковій школі [Електронний ресурс] / О. В. Шкуренко, А. О. Смик // Молодий вчений. – 2024. – № 7 (131). – Режим доступу: https://elibrary.kubg.edu.ua/id/eprint/50328/1/O_Shkurenko_A_Smyk.pdf. – Назва з екрана.

56. Кирстич І. П., Василенко Я. П. Про педагогічні та дидактичні особливості середовища Scratch як інструменту навчання основам алгоритмізації та програмування / І. П. Кирстич, Я. П. Василенко // Сучасні інформаційні технології та інноваційні методики навчання: досвід, тенденції, перспективи : матеріали III Міжнародної науково-практичної інтернет-конференції (м. Тернопіль, 5 квітня 2019 р.). – Тернопіль : ТНПУ ім. В. Гнатюка, 2019. – С. 21–24.

57. Машталір О. В., Лещук С. О., Дільна Н. З. Пропедевтика алгоритмізації засобами Scratch / О. В. Машталір, С. О. Лещук, Н. З. Дільна // Сучасні інформаційні технології та інноваційні методики навчання: досвід, тенденції, перспективи : матеріали IV Міжнародної науково-практичної інтернет-конференції (м. Тернопіль, 30 квітня 2020 р.). – Тернопіль : ТНПУ ім. В. Гнатюка, 2020. – С. 12–16.

58. Поліщук В. В. 3D технології в освіті / В. В. Поліщук // Актуальні питання сучасної інформатики. – 2017. – № 4. – С. 97–100.

59. Твердохліб І. В. Розвиток алгоритмічного мислення учнів засобами 3D-моделювання в умовах цифрового освітнього середовища [Електронний ресурс] / І. В. Твердохліб, А. С. Деркач // Психолого-педагогічні студії : зб.

наук. праць. – 2023. – № 1. – С. 113–121. – Режим доступу:
https://lib.iitta.gov.ua/id/eprint/739268/1/Твердохліб_Деркач%20PSP%202023.pdf.

– Назва з екрана.

fizmat@sspi.edu.ua
суворо дотримуйтесь
правил академічної
доброчесності

Додаток А

Умови організації навчальної діяльності здобувачів освіти для розвитку алгоритмічного мислення	
Психолого-педагогічні умови	мотивація
	соціальна взаємодія
	створення проблемних ситуацій
	вікова готовність до алгоритмізації
Методичні умови	інтеграція алгоритмічного мислення в інформатику
	інтеграція алгоритмічного мислення в математику
	інтеграція алгоритмічного мислення в технології
Технічні умови	наявність комп'ютерів
	наявність планшетів
	доступ до інтернету
	інтерактивні платформи
Організаційні умови	формат занять
	тривалість та частота занять
	підтримка з боку педагогів і батьків

Додаток Б

Назва ресурсу/платформи	Тип середовища	Вікова категорія	Форми алгоритмізації	Освітні можливості	Педагогічні особливості
Scratch	Візуальне, блокове середовище	6–12 років	Послідовність, цикл, умова	Створення ігор, анімацій, інтерактивних історій	Ігрова мотивація, розвиток творчості, доступність
Blockly Games	Візуальне, блокове середовище	8–14 років	Побудова алгоритмів, логіка	Головоломки, задачі на логіку	Поступове ускладнення, розвиток структурного мислення
Code.org	Візуальне + текстове середовище	6–16 років	Алгоритми, функції, події	Курси з героями, інтерактивні вирази	Інтеграція з навчальними програмами, адаптивність
LightBot	Ігрове середовище	7–12 років	Команди, цикли, умовні оператори	Логічні задачі, симуляція виконавця	Візуалізація алгоритмів, розвиток планування
Tynker	Візуальне + текстове середовище	8–16 років	Алгоритми, змінні, функції	Створення ігор, модифікація Minecraft	Гейміфікація, STEM-орієнтація, індивідуалізація
CodeCombat	Текстове середовище з реальним кодом	10–17 років	Алгоритми, об'єкти, функції	Python, JavaScript, сюжетні місії	Переходи до професійного коду, мотивація через сюжет
Roblox Studio	Ігрове, текстове середовище	12+ років	Сценарії, події, логіка	Створення 3D-ігор, Lua-програмування	Поглиблена алгоритмізація, командна робота
Unity	Професійне середовище	14+ років	Повноцінне об'єктно-орієнтоване програмування (ООП), складні алгоритми	Розробка ігор, симуляцій, інтерактивних додатків	Профорієнтація, інтеграція з STEAM-напрямами, проєктне навчання

Додаток В

Анкета для учнів (вхідне анкетування)

№	Питання	Так	Частково	Ні
1	Мені цікаво працювати з тривимірними об'єктами.			
2	Я розумію, що таке алгоритм у простих діях.			
3	Я можу описати послідовність дій для створення 3D-моделі.			
4	Я помічаю, коли дії потрібно виконувати в певному порядку.			
5	Я можу пояснити, чому об'єкт поводить себе саме так (рухається, змінюється).			
6	Мені подобається шукати помилки і виправляти їх у своїй роботі.			

Анкета для учнів (вихідне анкетування)

№	Питання	Так	Частково	Ні
1	Мені цікаво працювати з тривимірними об'єктами.			
2	Я розумію, що таке алгоритм у простих діях.			
3	Я можу описати послідовність дій для створення 3D-моделі.			
4	Я помічаю, коли дії потрібно виконувати в певному порядку.			
5	Я можу пояснити, чому об'єкт поводить себе саме так (рухається, змінюється).			
6	Мені подобається шукати помилки і виправляти їх у своїй роботі.			
7	Я почав(ла) краще розуміти, як працюють комп'ютерні програми.			
8	Я хочу й надалі працювати з 3D-моделюванням.			
9	Я відчуваю, що моє мислення стало більш логічним.			
10	Я можу самостійно спланувати просту 3D-сцену.			

Вхідне анкетування

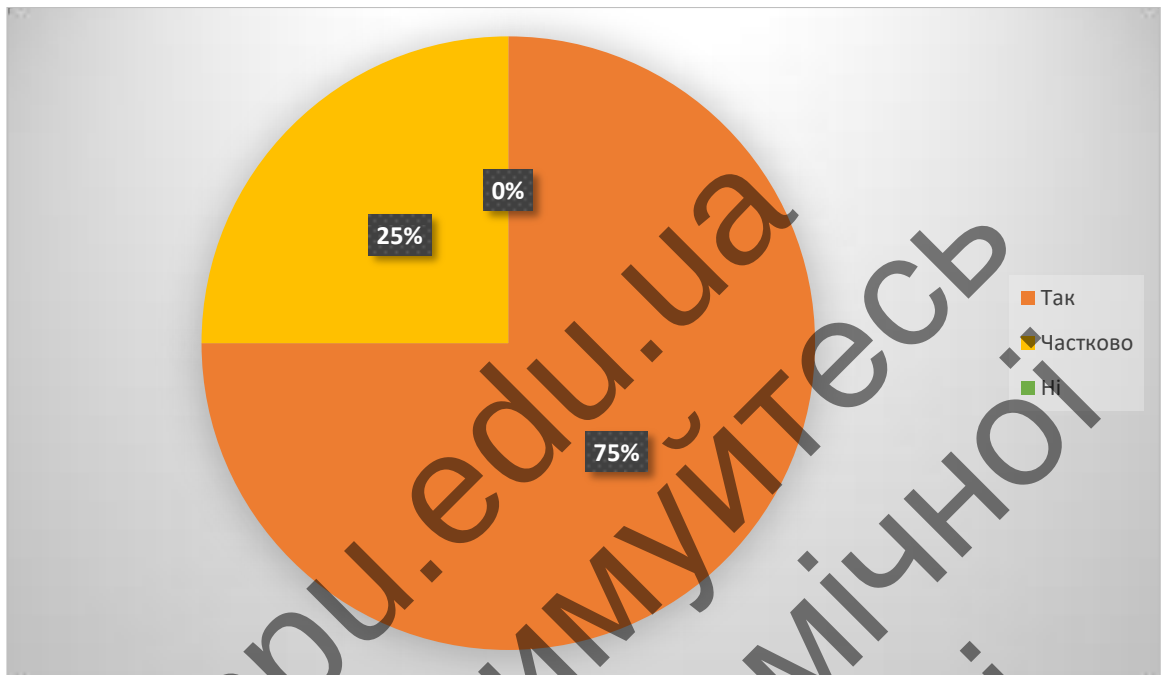


Рис. 1. Мені цікаво працювати з тривимірними об'єктами

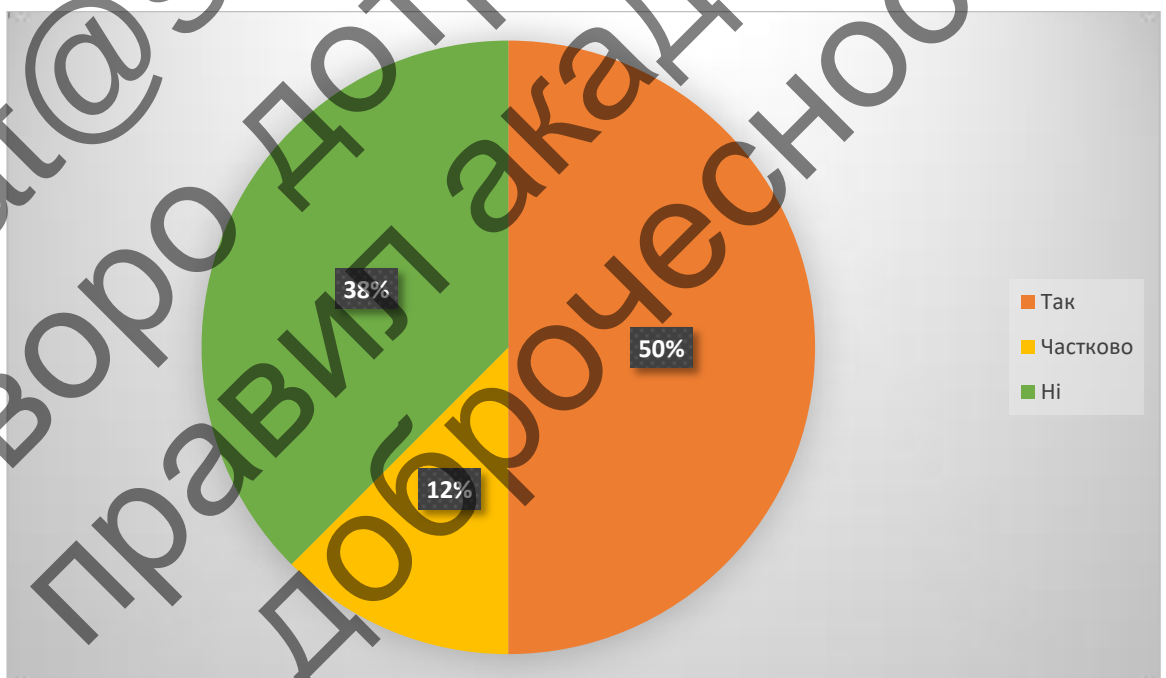


Рис. 2. Я розумію, що таке алгоритм у простих діях

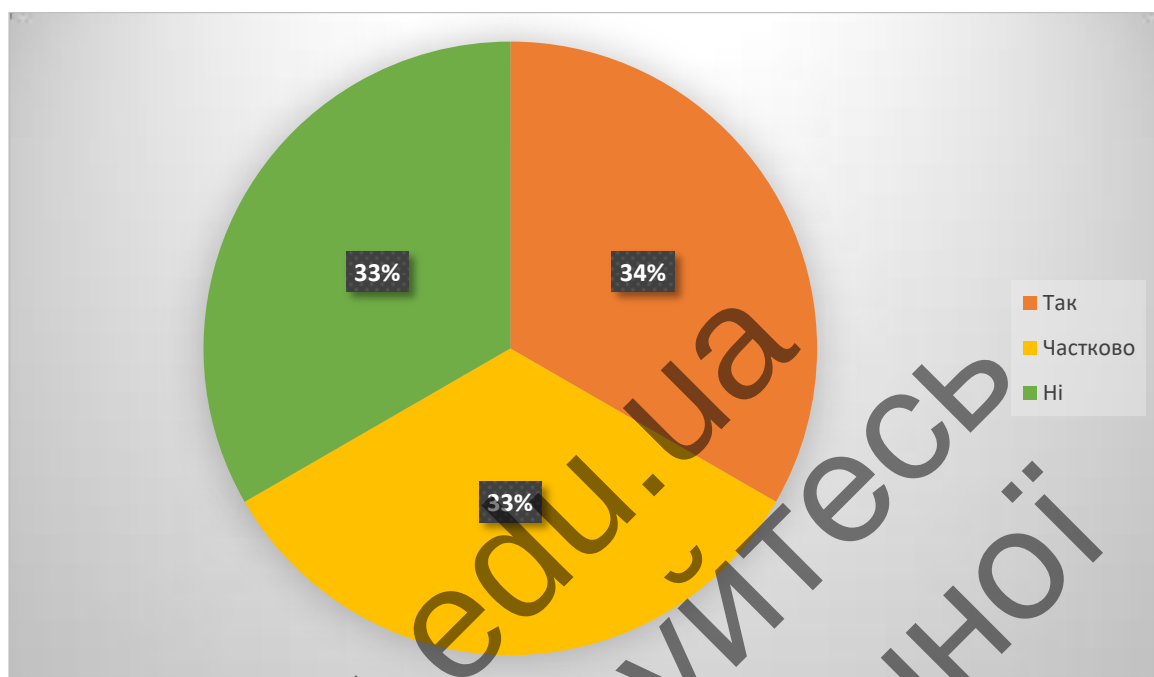


Рис. 3. Я можу описати послідовність дій для створення 3D-моделі

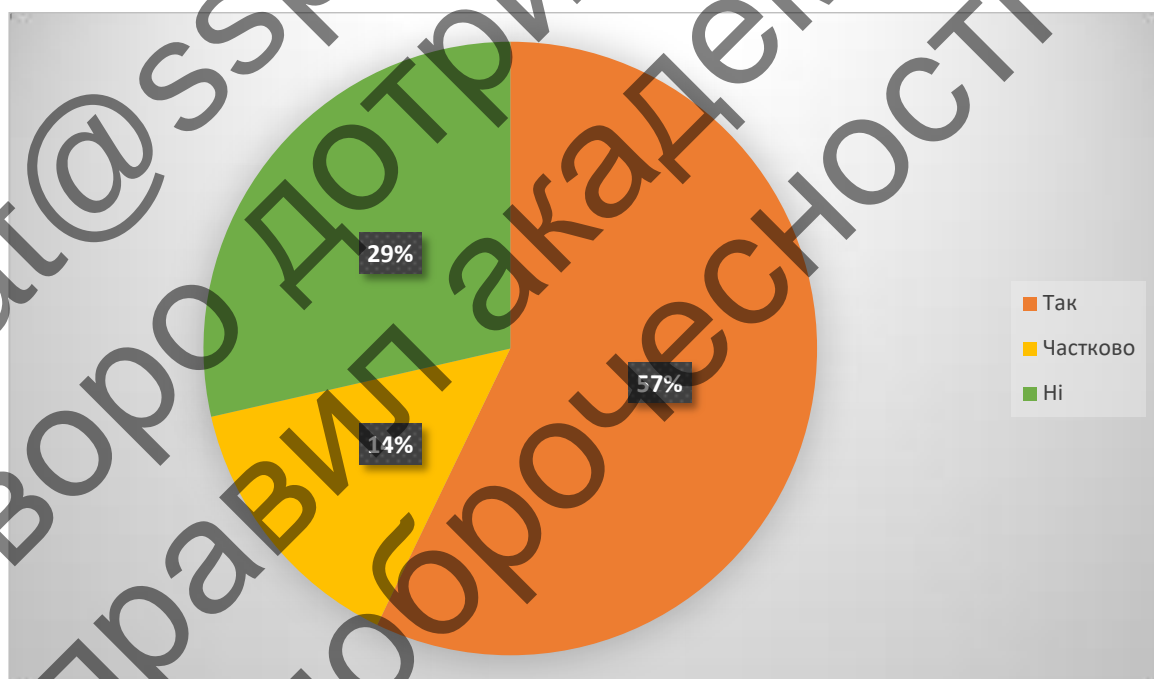


Рис. 4. Я помічаю, коли дії потрібно виконувати в певному порядку

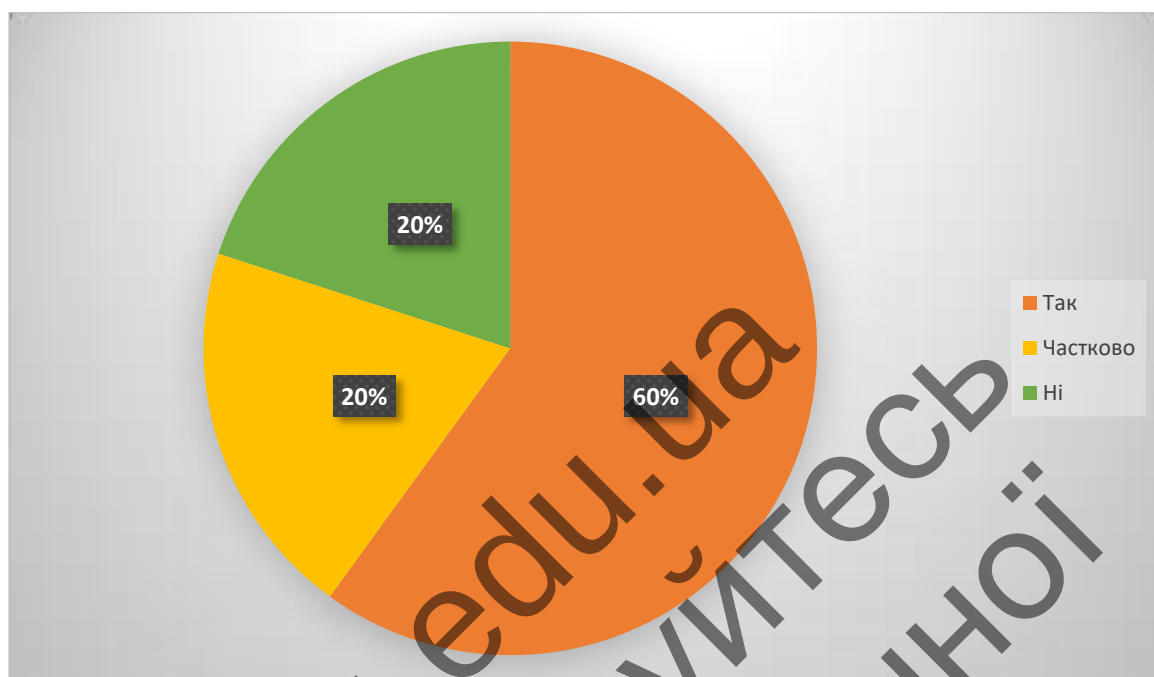


Рис. 5. Я можу пояснити, чому об'єкт поводитьсь саме так (рухається, змінюється)

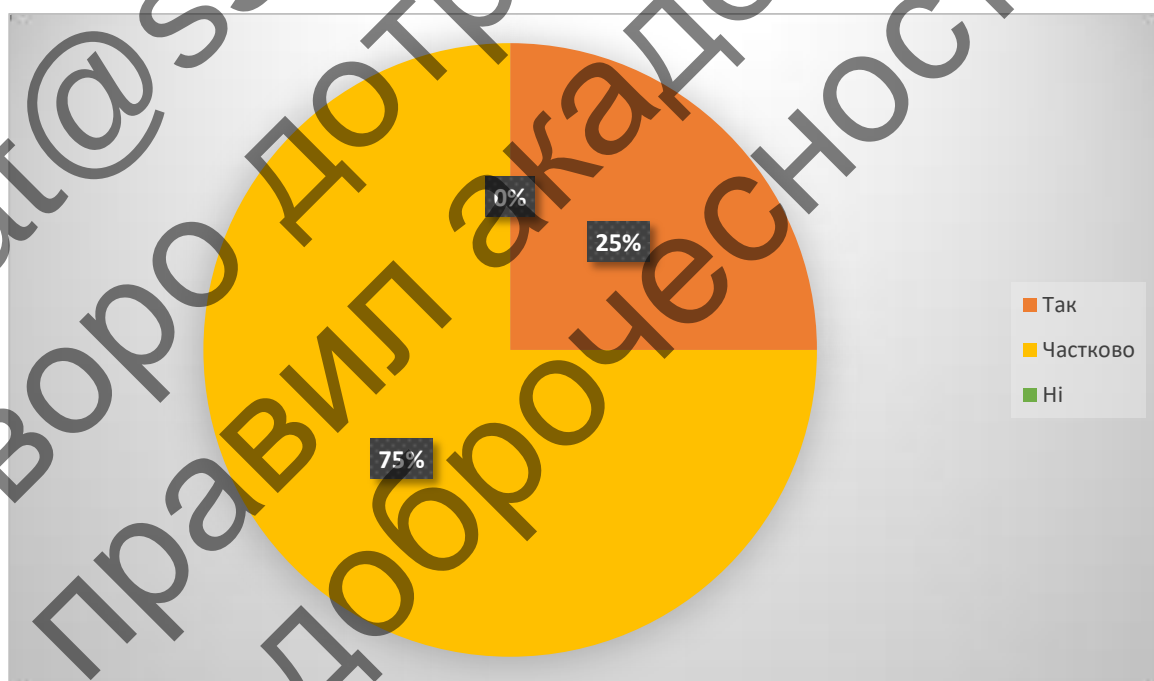


Рис. 6. Мені подобається шукати помилки і виправляти їх у своїй роботі

Вихідне анкетування

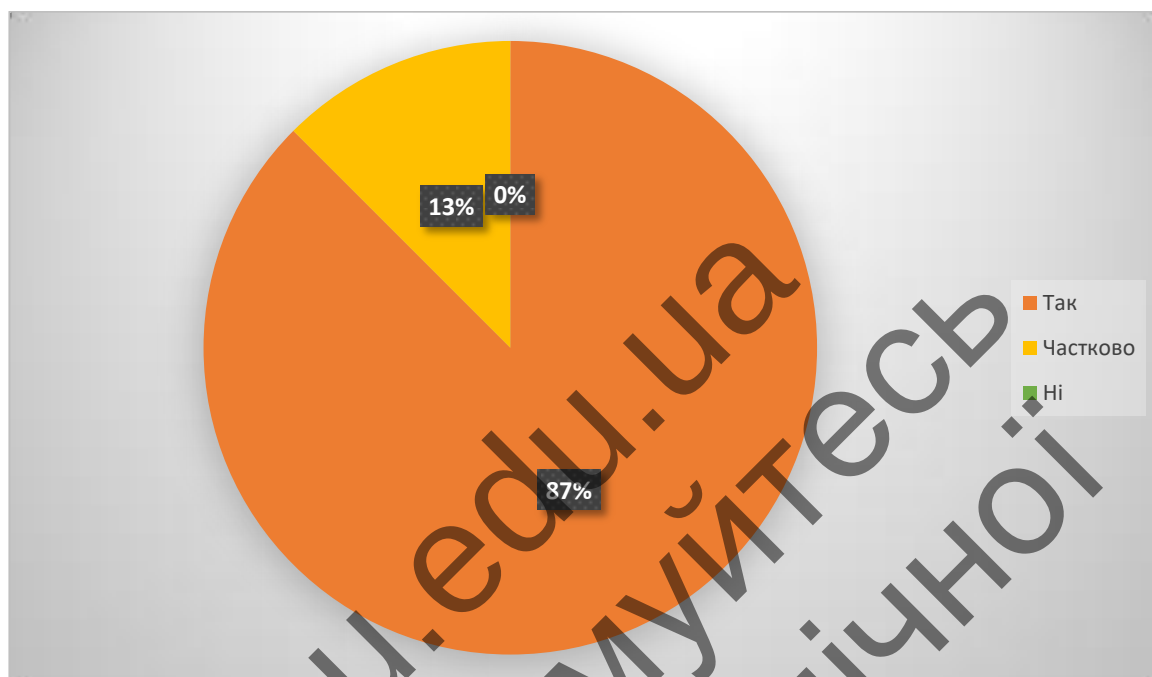


Рис. 7. Мені цікаво працювати з тривимірними об'єктами

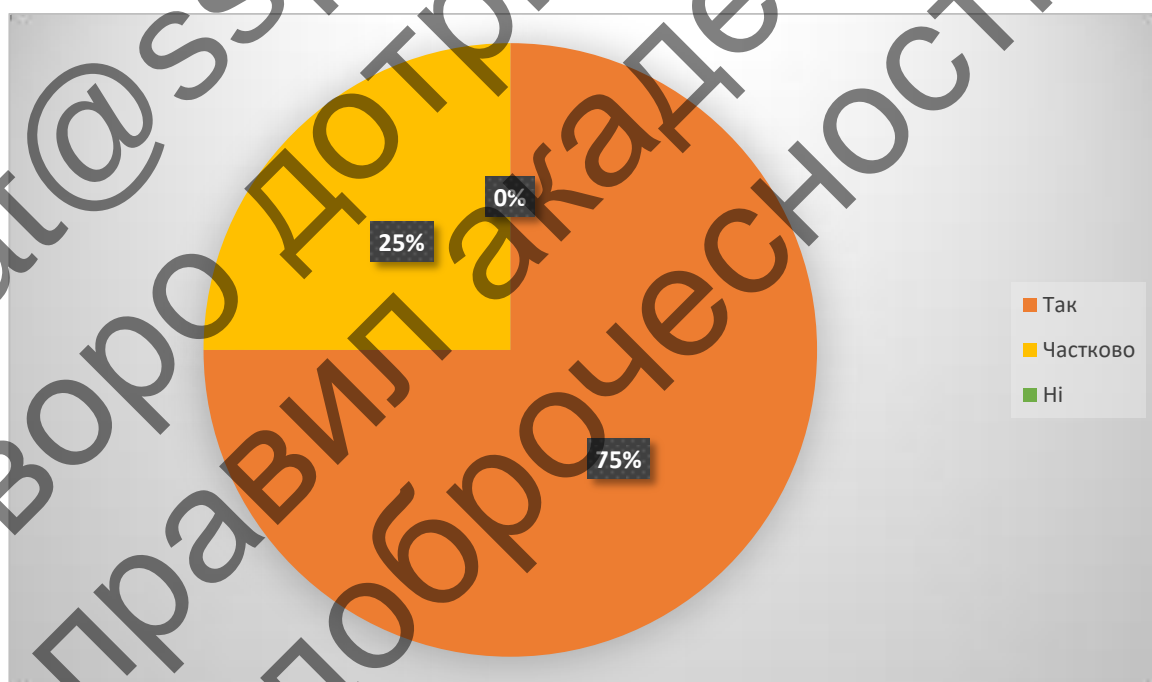


Рис. 8. Я розумію, що таке алгоритм у простих діях

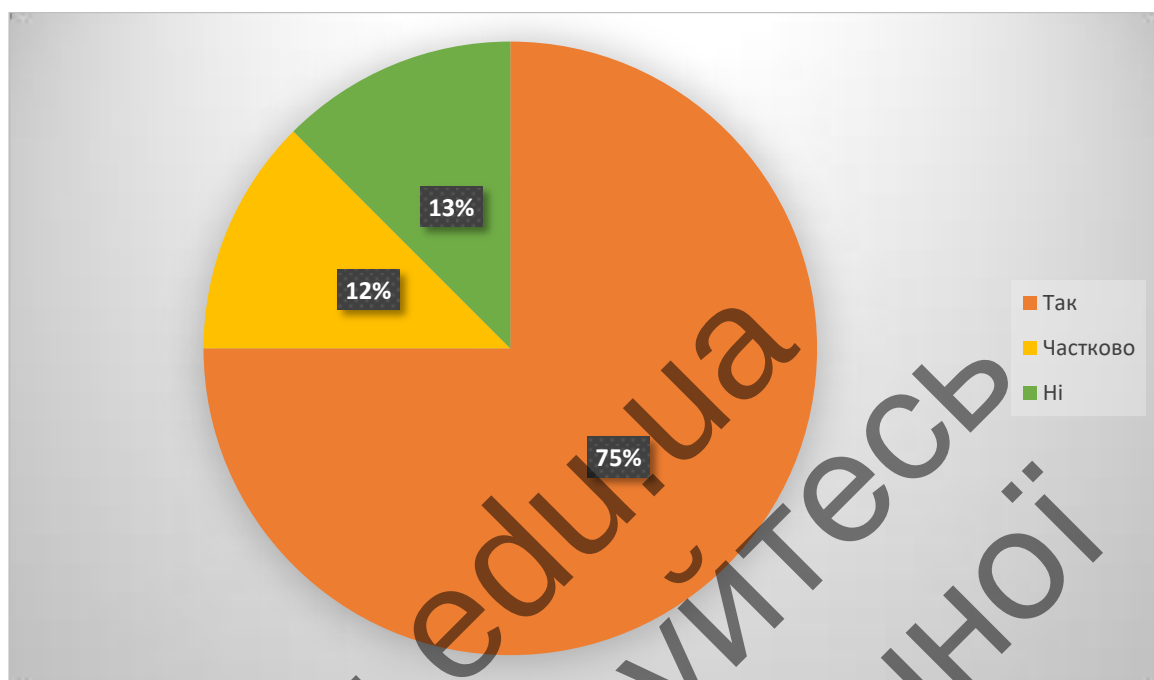


Рис. 9. Я можу описати послідовність дій для створення 3D-моделі

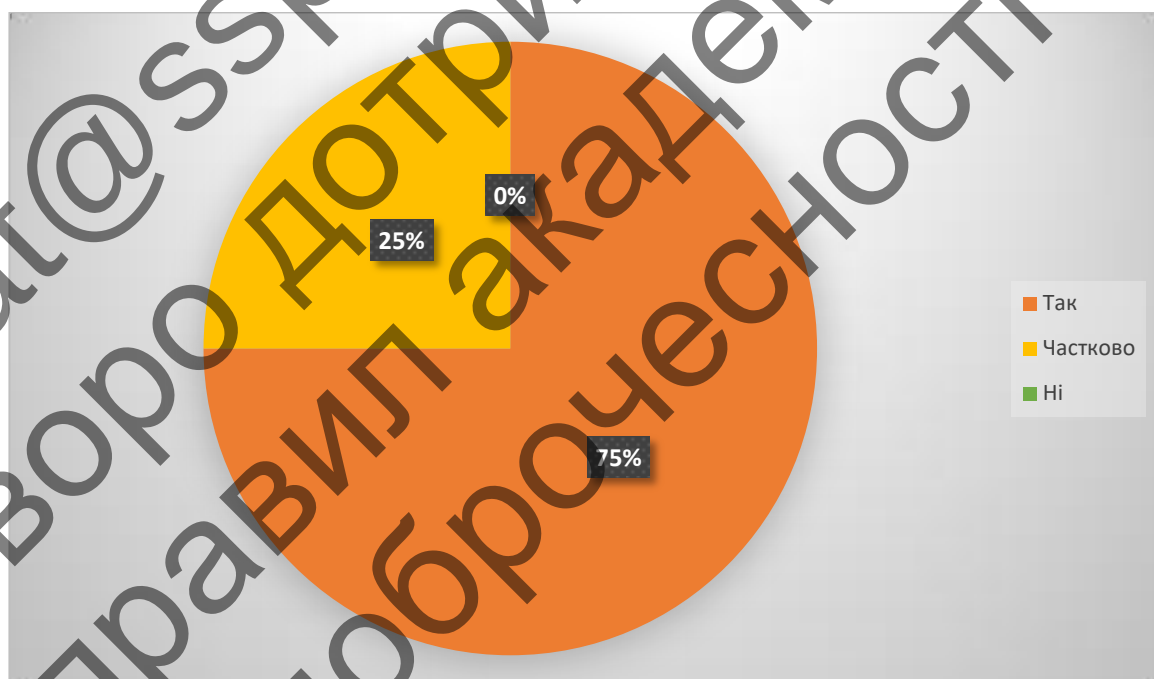


Рис. 10. Я помічаю, коли дії потрібно виконувати в певному порядку

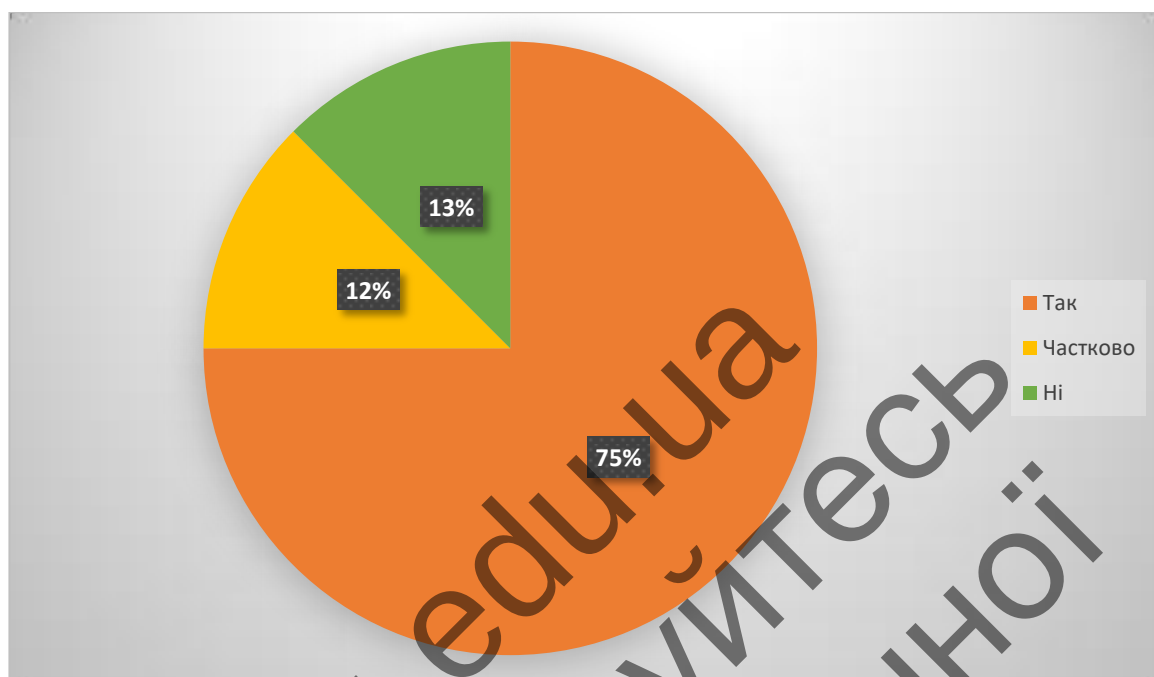


Рис. 11. Я можу пояснити, чому об'єкт поводить себе саме так (рухається, змінюється)

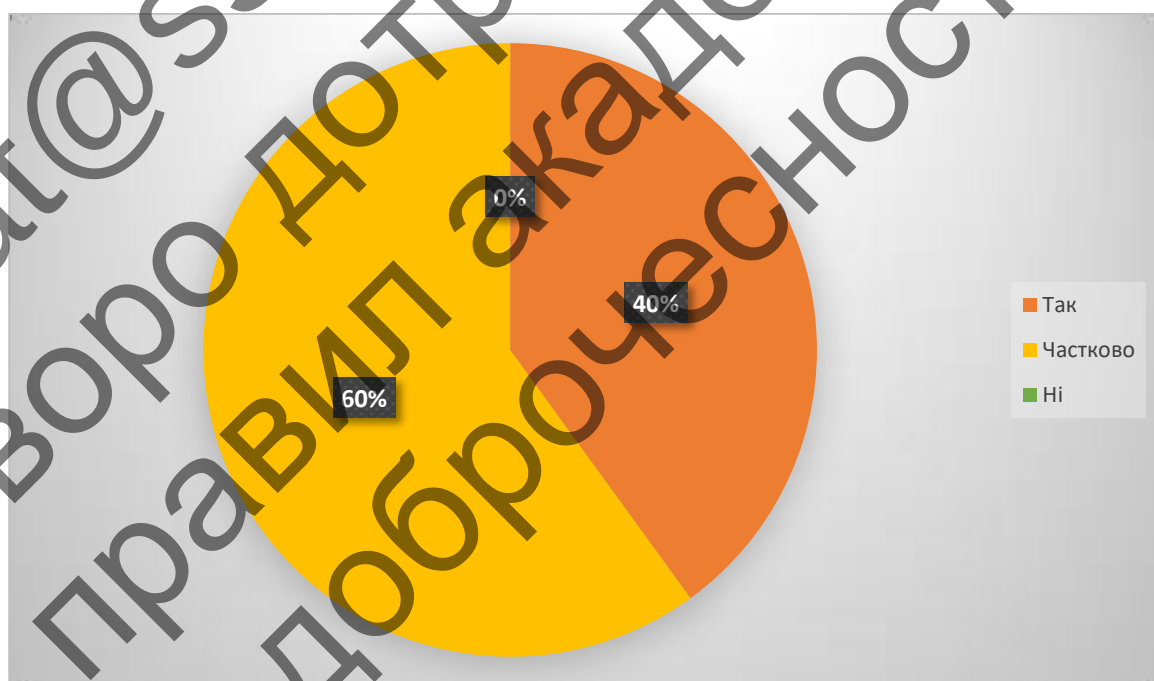


Рис. 12. Мені подобається шукати помилки і виправляти їх у своїй роботі

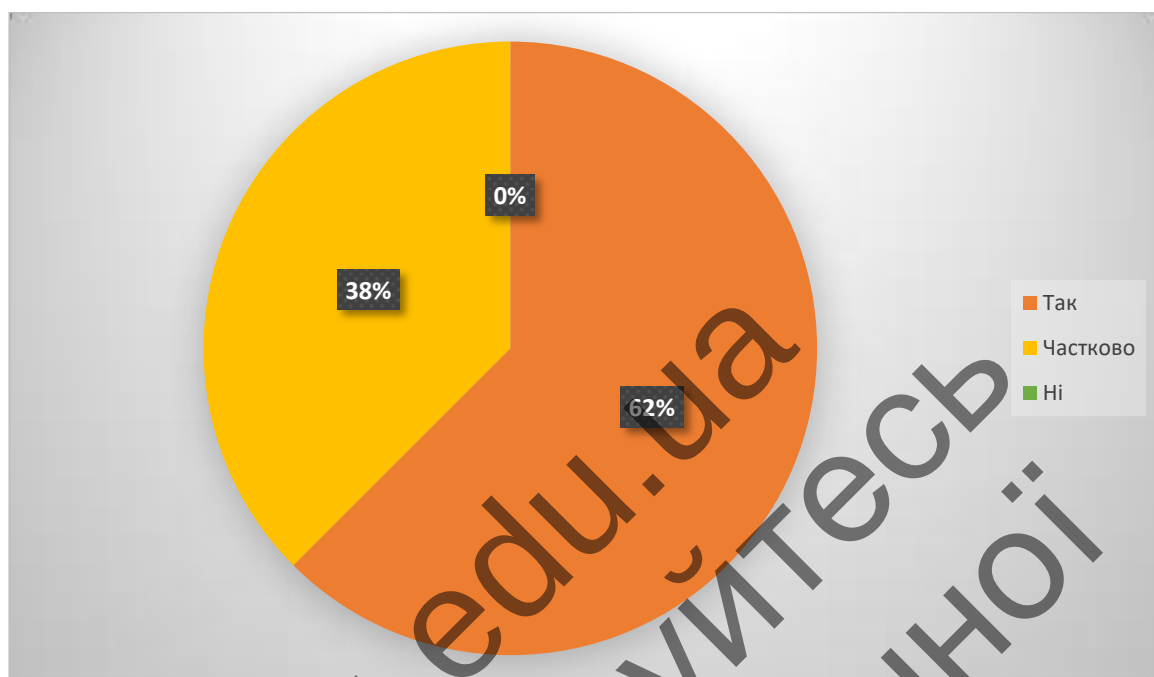


Рис. 13. Я почав(ла) краще розуміти, як працюють комп'ютерні програми

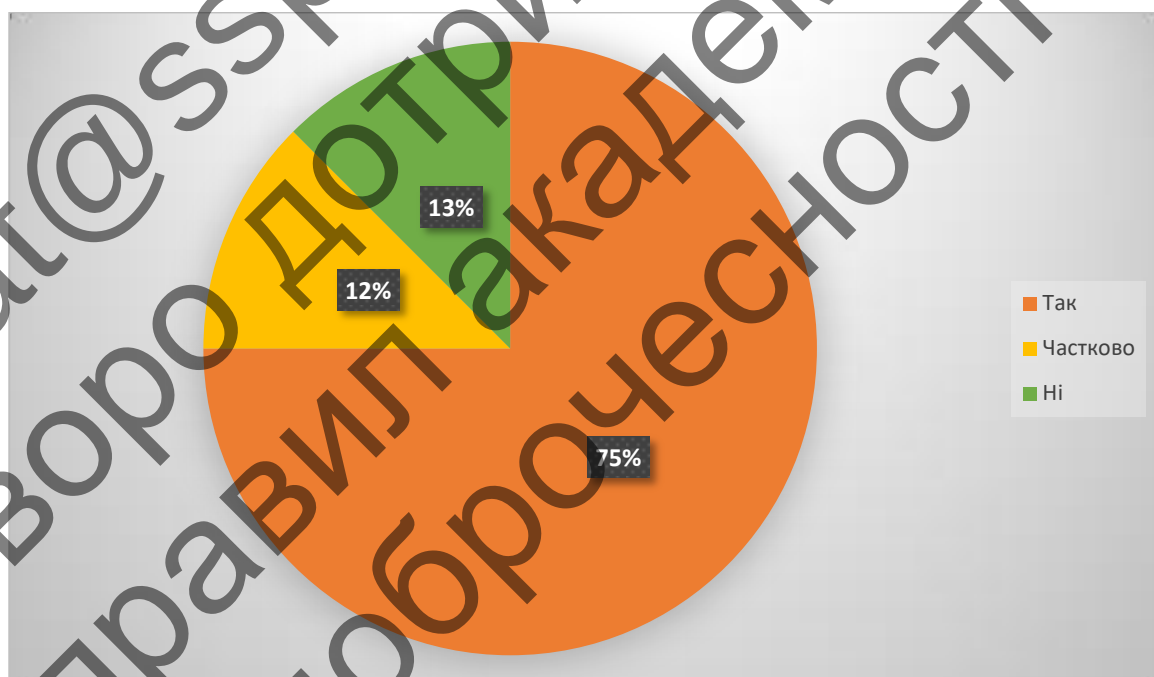


Рис. 14. Я хочу й надалі працювати з 3D-моделюванням

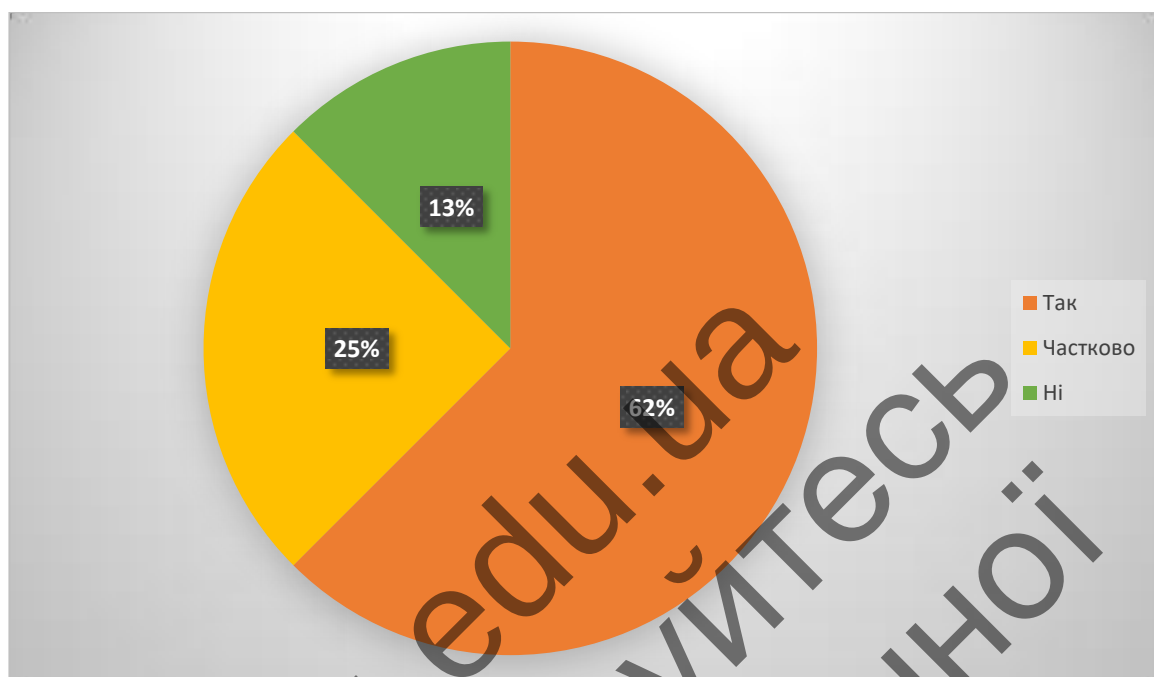


Рис. 15. Я відчуваю, що моє мислення стало більш логічним

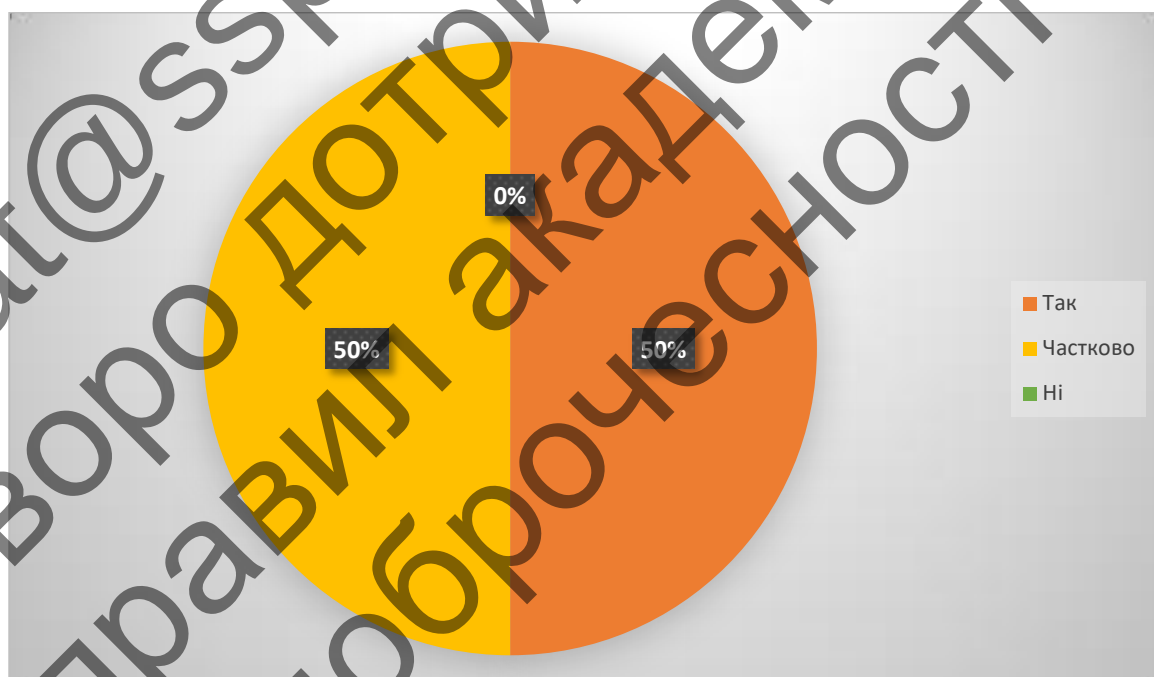


Рис. 16. Я можу самостійно спланувати просту 3D-сцену

Блок 1: Вступ. Алгоритм як послідовність дій

Завдання 1.1 Створи куб, змінюй його розміри, переміщуй у просторі.

Мета: усвідомити, що кожна дія – це крок алгоритму.

Інструкція

У далекій країні Геометрії жив Кубус – чарівний куб, який мріяв подорожувати світом. Але він не міг рушити з місця, поки хтось не відкриє його алгоритм дій. Тільки той, хто зуміє змінити його форму та перемістити у просторі, зможе пробудити його силу.

Коли ти переміщуєш Кубуса – він оживає! Уяви, що кожен твій клік – це магічне заклинання, яке відкриває нові координати його пригод.

Щоб створити куб, треба натиснути «Додати» → «Сіть» → «Куб». Куб створиться на позиції 3Д курсору. Щоб перемістити куб, треба обрати інструмент «Переміщення» і тягати цей куб за стрілочки осей або за біле коло, щоб переміщувати куб відносно камери. Так само відбувається і масштаб.

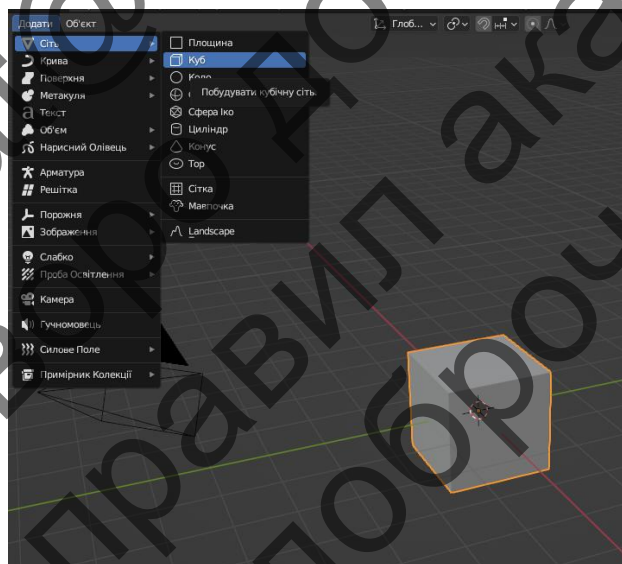


Рис. 1. Створення куба

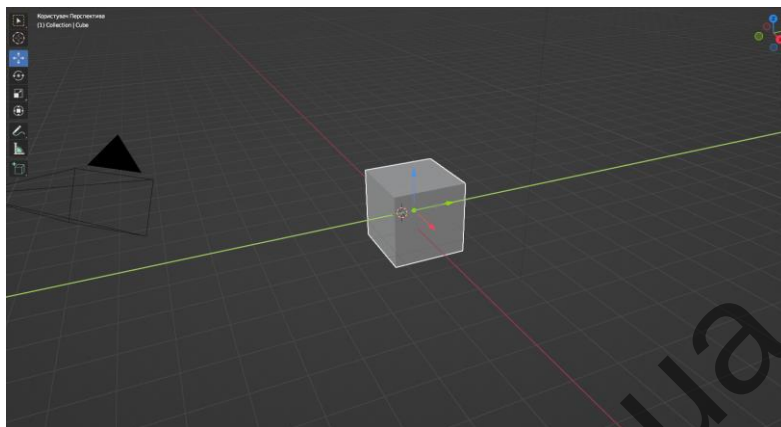


Рис. 2. Переміщення куба

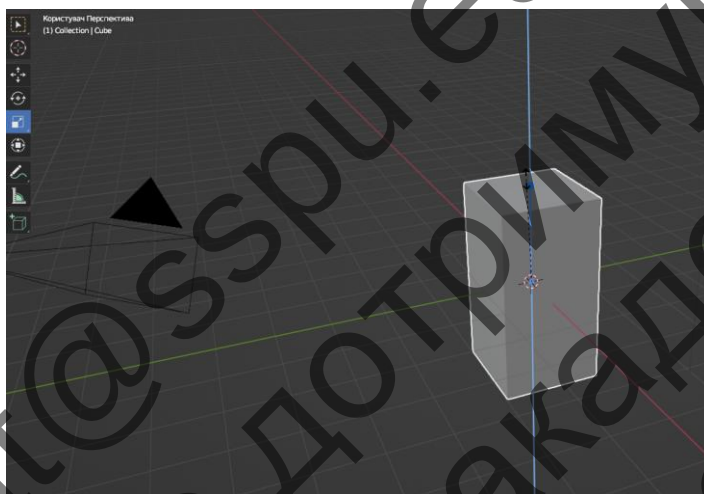


Рис. 3. Масштабування куба

Завдання 1.2 Напиши алгоритм створення сцени з двома об'єктами (на папері).

Мета: навчитися планувати дії до виконання.

У королівстві Простору живуть двоє друзів – Куб і Циліндр. Вони хочуть побудувати сцену для вистави, але не знають, з чого почати. Тільки ти можеш скласти для них план дій, щоб усе відбулося без плутанини.

Твій алгоритм – це сценарій вистави! Уяви, що ти режисер, а об'єкти – актори, які чекають твоїх вказівок.

Інструкція

Виконати дії на основі завдання 1.1.

Заповнити таблицю.

Прізвище та ім'я _____	Назва
Клас _____	сцени _____
Опис кроків алгоритму	Досягнення та складнощі
Пошук помилок та шляхи їх усунення	

Блок 2: Побудова тривимірних структур

Завдання 2.1 Створи модель будинку з примітивів (куб, циліндр, конус).

Мета: декомпозиція – розбиття складного на просте.

Інструкція

У лісі біля річки живе архітекторка Кубія. Вона мріє збудувати будинок для лісових мешканців – сови, білки та їжака. Але їй потрібна допомога з плануванням.

Кожен об'єкт – це частина чарівного будинку! Уяви, що ти будуєш притулок для фантастичних істот.

Створити будинок можна додаючи, переміщуючи, та масштабуючи об'єкти (як було в 1.1.) у формі будинку. Сам будинок буде кубом, ми додаємо ще куб для дверей та вікна. Дах треба вже редагувати в режимі редагування. В цьому режимі можна змінювати окремі вершини, ребра, та грані.



Рис. 4. Побудова будинку

Заповнити таблицю.

Прізвище та ім'я _____	Назва _____
Клас _____	сцени _____
Опис кроків алгоритму	Досягнення та складнощі
Пошук помилок та шляхи їх усунення _____	

Завдання 2.2 Застосуй модифікатор «Дзеркалення» для симетрії.

Мета: оптимізація алгоритму – уникнення дублювання.

Інструкція

У долині Симетрії живе чарівне Дзеркало, яке може подвоїти будь-який об'єкт. Але воно працює лише тоді, коли ти правильно налаштуєш його вісь.

Ти – володар дзеркального порталу! Один рух – і об'єкт має двійника.

Модифікатор «Дзеркалення» буде віддзеркалювати об'єкт відносно певних осей. Його можна додати у вікні «Властивості» → «Модифікатори» → «Додати модифікатор» → «Дзеркалення». Таким чином можна віддзеркалити половину об'єкта щоб моделювати лише його половину, зменшивши обсяг роботи.

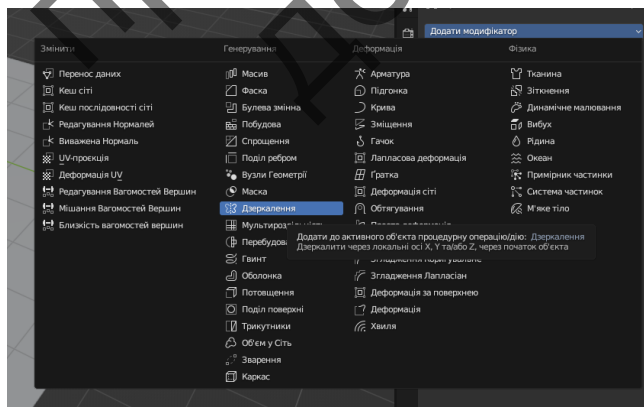


Рис. 5. Модифікатор «Дзеркалення»

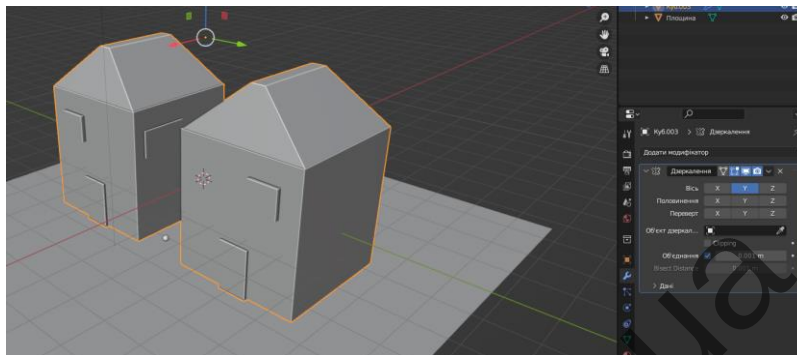


Рис. 6. Модифікатор «Дзеркалення» та готовий об'єкт
Заповнити таблицю.

Прізвище та ім'я _____	Назва _____
Клас _____	сцени _____
Опис кроків алгоритму	Досягнення та складнощі
Пошук помилок та шляхи їх усунення _____	

Блок 3: Матеріали, текстури, умови

Завдання 3.1 Створи матеріал, який змінює колір залежно від освітлення.

Мета: умовні оператори – дія залежить від параметра.

Інструкція

У місті Світлотон живе Хамелеон – об'єкт, який змінює колір залежно від сонця. Щоб він ожив, потрібно створити для нього магічний матеріал.

Твій об'єкт реагує на світло, як справжній хамелеон! Це магія фізики і кольору.

Текстури, або матеріали, можна задати у вікні «Властивості» → «Матеріали» → «Новий». Додається базовий матеріал, властивості якого можна

змінювати. Колір, металічність, та шорсткість – це найбільш важливі параметри.

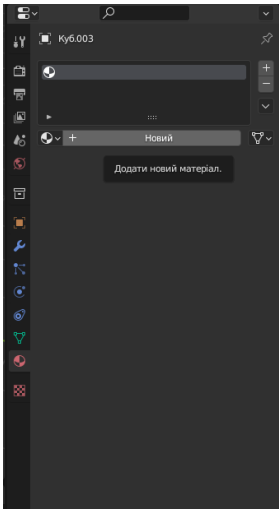


Рис. 7. Модифікатор «Матеріали»

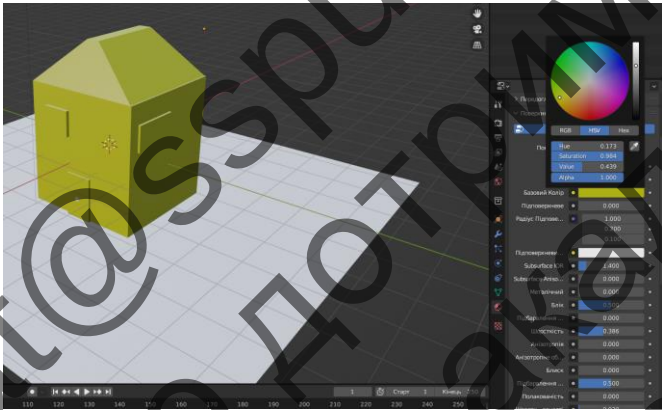


Рис. 8. Модифікатор «Матеріали» та готовий об’єкт
Заповни таблицю.

Прізвище та ім’я _____	Назва _____
Клас _____	сцени _____
Опис кроків алгоритму	Досягнення та складнощі
Пошук помилок та шляхи їх усунення _____	

Завдання 3.2 Застосуй текстуру лише до одного об’єкта.

Мета: навчитися працювати з умовами застосування.

Інструкція

У королівстві Візуалії живе Об’єкт, який хоче стати особливим. Він мріє світитися, як зірка, але тільки ти можеш подарувати йому унікальну текстуру.

Твій об’єкт пульсує світлом! Уяви, що він – маяк у темряві.

Шейдери, або відтінювачі, складаються із вузлів. Можна маніпулювати ними, коли буде відображатись шейдер, під’єднуючи вузли до його параметрів. Таким чином можна анімувати один із цих параметрів. Наприклад, «Сила емісії», щоб об’єкт пульсував світлом. Анімація відбувається відносно ключових кадрів, які ми задаємо самостійно на проміжку часу.

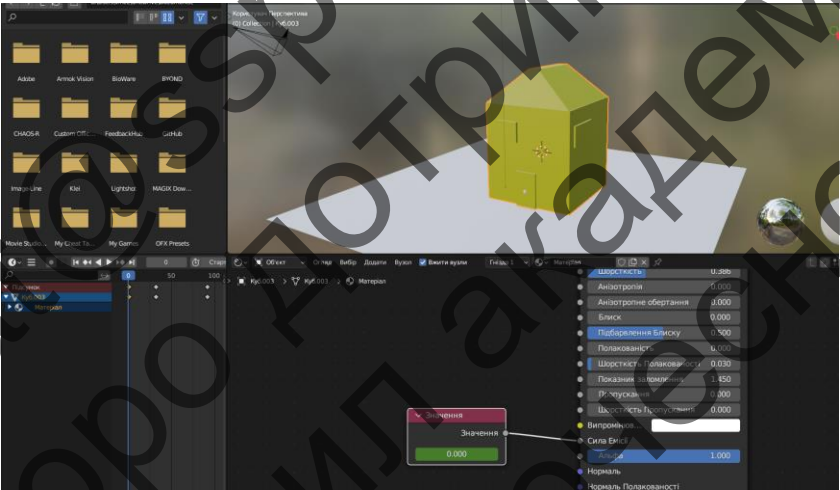


Рис. 9. Текстура та готовий об’єкт

Заповнити таблицю.

Прізвище та ім'я _____ Клас _____	Назва сцени _____ _____
Опис кроків алгоритму	Досягнення та складнощі

Пошук помилок та шляхи їх усунення _____	

Блок 4: Анімація і цикли

Завдання 4.1 Створи анімацію руху об'єкта по колу.

Мета: циклічність – повторення дій у часі.

Інструкція

У небі літає Зоряний Супутник. Його траєкторія – коло, але він загубив свій маршрут. Тільки ти можеш відновити його орбіту.

Твій об'єкт рухається, як планета навколо зірки! Це справжня космічна місія.

Потрібно зробити ключові кадри відносно уявного кола, по якому буде ковзати об'єкт. Застосувати інтерполяцію руху об'єкту між ключовими кадрами, та змінити її.

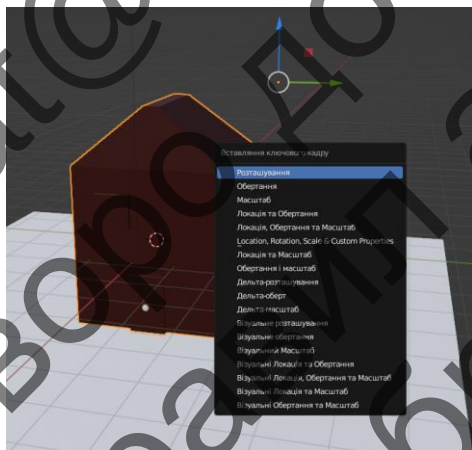


Рис. 10. Розташування об'єкту

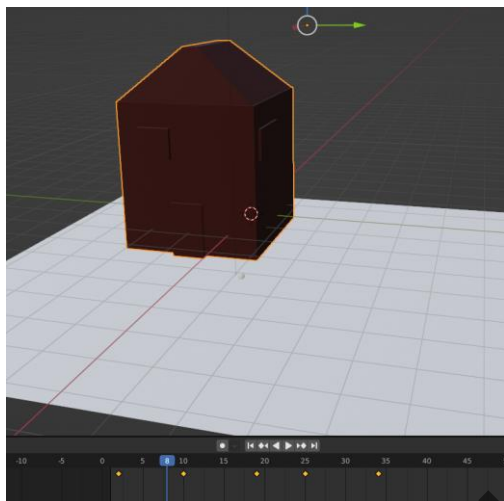


Рис. 11. Циклічність об'єкту

Заповнити таблицю.

Прізвище та ім'я _____	Назва _____
Клас _____	спени _____
Опис кроків алгоритму	Досягнення та складнощі
Пошук помилок та шляхи їх усунення	

Завдання 4.2 Створи анімацію зміни розміру об'єкту (зменшення → збільшення → зменшення).

Мета: логіка циклів, ключові кадри.

Інструкція

У чарівному саду росте Квітка Ритму. Вона розкривається і стискається в такт музики часу. Щоб її оживити, потрібно задати правильний цикл.

Об'єкт дихає, як жива істота! Ти керуєш його ритмом.

Використаємо редактор графів, у якому візуально видно зміну параметрів об'єкта при його анімації (наприклад, розташування по 3-м осям). Можна застосувати ці модифікатори щоб об'єкт змінювався нескінченно за заданими

параметрами. У цьому випадку, розмір об'єкту по осі Z буде збільшуватись та зменшуватись відносно часу.

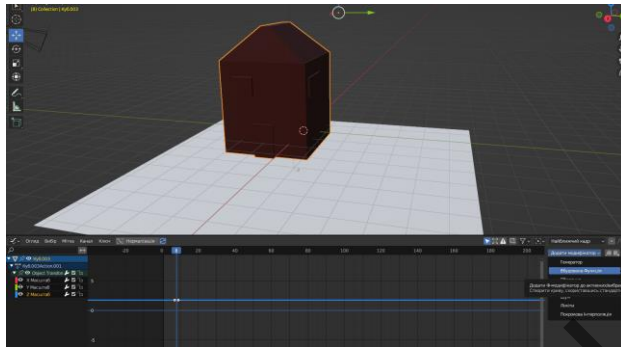


Рис. 12. Використання редактора графів

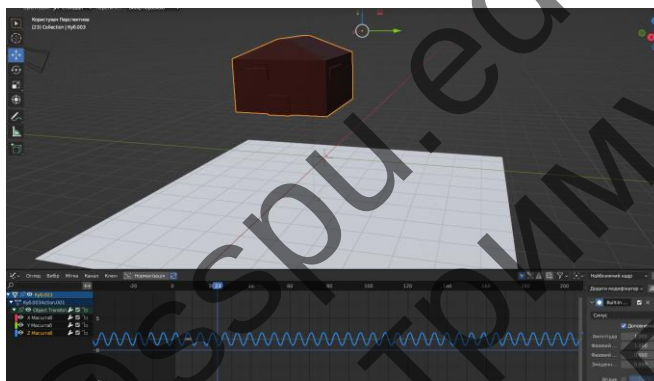


Рис. 13. Збільшення та зменшення об'єкту відносно часу

Заповнити таблицю.

Прізвище та ім'я _____	Назва сцени _____
Клас _____	_____
Опис кроків алгоритму	Досягнення та складнощі
_____	_____
_____	_____
_____	_____
_____	_____
Пошук помилок та шляхи їх усунення _____	

Блок 5: Проєкт і рефлексія

Завдання 5.1 Створи власну сцену (наприклад, «Місто майбутнього»).

Мета: самостійне планування алгоритму.

Завдання 5.2 Опиши свій алгоритм: які кроки, чому саме так, що вдалося.

Мета: рефлексія, усвідомлення алгоритмічного мислення.

Інструкція

Ти – архітектор майбутнього. Перед тобою – порожній простір, а в уяві – місто, де літають автівки, ростуть енергодерева і світяться дороги.

Твоя сцена – це портал у майбутнє! Усе, що ти створиш, може стати реальністю.

Опиши алгоритм дій та виконай завдання самостійно.

Заповни таблицю

Прізвище та ім'я _____ Клас _____	Назва сцени _____
Опис кроків алгоритму	Досягнення та складнощі
Пошук помилок та шляхи їх усунення _____	