

А.С. Литвинов

ВИКОРИСТАННЯ СЕРЕДОВИЩА SCRATCH У ПРОЦЕСІ НАВЧАННЯ ІНФОРМАТИЧНОЇ ОСВІТНЬОЇ ГАЛУЗІ



Міністерство освіти і науки України
Глухівський національний педагогічний університет
імені Олександра Довженка

А. Литвинов

**«ВИКОРИСТАННЯ СЕРЕДОВИЩА SCRATCH
У ПРОЦЕСІ НАВЧАННЯ
ІНФОРМАТИЧНОЇ ОСВІТНЬОЇ ГАЛУЗІ»**

Навчально-методичний посібник

Глухів – 2026

УДК 373.3:004:37.016

Л 64

Рекомендовано Вченою радою:

*Глухівського національного педагогічного університету
імені Олександра Довженка протокол № 8 від 25.02.2026*

Рецензенти:

Борисов В.В. – доктор педагогічних наук, професор, заслужений працівник освіти України, професор кафедри педагогіки та методик навчання Комунального закладу вищої освіти Запорізької обласної ради «Хортицька національна навчально-реабілітаційна академія»

Бурчак С.О. – доктор педагогічних наук, професор, завідувач кафедри професійної освіти та комп'ютерних технологій Глухівського національного педагогічного університету імені Олександра Довженка

Литвинов А.С.

Л 64 Використання середовища Scratch у процесі навчання інформатичної освітньої галузі: навчально-методичний посібник. Глухів, 2026. 67 с.

ISBN 978-966-376-138-1

Навчально-методичний посібник присвячено методичним аспектам використання середовища візуального програмування Scratch у процесі навчання інформатичної освітньої галузі. У посібнику розкрито можливості Scratch як цифрового навчального засобу для формування алгоритмічного мислення, розвитку логіки, умінь планувати та реалізовувати послідовність дій у цифровому середовищі. Практикум спрямовано на формування фахових компетентностей майбутніх учителів щодо створення, аналізу та методично обґрунтованого використання навчальних програм і проєктів у середовищі Scratch. Навчально-методичний посібник може бути використаний науково-педагогічними (педагогічними працівниками) під час підготовки до занять та організації самостійної роботи здобувачів. Розраховано на викладачів та студентів закладів вищої освіти, а також на всіх, хто цікавиться середовищем візуального програмування Scratch.

УДК 373.3:004:37.016

ISBN 978-966-376-138-1

© Литвинов А.С.
©РВВ Глухівського НПУ ім.О.Довженка

ЗМІСТ

ВСТУП	5
РОЗДІЛ I. ВІЗУАЛЬНЕ ПРОГРАМУВАННЯ SCRATCH	7
1.1. Методичні засади використання середовища Scratch у практиці навчання програмування	7
1.2. Особливості програмування в середовищі Scratch	11
1.3. Методичні передумови навчання програмування в середовищі Scratch	12
РОЗДІЛ II. ЛАБОРАТОРНИЙ ПРАКТИКУМ	15
■ ЛАБОРАТОРНА РОБОТА №1 Знайомство з середовищем Scratch. Створення простої анімації руху спрайта	15
■ ЛАБОРАТОРНА РОБОТА №2 Створення анімації спрайта шляхом послідовної зміни його образів у середовищі Scratch	18
■ ЛАБОРАТОРНА РОБОТА №3 Додавання нових сцен до проєкту. Зміна сцени	22
■ ЛАБОРАТОРНА РОБОТА №4 Додавання нового спрайта до проєкту. Створення анімації для нього	26
■ ЛАБОРАТОРНА РОБОТА №5 Додавання звукових ефектів до проєкту	29
■ ЛАБОРАТОРНА РОБОТА №6 Створення складної анімації в Scratch за участю одного спрайта	35
■ ЛАБОРАТОРНА РОБОТА №7 Створення складної анімації з кількома спрайтами	39
■ ЛАБОРАТОРНА РОБОТА №8 Створення та виконання алгоритмів з повторенням. Малювання геометричних фігур за допомогою олівця	44
■ ЛАБОРАТОРНА РОБОТА №9 Створення та виконання алгоритмів з розгалуженням	49
■ ЛАБОРАТОРНА РОБОТА №10 Створення навчального мініпроєкту в Scratch	53
РОЗДІЛ III. ЗРАЗКИ ВПРАВ ДЛЯ ПОГЛИБЛЕНОГО ВИВЧЕННЯ	56
Вправа 1. «Вкладені кола»	56
Вправа 2. «Симетричні вкладені кола»	57
Вправа 3. «24 квадрати»	58
Вправа 4. «Паркан»	59
Вправа 5. «Райдужна зірка»	60
Вправа 6. «Акваріум»	61
Вправа 7. «Полювання на гриби»	62
Вправа 8. «Геометричні фігури»	62
Вправа 9. «Скільки буде 2+3?»	63
Вправа 10. «Таблиця множення»	64
Вправа 11. «Вгадай число»	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	68

ВСТУП

Цифрова трансформація освіти в умовах Нової української школи актуалізує потребу у формуванні в здобувачів освіти алгоритмічного мислення, здатності до логічного аналізу, планування діяльності та усвідомленого використання цифрових інструментів. У цьому контексті особливого значення набуває впровадження таких програмних середовищ, які поєднують доступність, наочність і широкі можливості для організації діяльнісного навчання. Одним із таких середовищ є Scratch, що розглядається як ефективний інструмент візуального програмування та розвитку інформатичної компетентності.

Scratch орієнтований на поетапне формування уявлень про алгоритми, структуру програм і логіку виконання команд без перевантаження користувача складними синтаксичними конструкціями. Візуальний характер середовища дозволяє зосередитися на змісті діяльності, причинно-наслідкових зв'язках між командами та результатами їх виконання, що відповідає принципам компетентнісного та діяльнісного підходів, задекларованих у концепції Нової української школи. Саме тому Scratch широко використовується в освітній практиці як засіб формування базових алгоритмічних уявлень і розвитку цифрової грамотності.

Ефективне використання Scratch у навчальному процесі потребує технічного володіння середовищем, методично обґрунтованого підходу до добору завдань, організації навчальної діяльності та аналізу освітніх результатів. Майбутній учитель має розуміти, які навчальні дії виконує здобувач освіти під час роботи в середовищі Scratch, які когнітивні процеси при цьому активізуються та яким чином програмні завдання можуть бути інтегровані в структуру навчального заняття відповідно до поставлених цілей.

Навчально-методичний посібник «Використання середовища Scratch в інформатичній освітній галузі» розроблено як складову системи практичної підготовки здобувачів вищої освіти педагогічних спеціальностей у межах дисциплін, пов'язаних з методикою навчання інформатичної освітньої галузі. Його зміст спрямований на формування у студентів здатності створювати, аналізувати та методично оцінювати навчальні програми і проекти в середовищі Scratch, орієнтовані на реалізацію діяльнісного та компетентнісного підходів.

На відміну від традиційних мов програмування, Scratch дозволяє організувати навчальну діяльність через конструювання програм із готових блоків, що знижує когнітивне навантаження та сприяє зосередженню на логіці алгоритму. Робота в середовищі Scratch забезпечує формування умінь послідовно виконувати дії, використовувати повторення й розгалуження, планувати взаємодію об'єктів, аналізувати помилки та коригувати власні рішення. Такі уміння є фундаментальними для подальшого опанування інформатики та інших галузей знань.

Основною метою навчально-методичного посібника є формування методичної готовності майбутніх учителів до використання Scratch як освітнього інструменту. Практикум орієнтований на усвідомлене проектування навчальних завдань і проєктів, що забезпечують активну участь здобувачів освіти, розвиток алгоритмічного мислення та цифрової культури. У процесі виконання лабораторних робіт студенти аналізують результат роботи програми, логіку її побудови, можливі труднощі та педагогічні способи їх подолання.

Зміст лабораторного практикуму побудовано за принципом поступового ускладнення. Від опанування базових елементів інтерфейсу та простих алгоритмів студенти переходять до створення анімацій, роботи з кількома об'єктами, використання повторення, розгалуження, подій, сцен, звуків і графічних ефектів. Завершальним етапом є створення цілісного навчального мініпроєкту, у якому інтегруються всі раніше засвоєні елементи. Усі завдання орієнтовані на реальні освітні ситуації та моделюють діяльність учителя, який планує, організовує та аналізує використання середовища Scratch у навчальному процесі. Такий підхід сприяє формуванню професійної рефлексії, здатності аргументовано обґрунтовувати методичні рішення та усвідомлено використовувати цифрові технології відповідно до цінностей Нової української школи.

Матеріали навчально-методичного посібника можуть використовуватися в аудиторній роботі та під час самостійної підготовки студентів. Вони створюють методичну основу для подальшого розвитку професійної компетентності майбутніх учителів у сфері використання цифрових освітніх середовищ і є логічним етапом підготовки до більш складних форм цифрової освітньої діяльності.

РОЗДІЛ I. ВІЗУАЛЬНЕ ПРОГРАМУВАННЯ SCRATCH

1.1. Методичні засади використання середовища Scratch у практиці навчання програмування

Сучасне освітнє середовище дедалі більше орієнтується на формування у здобувачів освіти здатності діяти в цифровому просторі усвідомлено, цілеспрямовано й результативно. У цьому контексті програмування розглядається як технічна навичка, як засіб розвитку логічного мислення, умінь планувати діяльність, аналізувати ситуацію та прогнозувати результат. Візуальні середовища програмування, зокрема Scratch, створюють умови для реалізації таких освітніх завдань завдяки поєднанню наочності, доступності та діяльнісного характеру навчання.

Методична доцільність використання Scratch зумовлена передусім його структурною простотою та водночас достатньою функціональною насиченістю. Середовище дозволяє організувати навчальну діяльність таким чином, щоб основна увага була зосереджена на логіці побудови алгоритмів, зв'язку між окремими командами та результатами їх виконання. Це відповідає сучасним підходам до навчання інформатики, орієнтованим на розвиток мислення.

Організація навчання програмування в середовищі Scratch має будуватися за принципом поступового входження в діяльність. Початкові етапи роботи доцільно спрямовувати на формування загального уявлення про програму як упорядковану послідовність дій. У цьому контексті особливого значення набуває використання життєвих аналогій, які допомагають осмислити нові поняття без перевантаження абстрактними визначеннями. Алгоритм у Scratch постає як інструкція, що складається з окремих логічно пов'язаних кроків, кожен з яких має конкретний результат на сцені.

Важливим методичним кроком є ознайомлення зі структурою середовища Scratch. Розподіл робочого простору на функціональні зони – сцену, панель блоків, область складання скриптів, список спрайтів і вкладки редагування – сприяє формуванню системного бачення програми. Така організація дозволяє чітко відокремити об'єкти від дій, а дії – від результату, що є принципово важливим для розвитку алгоритмічного мислення. Усвідомлення ролі кожної зони зменшує хаотичність дій і підвищує рівень самостійності під час виконання завдань.

Перші практичні завдання в Scratch доцільно пов'язувати зі створенням простих програм, у яких реалізується базовий рух об'єкта. Саме такі завдання дозволяють сформувати первинні вміння складання скриптів, продемонструвати послідовність виконання команд і наочно показати причинно-наслідкові зв'язки між діями та результатом. Важливо, що навіть найпростіші програми в Scratch можуть бути візуально привабливими, що підсилює мотивацію та створює позитивне ставлення до програмування.

На цьому етапі особливої уваги потребує питання запуску програм. Scratch надає можливість ініціювати виконання скриптів різними способами, що дозволяє поступово підводити до розуміння подієвого характеру програмування. Усвідомлення того, що одна програма може містити кілька незалежних або взаємопов'язаних скриптів, створює основу для подальшої роботи з взаємодією об'єктів і складнішими сценаріями.

Подальший розвиток практичних умінь передбачає поступове введення базових понять, зокрема «скрипт» і «програма». Методично доцільним є співвіднесення цих понять з уже відомим поняттям алгоритму. Такий підхід дозволяє уникнути формального заучування визначень і сприяє глибшому розумінню сутності програмування як процесу організації діяльності. У Scratch алгоритм не залишається абстрактною схемою, а відразу матеріалізується у вигляді конкретних дій на сцені.

Робота з окремими групами блоків має здійснюватися послідовно й логічно. Блоки руху та зовнішнього вигляду дозволяють створювати прості анімації, які поєднують зміну положення об'єкта з трансформацією його зовнішніх характеристик. Такі завдання сприяють формуванню уявлень про часову послідовність дій, ритм і синхронізацію, що є важливими компонентами алгоритмічного мислення. Водночас вони створюють простір для творчості й експериментування, що підвищує пізнавальний інтерес.

Подальше розширення навчальної діяльності в середовищі Scratch доцільно пов'язувати з поступовим ускладненням програм і введенням нових способів впливу на поведінку об'єктів. На цьому етапі важливого значення набуває робота з блоками зовнішнього вигляду та звуку, оскільки саме вони дозволяють перетворити формально правильну програму на цілісний мультимедійний продукт. Зміна костюмів, використання пауз, додавання звукових ефектів сприяють формуванню уявлень про анімацію як результат послідовного виконання дій у часі.

Методично доцільно розглядати анімацію не як самоціль, а як засіб візуалізації алгоритму. Завдання, у яких поєднуються рух об'єкта, зміна його зовнішнього вигляду та відтворення звуків, дозволяють простежити, як окремі команди взаємодіють між собою та впливають на кінцевий результат. У процесі виконання таких завдань формується розуміння того, що навіть незначна зміна в алгоритмі може суттєво змінити поведінку програми.

Важливим компонентом навчальної діяльності є робота зі спрайтами, зокрема використання бібліотеки об'єктів та створення власних графічних елементів. Інтегрований графічний редактор Scratch надає можливість обирати готові зображення та редагувати їх відповідно до задуму програми. Така діяльність сприяє розвитку просторових уявлень, уважності до деталей і навичок цифрової творчості. Водночас вона формує усвідомлення того, що зовнішній вигляд об'єкта є частиною загальної логіки програми й має відповідати її змісту.

Методичну цінність має повернення до раніше створених програм з метою їх удосконалення. Заміна спрайтів, додавання нових костюмів або

звуків, коригування руху дозволяють реалізувати ідею поетапного розвитку проекту. Такий підхід сприяє формуванню рефлексивних умінь, оскільки здобувачі освіти аналізують власні попередні рішення, виявляють недоліки та пропонують способи їх усунення. Повторне звернення до вже знайомих конструкцій у новому контексті поглиблює розуміння їх функціонального призначення.

Наступним важливим етапом є ознайомлення з розширеними середовища Scratch, зокрема з інструментами малювання. Використання олівця відкриває можливості для реалізації алгоритмів, пов'язаних із побудовою геометричних фігур, орнаментів і ламаних ліній. У таких завданнях наочно проявляється зв'язок між програмуванням і математичними уявленнями про форму, напрям, кут і довжину. Малювання за допомогою алгоритмів дозволяє продемонструвати, як повторення однотипних дій приводить до створення складніших структур.

Робота з малюванням у Scratch створює сприятливі умови для введення поняття циклу. Багаторазове виконання однакових команд у межах одного алгоритму наочно ілюструє сутність повторення як фундаментальної алгоритмічної конструкції. Методично важливо, що здобувачі освіти спочатку спостерігають результат роботи циклу, а вже потім осмислюють його логіку. Такий порядок сприяє глибшому розумінню, оскільки абстрактне поняття спирається на конкретний візуальний досвід.

Поступово завдання з використанням циклів можуть ускладнюватися за рахунок зміни параметрів, комбінування різних команд і створення орнаментів або замкнених фігур. У процесі такої роботи формується вміння прогнозувати результат виконання алгоритму, що є важливою складовою алгоритмічного мислення. Крім того, подібні завдання створюють умови для міжпредметних зв'язків, зокрема з математикою та образотворчим мистецтвом.

Завдання, пов'язані з малюванням і анімацією, доцільно поєднувати з розвитком умінь самостійного експериментування. Можливість змінювати параметри команд і спостерігати результат у режимі реального часу стимулює пізнавальну активність і сприяє формуванню дослідницького підходу до навчання. У такій діяльності помилка розглядається не як негативний результат, а як джерело нового досвіду й підґрунтя для подальшого аналізу.

Алгоритми з повторенням посідають центральне місце в структурі навчання програмування в середовищі Scratch, оскільки саме вони дозволяють перейти від поодиноких дій до організованих, системних алгоритмів. Повторення в Scratch реалізується у наочній формі, що забезпечує інтуїтивне розуміння сутності циклу як багаторазового виконання певної послідовності команд. Така наочність створює сприятливі умови для формування стійких алгоритмічних уявлень і знижує складність сприйняття абстрактних понять.

Методично доцільно вводити цикли через практичні завдання, у яких повторення є природною потребою, а не формальною вимогою. Зокрема, малювання геометричних фігур і орнаментів дозволяє продемонструвати, як багаторазове виконання однакових дій приводить до створення завершеної структури. У процесі такої роботи здобувачі освіти поступово усвідомлюють залежність між кількістю повторень, кутом повороту та формою отриманого зображення, що сприяє розвитку логічного й просторового мислення.

Особливу методичну цінність мають завдання, у яких використовується замкнена ламана лінія або орнамент із повторюваних елементів. Вони дозволяють закріпити поняття циклу та сформувати уявлення про закономірність і симетрію. Візуальний результат виконання алгоритму створює умови для самоконтролю: здобувач освіти може одразу оцінити правильність побудови алгоритму, порівняти очікуваний і фактичний результат та внести необхідні корективи.

Використання циклів у Scratch сприяє також формуванню вміння планувати алгоритм до початку його реалізації. Перед створенням програми необхідно визначити, які дії будуть повторюватися, скільки разів і в якій послідовності. Така попередня аналітична діяльність є важливим кроком до розвитку свідомого програмування та переходу від інтуїтивних дій до обґрунтованих рішень. З методичної точки зору це дозволяє сформувати уявлення про програмування як технічну операцію та процес мислення.

Наступним етапом ускладнення алгоритмічної діяльності є поєднання циклів з іншими конструкціями, зокрема з умовами та подіями. Проте ще до введення розгалужень доцільно забезпечити достатній рівень сформованості вмінь роботи з повторенням. Саме цикли створюють основу для розуміння більш складних алгоритмічних структур, оскільки вимагають усвідомлення тривалості процесу, контролю над його перебігом і прогнозування результату.

Робота з геометричними фігурами в Scratch має також виразний міжпредметний потенціал. Під час створення алгоритмів малювання активізуються знання з математики, зокрема уявлення про кути, сторони, координати та напрям руху. Таке поєднання сприяє формуванню цілісного бачення навчального матеріалу й демонструє практичну цінність програмування як інструменту моделювання математичних об'єктів. Водночас виконання творчих завдань з елементами орнаменту створює зв'язок з образотворчим мистецтвом, розвиваючи естетичне сприйняття та креативність.

Методично важливою є організація навчальної діяльності таким чином, щоб здобувачі освіти мали можливість експериментувати з параметрами алгоритмів. Зміна кількості повторень, довжини кроку або кута повороту дозволяє дослідити вплив кожного параметра на результат. Такий підхід сприяє формуванню дослідницьких умінь і підсилює пізнавальну мотивацію, оскільки навчання набуває характеру відкриття закономірностей.

У процесі виконання завдань із використанням циклів доцільно звертати увагу на необхідність очищення сцени перед початком нового малювання. Усвідомлення того, що програмний результат залежить від алгоритму та стану середовища, формує уявлення про важливість підготовчих дій у програмуванні. Такі, на перший погляд, технічні моменти мають методичну цінність, оскільки сприяють розвитку уважності та відповідальності за результат власної діяльності.

Отже, алгоритми з повторенням у середовищі Scratch виконують роль методичного ядра, навколо якого вибудовується подальше навчання програмування. Вони забезпечують перехід від простих дій до складніших алгоритмічних конструкцій, створюють умови для міжпредметної інтеграції та формують основу для подальшого опанування умов, змінних і проєктної діяльності.

1.2. Особливості програмування в середовищі Scratch

Scratch належить до середовищ візуального програмування, які орієнтовані на формування алгоритмічного мислення через безпосередню діяльність користувача. Його концепція ґрунтується на ідеї конструювання програм із готових функціональних елементів, що дозволяє змінити акцент уваги з формальних правил запису команд, а на логіку побудови алгоритму та взаємозв'язку між окремими діями. Такий підхід створює сприятливі умови для осмисленого входження у програмування й поступового ускладнення навчальної діяльності.

Програмування в середовищі Scratch реалізується у блочній формі, де кожна команда має чітко визначене функціональне призначення та візуальне представлення. Блоки поєднуються між собою за принципом сумісності, що унеможливує створення синтаксично некоректних конструкцій. Це знижує кількість технічних помилок і дозволяє зосередитися на змістовій стороні програмування. Візуальна форма блоків водночас виконує дидактичну функцію, оскільки допомагає розрізнити типи команд, умови, числові значення та логічні вирази.

Однією з важливих особливостей Scratch є можливість змінювати програму безпосередньо під час її виконання. Така організація роботи сприяє експериментуванню та дослідницькому підходу до навчання. Користувач може спостерігати результат кожної зміни в режимі реального часу, аналізувати наслідки власних рішень і коригувати алгоритм. У методичному аспекті це створює умови для формування рефлексивних умінь і розвитку здатності аналізувати причинно-наслідкові зв'язки між командами та результатом їх виконання.

Програми, створені в Scratch, зазвичай мають об'єктно-орієнтований характер. Кожен спрайт розглядається як окремий об'єкт із власними властивостями, зовнішнім виглядом і набором скриптів. Взаємодія між кількома об'єктами відбувається через події, повідомлення або сенсорні

умови, що дозволяє моделювати складні сценарії поведінки. Така організація програм сприяє формуванню уявлень про паралельне виконання дій і координацію процесів, навіть без введення спеціальної термінології. У межах одного проекту можна поєднувати графіку, анімацію, звук і рух, що дозволяє створювати цілісні мультимедійні продукти. Це забезпечує формування уявлення про програмування як засіб розв'язування формалізованих задач та універсальний інструмент для реалізації творчих і навчальних ідей. Робота з такими проектами сприяє розвитку комплексного бачення цифрового продукту та усвідомлення ролі програмування в сучасному інформаційному середовищі.

Scratch підтримує різні рівні складності навчальних завдань, що дозволяє використовувати його як на початкових етапах навчання програмуванню, так і для створення досить складних проектів. Простота входження поєднується з можливістю поступового розширення функціональних можливостей програм, що забезпечує реалізацію принципу поетапного ускладнення. Така характеристика середовища створює умови для диференціації навчальної діяльності та врахування індивідуальних освітніх потреб.

У процесі роботи з Scratch здобувачі освіти опановують базові алгоритмічні конструкції, зокрема послідовність дій, повторення, розгалуження, події та змінні. Засвоєння цих понять відбувається не через формальне вивчення визначень, а через практичну діяльність і аналіз результатів виконання програм. Такий підхід сприяє формуванню глибшого розуміння сутності алгоритмів і створює підґрунтя для подальшого опанування більш складних мов і середовищ програмування.

Важливою особливістю Scratch є також можливість спільної роботи та обміну проектами. Ознайомлення з готовими роботами, їх аналіз і модифікація створюють умови для розвитку комунікативних і рефлексивних умінь, а також формують уявлення про програмування як соціально значущу діяльність. У методичному контексті це дозволяє організувати навчання у формі проектної діяльності та сприяти розвитку навичок співпраці.

Отже, середовище Scratch поєднує простоту інтерфейсу з широкими можливостями для організації навчальної діяльності. Робота в ньому не зводиться лише до засвоєння окремих команд чи технічних прийомів. Насамперед Scratch створює умови для розвитку алгоритмічного мислення, уміння планувати послідовність дій, перевіряти результат і вдосконалювати власні рішення. Завдяки візуальній структурі блоків і можливості експериментувати з проектами середовище сприяє творчій активності та усвідомленню опануванню основ програмування як складової інформаційної освітньої галузі.

1.3. Методичні передумови навчання програмування в середовищі Scratch

Навчання програмування на початкових етапах потребує особливого методичного підходу, оскільки воно поєднує елементи логічного мислення, планування діяльності, роботи з абстрактними моделями та взаємодії з цифровим середовищем. Ефективність такого навчання значною мірою залежить від добору програмного середовища, яке забезпечує доступність, наочність і можливість поступового ускладнення навчальних завдань. У цьому контексті Scratch розглядається як середовище, що створює сприятливі умови для формування алгоритмічного мислення без перевантаження формальними аспектами програмування.

Методична цінність Scratch пов'язана передусім із його візуальною природою. Створення програми відбувається через поєднання графічних блоків, кожен з яких відображає окрему дію або логічну конструкцію. Такий спосіб організації роботи дозволяє студентам осмислювати послідовність дій, умови та повторення як змістові елементи алгоритму. Відсутність складних правил запису та необхідності запам'ятовувати службові слова зменшує зайве інтелектуальне навантаження й створює умови для глибшого розуміння базових алгоритмічних понять.

Середовище Scratch побудоване таким чином, що синтаксичні помилки фактично виключені. Блоки поєднуються лише у структурно допустимі комбінації, тому процес створення програми відбувається без технічних збоїв, пов'язаних із формою запису. У центрі уваги опиняється логіка розв'язання завдання: вибір способу дії, перевірка умов, організація повторень. Помилки, що виникають під час роботи, здебільшого стосуються змісту алгоритму, що надає навчальній діяльності дослідницького характеру та сприяє усвідомленню причин і наслідків власних програмних рішень.

Scratch забезпечує широкі можливості для роботи з візуальними та мультимедійними даними. Поєднання графіки, анімації, звуку й руху дозволяє створювати програми, які є зрозумілими й наочними навіть на ранніх етапах навчання. Це сприяє формуванню мотивації до навчальної діяльності, оскільки результат програмування має безпосереднє візуальне відображення і може бути швидко оцінений. Можливість експериментувати з параметрами програм у процесі виконання завдання стимулює пізнавальний інтерес і розвиває здатність аналізувати наслідки власних дій.

Методично значущим є також об'єктно-орієнтований характер програм, створюваних у Scratch. Кожен об'єкт має власні властивості, зовнішній вигляд і набір скриптів, що дозволяє моделювати взаємодію між кількома об'єктами в межах одного проєкту. Така організація програмної діяльності сприяє формуванню уявлень про паралельність процесів, координату дії та взаємозалежність елементів системи.

Особливу роль у навчанні програмування відіграє можливість реалізації алгоритмів із повторенням, розгалуженням і використанням подій. Ці конструкції в Scratch мають наочне представлення, що дозволяє поступово переходити від лінійних алгоритмів до складніших моделей поведінки програм. Робота з умовами й сенсорами формує розуміння того, що

виконання програми залежить від заздалегідь визначеної послідовності дій та поточного стану середовища. Такий підхід створює підґрунтя для розвитку логічного мислення та здатності приймати рішення в змінних умовах.

Використання змінних у Scratch доцільно розглядати як етап, що логічно випливає з потреби зберігати й опрацьовувати дані в межах програми. Змінні вводяться не абстрактно, а в контексті конкретних завдань, де їх роль є зрозумілою й обґрунтованою. Це дозволяє сформувати усвідомлене ставлення до роботи з даними та підготувати здобувачів освіти до опанування більш формалізованих мов програмування в майбутньому.

Методичний потенціал Scratch значною мірою реалізується через проектну діяльність. Створення власних програм і проектів дозволяє інтегрувати різні алгоритмічні конструкції, розвивати навички планування, аналізу та презентації результатів. Проектна форма роботи сприяє формуванню відповідальності за результат, розвитку рефлексивних умінь і здатності аргументовано обґрунтовувати власні рішення. Крім того, можливість обміну проектами та їх модифікації створює умови для формування навичок співпраці та ознайомлення з елементами цифрової культури.

Таким чином, середовище Scratch створює комплексні методичні умови для навчання програмування на початкових етапах. Його візуальна природа, захист від синтаксичних помилок, підтримка алгоритмічних конструкцій і проектної діяльності дозволяють організувати навчання як поступовий, усвідомлений і діяльнісно орієнтований процес. У межах навчально-методичного посібника Scratch розглядається як ефективний засіб формування алгоритмічного мислення та готовності до подальшого опанування цифрових технологій.

РОЗДІЛ II. ЛАБОРАТОРНИЙ ПРАКТИКУМ

■ ЛАБОРАТОРНА РОБОТА №1

Знайомство з середовищем Scratch.

Створення простої анімації руху спрайта

Мета. Сформувати у студентів початкові практичні вміння роботи в середовищі Scratch, вміння орієнтуватися в інтерфейсі програми, створювати прості скрипти для керування спрайтом і аналізувати навчальні дії, які виконують молодші школярі під час роботи з такими завданнями.

Завдання лабораторної роботи:

- ознайомитися з інтерфейсом середовища Scratch;
- визначити призначення основних елементів робочого вікна;
- навчитися запускати програму за допомогою події;
- створити просту анімацію руху спрайта;
- проаналізувати дидактичну цінність подібного завдання для початкової школи.

Матеріали та засоби:

- середовище Scratch (онлайн або офлайн-версія);
- персональний комп'ютер або ноутбук;
- текстовий редактор;
- засіб створення скріншотів.

Хід виконання лабораторної роботи

1. Ознайомлення з інтерфейсом Scratch

1. Запустіть середовище Scratch.
2. Розгляньте головне вікно програми.
3. Зверніть увагу на такі елементи:
 - сцену;
 - список спрайтів;
 - область блоків команд;
 - область складання скриптів;
 - кнопки запуску та зупинки програми.

НАПРИКЛАД. Після запуску Scratch я звернув(ла) увагу, що робоче вікно поділене на кілька зон. Ліворуч розташовані блоки команд, у центрі – область для складання скриптів, праворуч – сцена, де одразу видно результат виконання програми.

ЗВЕРНІТЬ УВАГУ! Інтерфейс Scratch побудований за принципом візуальної зрозумілості, що є важливим для роботи з молодшими школярами.

2. Ознайомлення зі спрайтом

1. Зверніть увагу на стандартний спрайт (Kit).
2. Визначте:
 - його положення на сцені;
 - напрям руху;
 - розмір.
3. Перемістіть спрайт у лівий нижній кут сцени.

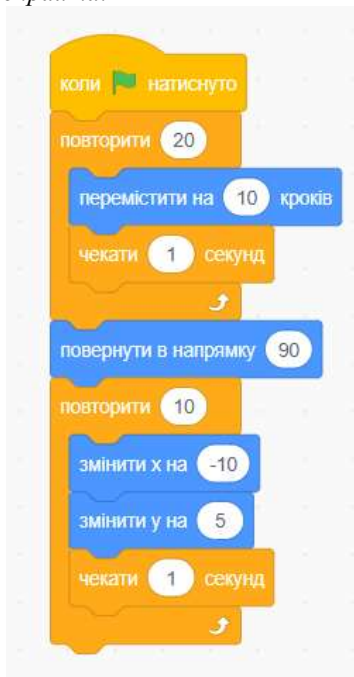
НАПРИКЛАД. *Я перемістив(ла) спрайт у лівий нижній кут сцени, перетягнувши його мишею. Це дозволило задати початкову позицію персонажа перед початком руху.*


ЗВЕРНІТЬ УВАГУ! *Такі дії формують у дітей просторові уявлення ще до введення формальних понять координат.*

3. Створення скрипта руху спрайта

1. Перейдіть у вкладку Скрипти для вибраного спрайта.
2. З категорії Події додайте блок запуску програми.
3. З категорії Рух додайте блоки переміщення спрайта.
4. З'єднайте блоки у правильній послідовності.

Для виконання лабораторної роботи необхідно створити такий програмний код для стандартного спрайта.



Перевірте створений проект – натисніть зелений прапорець . Переконайтеся, що Ваш спрайт виконує дії, зазначені в завданні.

ЗВЕРНІТЬ УВАГУ! *На цьому етапі не потрібно використовувати цикли або складні команди. Головне – усвідомити послідовність виконання дій.*

7. Створення власного скрипта руху спрайта

1. Змініть програму так, щоб Кіт після того, як дійде до правого кута, йшов вгору до самого кінця.
2. Складіть наступну анімацію для спрайту Кіт: Кіт знаходиться в правому нижньому кутку, потім він доходить до середини екрану, після чого розвертається і повертається назад.
3. Складіть наступну анімацію для спрайту Кіт: Кіт проходить 4 рази по периметру екрану

8. Методичний аналіз завдання

1. Визначте, які навчальні дії виконує дитина.
2. Проаналізуйте, які уміння формуються.
3. Зробіть короткий методичний висновок.

НАПРИКЛАД. *На мою думку, завдання формує розуміння послідовності дій, зв'язку між командою та результатом, а також розвиває просторові уявлення.*

ЗВЕРНІТЬ УВАГУ! *Методична цінність завдання полягає в простоті та наочності.*

Форма звітності

1. Створити файл звіту `LP_Scratch_1_Прізвище_Група.doc`.
2. Орієнтовна структура звіту:
 - характеристика середовища Scratch;
 - опис інтерфейсу;
 - опис створеного скрипта;
 - методичний аналіз;
 - узагальнюючий висновок.

Вимоги до оформлення

- обсяг: **3–4 сторінки**;
- шрифт: *Times New Roman*, 14;
- міжрядковий інтервал: 1,5;
- власні формулювання;
- **обов'язкова наявність скріншотів:**
 - головного вікна Scratch;

- зібраного скрипта;
- сцени з результатом виконання програми;
- скріншоти мають бути підписані та розміщені після відповідного опису;
- після скріншотів додайте покликання на створені скрипти.

Критерії оцінювання

- коректність програмного коду – 2 бали;
- усвідомлений опис дій – 1 бал;
- методичний аналіз – 1 бал;
- оформлення, скріншоти, покликання – 1 бал.

■ ЛАБОРАТОРНА РОБОТА №2

Створення анімації спрайта шляхом послідовної зміни його образів у середовищі Scratch.

Мета. Сформувати у студентів практичні вміння створювати анімацію в Scratch за допомогою зміни образів спрайта, уміння аналізувати навчальні дії молодших школярів під час виконання подібних завдань.

Завдання лабораторної роботи:

- ознайомитися з поняттям «костюм» у Scratch;
- додати нові костюми до спрайта;
- створити скрипт, що забезпечує зміну костюмів;
- отримати ефект анімації руху;
- здійснити методичний аналіз навчального завдання.

Матеріали та засоби:

- середовище Scratch (онлайн або офлайн-версія);
- персональний комп'ютер або ноутбук;
- текстовий редактор;
- засіб створення скріншотів.

Хід виконання лабораторної роботи

1. Ознайомлення з костюмами спрайта

1. Відкрийте попередній проєкт або створіть новий.
2. Виділіть стандартний спрайт (Кіт) або оберіть новий.
3. Перейдіть у вкладку «Образи».
4. Перегляньте наявні образи спрайта.

НАПРИКЛАД. Я перейшов(ла) у вкладку «Образи» та побачив(ла), що спрайт має кілька різних зображень, які відрізняються положенням _____ . Це дозволяє створювати ілюзію руху.

ЗВЕРНІТЬ УВАГУ! Для молодших школярів зміна костюмів є більш зрозумілою, ніж складні рухи або обчислення.

2. Додавання нового образу

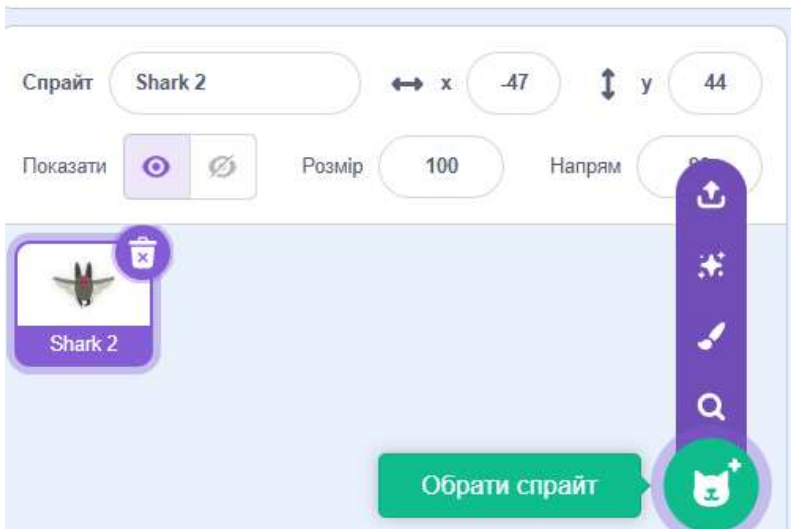
1. Додайте ще один образ до спрайта (скопійуйте наявний або оберіть із бібліотеки).
2. За потреби змініть положення частин зображення.
3. Збережіть зміни.

НАПРИКЛАД. Я додав(ла) новий костюм і трохи змінив положення лап персонажа, щоб під час зміни костюмів було помітно рух.

ЗВЕРНІТЬ УВАГУ! Навіть незначна зміна зображення створює відчуття анімації.

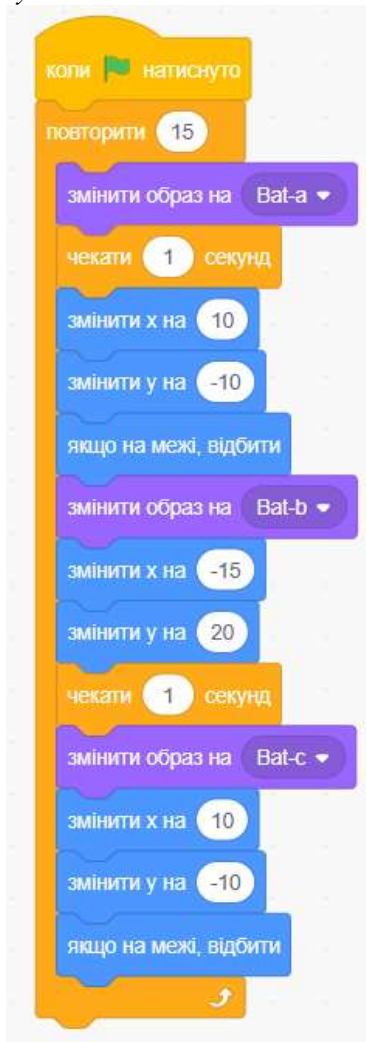
3. Створення скрипта зміни образів


1. Перейдіть у вкладку Скрипти.
2. Оберіть спрайт (Кажан).



3. З категорії Події додайте блок запуску програми.
4. З категорії Вигляд додайте блок зміни костюма.
5. Додайте паузу між змінами костюмів.

Зразок програмного коду для виконання завдання



Перевірте створений проект – натисніть зелений прапорець . Переконайтеся, що Ваш спрайт виконує дії, зазначені в завданні.

НАПРИКЛАД. *Я створив(ла) програму, у якій після натискання зеленого прапорця спрайт починає змінювати свої образи з невеликою паузою. У результаті складається враження, що персонаж рухається.*

ЗВЕРНІТЬ УВАГУ! *На цьому етапі не потрібно використовувати цикли – достатньо послідовної зміни костюмів.*

4. Створення власного скрипта зміни образу спрайта.

1. Змініть проєкт так, щоб кажан рухався вимахуючи крилами та переміщуючись по екрану.
2. Використовуючи галерею спрайтів, створіть анімацію танцюючої людини.
3. Спостерігайте за анімацією спрайта.
4. Змініть час паузи та порівняйте результат.

НАПРИКЛАД. *Після зменшення часу очікування анімація стала швидшою, що дозволяє пояснити учням вплив параметрів команд на результат.*

ЗВЕРНІТЬ УВАГУ! *Регулювання швидкості анімації розвиває уявлення про темп і послідовність дій.*

5. Методичний аналіз завдання

1. Визначте, які навчальні дії виконує дитина.
2. Проаналізуйте, які вміння формуються.
3. Зробіть методичний висновок.

НАПРИКЛАД. *Це завдання формує у дітей уявлення про анімацію, послідовність команд та зв'язок між зображенням і рухом.*

ЗВЕРНІТЬ УВАГУ! *Анімація через зміну образів є підготовчим етапом до використання циклів.*

Форма звітності

1. Створити файл звіту `ЛР_Scratch_2_Прізвище_Група.doc`.
2. Орієнтовна структура звіту:
 - опис образів спрайта;
 - опис програмного коду;
 - аналіз результату;
 - методичний висновок.

Вимоги до оформлення

- обсяг: **3–4 сторінки**;
- шрифт: *Times New Roman*, 14;
- міжрядковий інтервал: 1,5;
- власні формулювання;
- обов'язкова наявність скріншотів:
 - вкладки «Костюми»;

- зібраного скрипта;
- сцени з анімацією;
- скріншоти мають бути підписані;
- після скріншотів додайте покликання на створені скрипти..

Критерії оцінювання

- коректність програмного коду – 2 бала;
- усвідомлений опис анімації – 1 бал;
- методичний аналіз – 1 бал;
- Оформлення і скріншоти – 1 бал.

▣ ЛАБОРАТОРНА РОБОТА №3

Додавання нових сцен до проєкту. Зміна сцени

Мета. Сформувати у студентів практичні вміння працювати зі сценами в середовищі Scratch, додавати нові фони, керувати зміною сцен за допомогою програмних команд та аналізувати дидактичну доцільність використання сцен у навчальних проєктах для початкової школи.

Завдання лабораторної роботи:

- ознайомитися з поняттям «сцена» у Scratch;
- навчитися додавати нові фони до проєкту;
- створити програму зі зміною сцени за подією;
- проаналізувати навчальні дії, які виконує дитина;
- здійснити методичний аналіз завдання.

Матеріали та засоби:

- середовище Scratch (онлайн або офлайн-версія);
- персональний комп'ютер або ноутбук;
- текстовий редактор;
- засіб створення скріншотів.

Хід виконання лабораторної роботи

1. Ознайомлення з поняттям сцени в Scratch

1. Запустіть середовище Scratch і відкрийте новий проєкт.
2. Зверніть увагу на область сцени та кнопку керування фонами.
3. Визначте, що сцена може містити кілька фонів, які змінюються під час виконання програми.

НАПРИКЛАД. Після запуску Scratch я звернув(ла) увагу, що сцена є окремим об'єктом проєкту і має власні налаштування. Я зрозумів(ла), що зміна сцени дозволяє змінювати сюжет або умови виконання програми.

ЗВЕРНІТЬ УВАГУ! Для молодших школярів сцена часто сприймається як «місце події», що полегшує розуміння сюжету програми.

2. Додавання нових фонів до сцени

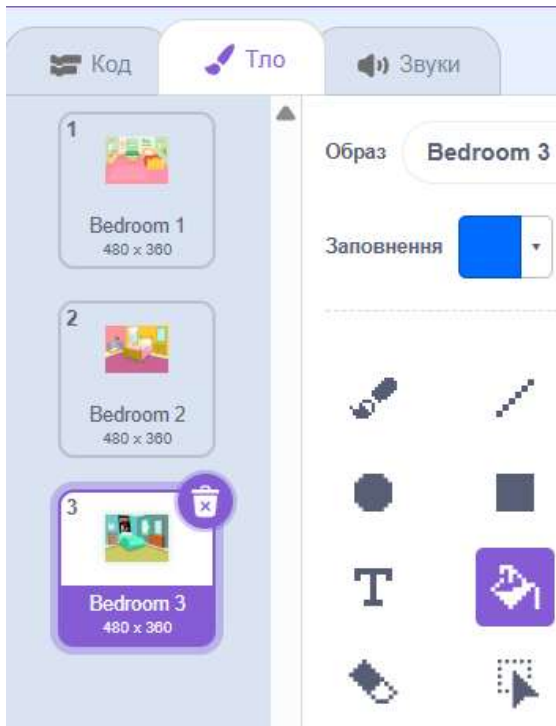
1. Натисніть кнопку додавання фону (гло).
2. Оберіть два різні фони з бібліотеки Scratch (наприклад, «Кімната» та «Парк»).
3. Перейдіть у вкладку Фони та перегляньте додані зображення.

НАПРИКЛАД. Я додав(ла) до проєкту два фони: перший – для початку програми, другий – для наступного етапу. Це дозволило створити відчуття зміни подій.

ЗВЕРНІТЬ УВАГУ! Кількість фонів має бути обмеженою, щоб не перевантажувати сприйняття дитини.

3. Ознайомлення з командами керування сценою

1. Виділіть об'єкт Сцена.
2. Перейдіть до категорії Вигляд.
3. Знайдіть команди:
 - «встановити фон ...»;
 - «наступний фон».



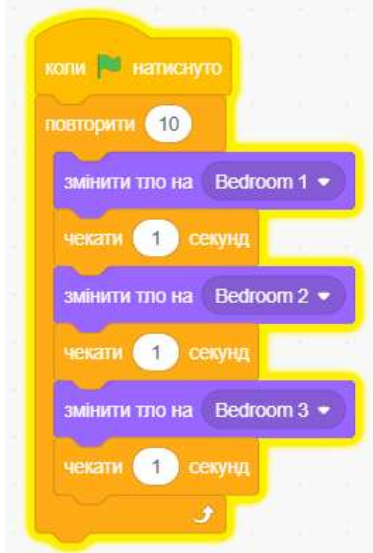
НАПРИКЛАД. Я помітив(ла), що для сцени доступні окремі команди, які дозволяють змінювати фон незалежно від спрайтів.


ЗВЕРНІТЬ УВАГУ! Робота зі сценою формує уявлення про різні об'єкти проєкту та їхню роль.

4. Створення програми зі зміною сцени

1. Додайте блок запуску програми.
2. Встановіть початковий фон сцени.
3. Додайте команду зміни сцени з паузою.

Зразок програмного коду для виконання завдання



Перевірте створений проєкт – натисніть зелений прапорець .

НАПРИКЛАД. Я створив(ла) програму, у якій після запуску встановлюється перше тло, а через кілька секунд сцена змінюється. Це створює ефект переходу між подіями.

ЗВЕРНІТЬ УВАГУ! На початковому етапі доцільно використовувати просту послідовну зміну сцен без умов і циклів.

5. Створення власного скрипта зміни фону проєкта.

1. Запустіть програму.
2. Змініть порядок зміни фонів.
3. Створіть проєкт, який демонструє зміну пір року.
4. Змініть час очікування або порядок фонів.

НАПРИКЛАД. Після зміни часу очікування я помітив(ла), що сцена змінюється швидше або повільніше, що впливає на сприйняття сюжету.

ЗВЕРНІТЬ УВАГУ! Регулювання часу є важливим елементом побудови навчального сценарію.

6. Методичний аналіз завдання

1. Визначте, які навчальні дії виконує дитина.
2. Проаналізуйте, які вміння формуються.
3. Зробіть методичний висновок.

НАПРИКЛАД. На мою думку, завдання формує у дітей уявлення про послідовність подій і розвиває сюжетне мислення.

ЗВЕРНІТЬ УВАГУ! Зміна сцен є ефективним засобом інтеграції програмування з мовно-літературною та мистецькою галузями.

Форма звітності

1. Створити файл звіту `LP_Scratch_3_Прізвище_Група.doc`.
2. Орієнтовна структура звіту:
 - опис використаних фонів;
 - опис програмного коду;
 - аналіз результату;
 - методичний висновок.

Вимоги до оформлення

- обсяг: **3–4 сторінки**;
- шрифт: *Times New Roman*, 14;
- міжрядковий інтервал: 1,5;
- власні формулювання;
- **обов'язкова наявність скріншотів:**
 - вкладки «Фони»;
 - зібраного скрипта для сцени;
 - сцени до і після зміни фону;
- скріншоти мають бути підписані;
- після скріншотів додайте покликання на створені скрипти.

Критерії оцінювання

- коректність керування сценами – 2 бала;
- усвідомлений опис результату – 1 бал;
- методичний аналіз – 1 бал;
- оформлення і скріншоти – 1 бал.

■ ЛАБОРАТОРНА РОБОТА №4

Додавання нового спрайта до проєкту. Створення анімації для нього

Мета. Сформувати у студентів практичні вміння додавати нові спрайти до проєкту Scratch, створювати для них прості програми анімації, аналізувати навчальні дії молодших школярів під час виконання подібних завдань.

Завдання лабораторної роботи:

- ознайомитися зі способами додавання спрайтів у Scratch;
- додати новий спрайт до проєкту;
- створити програму анімації для нового спрайта;
- синхронізувати дії нового спрайта із запуском проєкту;
- здійснити методичний аналіз завдання.

Матеріали та засоби:

- середовище Scratch (онлайн або офлайн-версія);
- персональний комп'ютер або ноутбук;
- текстовий редактор;
- засіб створення скріншотів.

Хід виконання лабораторної роботи

1. Додавання нового спрайта до проєкту

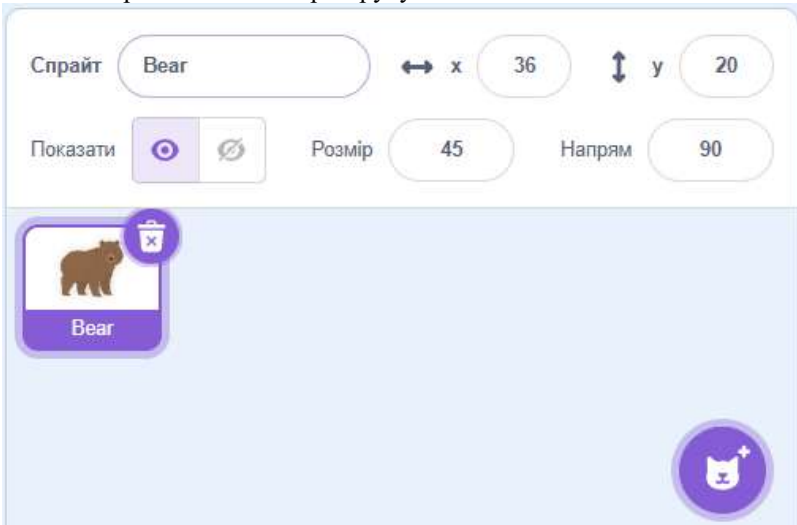
1. Відкрийте середовище Scratch і створіть новий проєкт або відкрийте попередній.
2. Натисніть кнопку «Обрати спрайт».
3. Оберіть будь-який спрайт із бібліотеки (наприклад, м'яч або тварину).
4. Переконайтеся, що новий спрайт з'явився у списку спрайтів.

НАПРИКЛАД. *Я додав(ла) до проєкту новий спрайт із бібліотеки. Після цього він з'явився на сцені поряд зі стандартним спрайтом, і для нього можна створювати окрему програму.*

ЗВЕРНІТЬ УВАГУ! *Кожен спрайт у Scratch має власні скрипти, костюми та налаштування.*

2. Ознайомлення з властивостями нового спрайта

1. Виділіть доданий спрайт.
2. Змініть його розмір або положення на сцені.
3. За потреби змініть напрям руху.



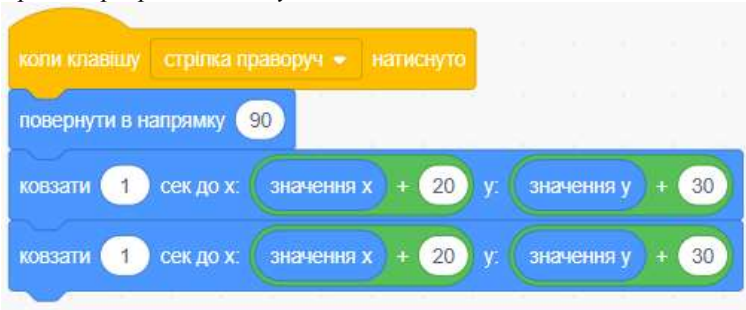
НАПРИКЛАД. Я зменшив(ла) розмір спрайта та перемістив(ла) його в іншу частину сцени, щоб він не перекривав інші об'єкти.

ЗВЕРНІТЬ УВАГУ! Робота з властивостями спрайта розвиває просторове мислення учнів.

3. Створення програми анімації для нового спрайта

1. Перейдіть у вкладку Скрипти для нового спрайта (жук).
2. Додайте блок запуску програми.
3. Додайте команди руху або зміни вигляду спрайта.
4. За потреби додайте паузи між діями (використовуємо команди з блоку «події», «рух», «оператори»).

Зразок програмного коду для виконання завдання





НАПРИКЛАД. Я створив(ла) програму, у якій новий спрайт після запуску переміщується вперед, робить паузу та змінює напрям руху. У результаті він виглядає як анімований об'єкт.

ЗВЕРНІТЬ УВАГУ! На цьому етапі достатньо простих рухів без використання циклів або умов.

4. Аналіз результату виконання програми

1. Запустіть проєкт.
2. Простежте за поведінкою нового спрайта.
3. Додайте можливість здійснювати рух спрайта (жука) при натисканні на кнопки “стрілка вгору” і “стрілка вниз”.
4. Створіть анімацію крокуючої людини, з кнопками керування; стиль руху, фон і звук вибрати довільно.
5. Змініть параметри команд і проаналізуйте результат.

НАПРИКЛАД. Після зміни кількості кроків спрайт почав рухатися швидше, що дозволило наочно побачити вплив параметрів команди.

ЗВЕРНІТЬ УВАГУ! Експериментування з параметрами команд є важливою складовою навчального процесу.

5. Методичний аналіз завдання

1. Визначте, які навчальні дії виконує дитина.
2. Проаналізуйте, які вміння формуються.
3. Зробіть методичний висновок.

НАПРИКЛАД. Завдання формує у дітей уявлення про об'єкти програми, розвиває вміння керувати кількома персонажами та розуміння зв'язку між командою і результатом.

ЗВЕРНІТЬ УВАГУ! Додавання нових спрайтів є основою для подальшого створення сюжетних проєктів.

Форма звітності

1. Створити файл звіту **ЛР_Scratch_4_Прізвище_Група.doc**.
2. Орієнтовна структура звіту:
 - опис доданого спрайта;
 - опис програмного коду;
 - аналіз результату;
 - методичний висновок.

Вимоги до оформлення

- обсяг: 3–4 сторінки;
- шрифт: *Times New Roman*, 14;
- міжрядковий інтервал: 1,5;
- власні формулювання;
- обов'язкова наявність скріншотів:
 - списку спрайтів;
 - зібраного скрипта;
 - сцени з анімацією;
- скріншоти мають бути підписані;
- після скріншотів додайте покликання на створені скрипти.

Критерії оцінювання

- коректність програмного коду – 2 бала;
- усвідомлений опис анімації – 1 бал;
- методичний аналіз – 1 бал;
- оформлення і скріншоти – 1 бал.

■ ЛАБОРАТОРНА РОБОТА №5 Додавання звукових ефектів до проєкту

Мета. Сформувати у студентів практичні вміння додавати звуки до спрайтів у середовищі Scratch, керувати їх відтворенням за допомогою програмних команд та аналізувати дидактичну доцільність використання звукових ефектів у навчальних проєктах для початкової школи.

Завдання лабораторної роботи:

- ознайомитися з можливостями роботи зі звуками в Scratch;
- додати звуковий ефект до спрайта;
- створити програму, що відтворює звук за подією;
- поєднати звуковий ефект з рухом або дією спрайта;
- здійснити методичний аналіз навчального завдання.

Матеріали та засоби:

- середовище Scratch (онлайн або офлайн-версія);
- персональний комп'ютер або ноутбук з навушниками або колонками;
- текстовий редактор;
- засіб створення скріншотів.

Хід виконання лабораторної роботи

1. Ознайомлення з роботою зі звуками в Scratch

1. Відкрийте проект Scratch або створіть новий.
2. Виділіть будь-який спрайт (наприклад, стандартного Кота).
3. Перейдіть у вкладку Звуки.
4. Перегляньте наявні звуки спрайта.

НАПРИКЛАД. *Я перейшов(ла) у вкладку «Звуки» і побачив(ла), що кожен спрайт може мати один або кілька звуків, які використовуються у програмі.*

ЗВЕРНІТЬ УВАГУ! *Звуки в Scratch належать конкретному спрайту, а не всьому проєкту одразу.*

2. Додавання нового звукового ефекту

1. Натисніть кнопку «Обрати звук».
2. Оберіть будь-який короткий звук із бібліотеки (наприклад, кроки, удар або сигнал).
3. За потреби прослухайте звук та підтвердьте вибір.

НАПРИКЛАД. *Я додав(ла) короткий звуковий ефект, який добре підходить для супроводу руху спрайта.*

ЗВЕРНІТЬ УВАГУ! *Для початкової школи доцільно використовувати короткі, ненав'язливі звуки.*

3. Ознайомлення з командами керування звуком

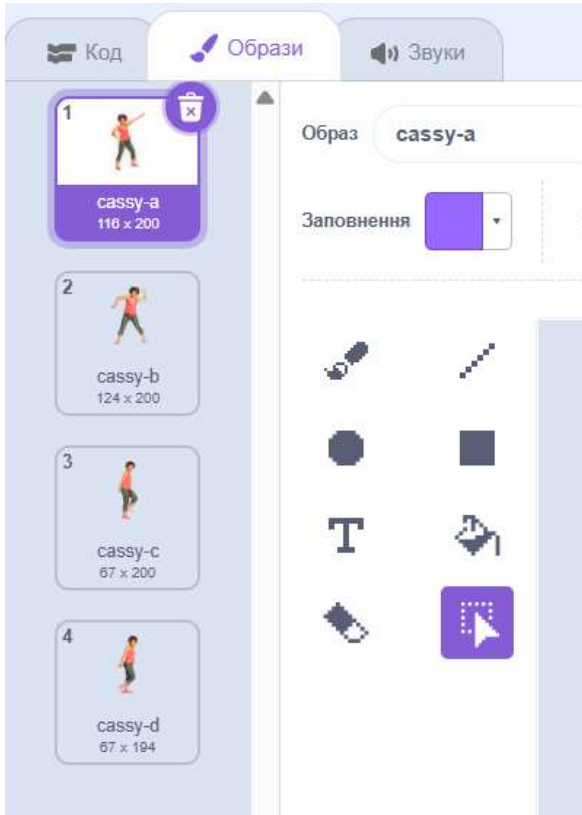
1. Перейдіть у вкладку Скрипти.
2. Відкрийте категорію Звук.
3. Знайдіть команди:
 - відтворити звук [];
 - відтворити звук [] до кінця;
 - встановити гучність.

НАПРИКЛАД. *Я звернув(ла) увагу, що Scratch дозволяє або запускати звук паралельно з іншими діями, або чекати завершення його відтворення.*

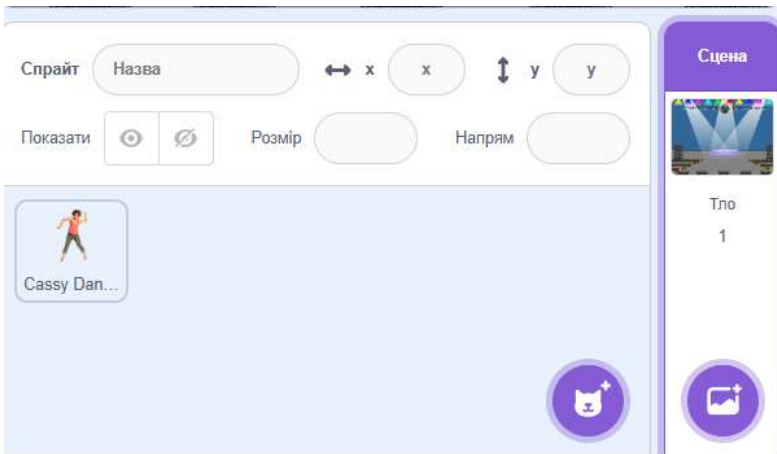
ЗВЕРНІТЬ УВАГУ! *Команда «відтворити звук до кінця» зупиняє виконання наступних команд, доки звук не завершиться.*

4. Створення спрайту з відтворенням звуку

1. Запустіть програму.
2. Замініть стандартний спрайт (Кіт) новим (наприклад .
3. Додайте до проекту спрайт під назвою Cassy Dance.
4. Для нового спрайту додайте ще три образи cassy-b, cassy-c, cassy-d. У Вас повинно вийти приблизно так, як показано на малюнку



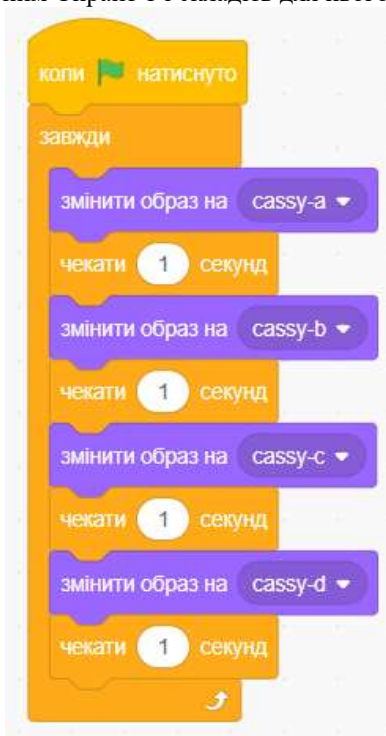
5. Зробіть активною Сцену та перейдіть в закладку Звуки.



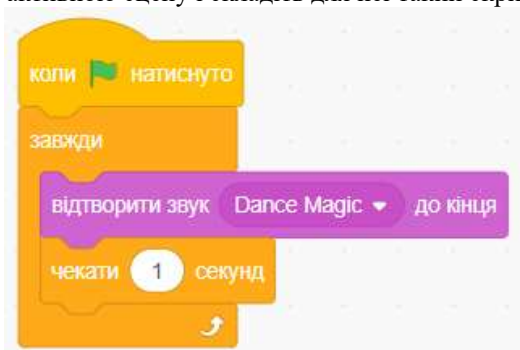
6. Використовуючи кнопку «Обрати звук», додайте до проекту новий звук Dance Funky або власний.



7. Зробіть активним Спрайт 1 і складіть для нього такий скрипт



8. Зробіть активною сцену і складіть для неї такий скрипт.



9. Для запуску проєкту натисніть на зелений прапорець. Переконайтеся, що він працює нормально.

5. Створення власного проєкту з відтворенням звуку

1. Відкрийте попередній проєкт або створіть новий.
2. Додайте до проєкту будь-який фон за вашим бажанням.

3. Створіть проект, в якому людина займається активним видом спорту під музику.

ЗВЕРНІТЬ УВАГУ! *Гучність звуку має відповідати віковим особливостям учнів.*

6. Методичний аналіз завдання

1. Визначте, які навчальні дії виконує дитина.
2. Проаналізуйте, які вміння формуються.
3. Зробіть методичний висновок.

НАПРИКЛАД. *Завдання формує у дітей уявлення про мультимедійні елементи програми та розвиває зв'язок між подією, дією і звуковим ефектом.*

ЗВЕРНІТЬ УВАГУ! *Звук має підсилювати навчальну дію, а не відволікати від неї.*

Форма звітності

1. Створити файл звіту ЛР_Scratch_5_Прізвище_Група.doc.
2. Орієнтовна структура звіту:
 - опис використаного звуку;
 - опис програмного коду;
 - аналіз результату;
 - методичний висновок.

Вимоги до оформлення

- обсяг: 3–4 сторінки;
- шрифт: *Times New Roman*, 14;
- міжрядковий інтервал: 1,5;
- власні формулювання;
- обов'язкова наявність скріншотів:
 - вкладки «Звуки»;
 - зібраного скрипта;
 - сцени під час виконання програми;
- скріншоти мають бути підписані;
- після скріншотів додайте покликання на створені скрипти.

Критерії оцінювання

- Коректність використання звуків – 2 бала;
- Усвідомлений опис результату – 1 бал;
- Методичний аналіз – 1 бал;
- Оформлення і скріншоти – 1 бал.

■ ЛАБОРАТОРНА РОБОТА №6

Створення складної анімації в Scratch за участю одного спрайта

Мета. Сформувати у студентів уміння створювати складні анімаційні сцени в Scratch на основі одного спрайта, поєднуючи кілька типів команд, аналізувати навчальні дії молодших школярів під час виконання подібних завдань.

Завдання лабораторної роботи:

- створити анімацію з кількох послідовних дій одного спрайта;
- використати рух, зміну костюмів і звуковий ефект;
- синхронізувати дії спрайта в часі;
- проаналізувати результат виконання програми;
- здійснити методичний аналіз завдання.

Матеріали та засоби:

- середовище **Scratch** (онлайн або офлайн-версія);
- персональний комп'ютер або ноутбук з навушниками або колонками;
- текстовий редактор;
- засіб створення скріншотів.

Хід виконання лабораторної роботи

1. Підготовка спрайта до анімації

1. Відкрийте новий проект у Scratch.
2. Оберіть один спрайт (стандартний або з бібліотеки).
3. Переконайтеся, що спрайт має декілька костюмів.
4. За потреби додайте або відредагуйте костюми.

НАПРИКЛАД. *Я обрав(ла) спрайт, який має кілька костюмів. Це дозволяє створити анімацію руху без додавання інших персонажів.*

ЗВЕРНІТЬ УВАГУ! *Складна анімація не означає багато спрайтів – важливою є послідовність дій.*

2. Планування послідовності анімації

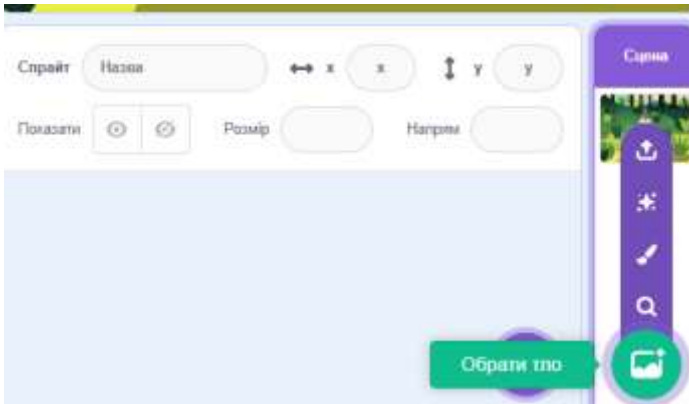
1. Визначте, які дії виконуватиме спрайт:
 - початкове положення;
 - рух;
 - зміна костюмів;
 - звуковий ефект;
 - завершення анімації.
2. Продумайте порядок виконання цих дій.

НАПРИКЛАД. Я запланував(ла), що спрайт спочатку з'являється внизу сцени, потім рухається, змінює костюми й наприкінці відтворює звук.

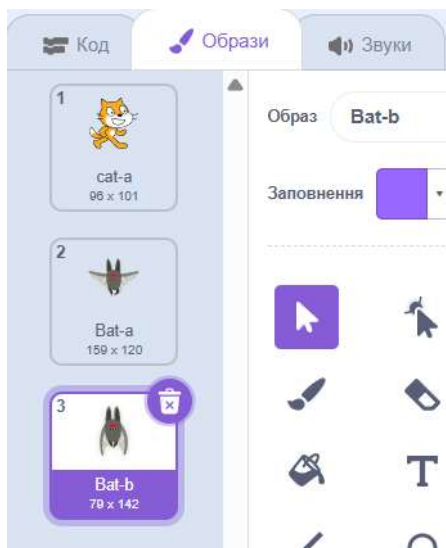
ЗВЕРНІТЬ УВАГУ! Планування дій є важливою передумовою формування алгоритмічного мислення.

3. Створення програми складної анімації

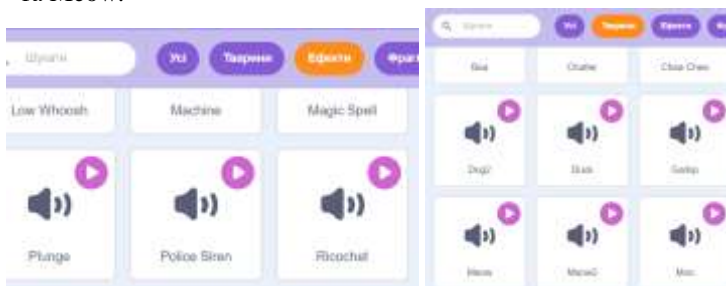
1. Запустіть програму Scratch.
2. В об'єктах виберіть Сцену.
3. Натисніть «Обрати тло» та оберіть нове тло.



4. Перемістіть спрайт (Кіт) в лівий нижній кут.
5. Перейдіть в закладку Образи, видаліть образ cat-b. З папки Тварини імпортуйте образи bat-a та bat-b.



6. Перейдіть в закладку Звуки, з папки Effect імпортуйте звук Ricochet та Meow.



7. Перейдіть до вкладки «Код» та складіть програму, в якій кіт йде зліва направо до кінця, говорить “Привіт”, повертається на 90 градусів, чекає 3 секунди, повертається назад, перетворюється в кажана та летить за діагоналлю, запитуючи в кінці “Все зрозуміло?”

5. Методичний аналіз завдання

1. Визначте, які навчальні дії виконує дитина.
2. Проаналізуйте, які вміння формуються.
3. Зробіть методичний висновок.

НАПРИКЛАД. Завдання формує у дітей уміння поєднувати кілька типів дій, планувати послідовність та прогнозувати результат.

ЗВЕРНІТЬ УВАГУ! Складна анімація є підготовчим етапом до створення власних проєктів.

Форма звітності

1. Створити файл звіту `ЛР_Scratch_6_Прізвище_Група.doc`.
2. Орієнтовна структура звіту:
 - опис задуму анімації;
 - опис програмного коду;
 - аналіз результату;
 - методичний висновок.

Вимоги до оформлення

- обсяг: 3–4 сторінки;
- шрифт: *Times New Roman*, 14;
- міжрядковий інтервал: 1,5;
- власні формулювання;
- обов'язкова наявність скріншотів:
 - спрайта і його костюмів;
 - зібраного скрипта;
 - сцени на різних етапах анімації;
- скріншоти мають бути підписані;
- після скріншотів додайте покликання на створені скрипти.

Критерії оцінювання

- Коректність програмного коду – 2 бала;
- Цілісність анімації – 1 бал;
- Методичний аналіз – 1 бал;
- Оформлення і скріншоти – 1 бал.

■ ЛАБОРАТОРНА РОБОТА №7

Створення складної анімації з кількома спрайтами

Мета. Сформувати у студентів практичні вміння створювати анімаційні сцени з кількома спрайтами в Scratch, узгоджувати дії різних

персонажів, аналізувати дидактичну цінність таких завдань для формування алгоритмічного мислення учнів початкової школи.

Завдання лабораторної роботи:

- додати до проєкту кілька спрайтів;
- створити окремі програми для кожного спрайта;
- узгодити дії спрайтів у межах однієї анімації;
- проаналізувати результат виконання програми;
- здійснити методичний аналіз завдання.

Матеріали та засоби:

- середовище Scratch (онлайн або офлайн-версія);
- персональний комп'ютер або ноутбук;
- текстовий редактор;
- засіб створення скріншотів.

Хід виконання лабораторної роботи

1. Підготовка проєкту та спрайтів

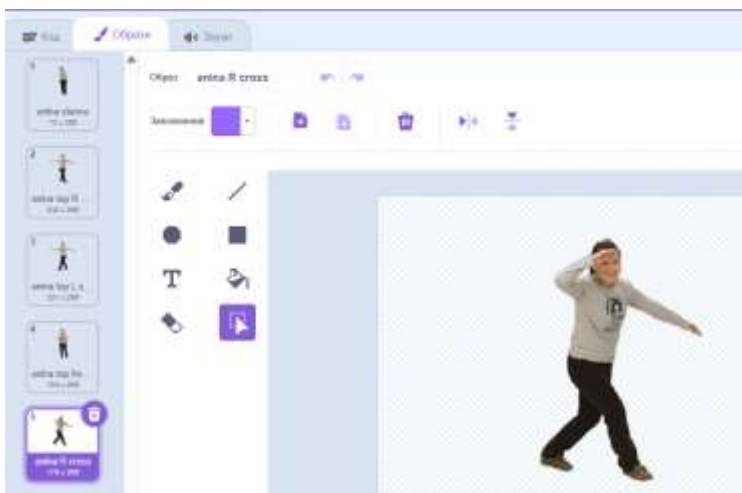
1. Відкрийте новий проєкт у Scratch.
2. Додайте два або три спрайти з бібліотеки.
3. Розмістіть їх у різних частинах сцени.
4. За потреби змініть розмір кожного спрайта.

НАПРИКЛАД. *Я додав(ла) до проєкту три спрайти та розмістив(ла) їх у різних частинах сцени, щоб кожен персонаж був добре видимий.*

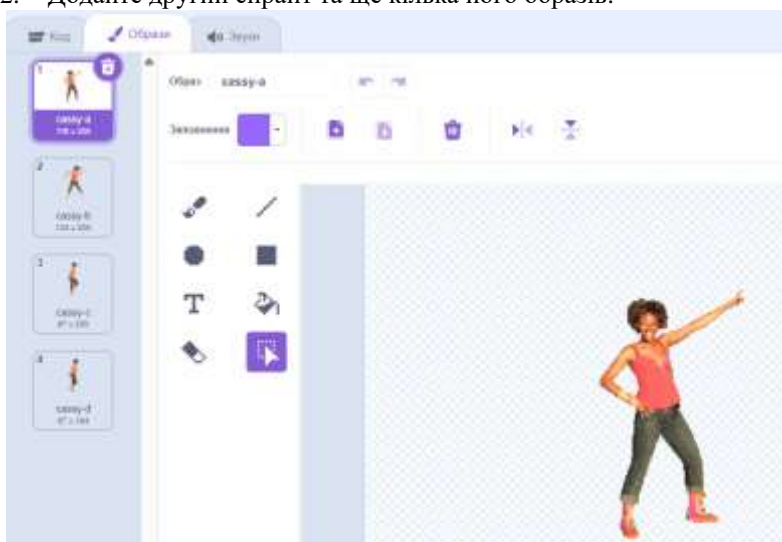
ЗВЕРНІТЬ УВАГУ! *Кількість спрайтів має бути невеликою, щоб дитина могла легко відстежувати їхні дії.*

2. Створення програми для першого спрайта

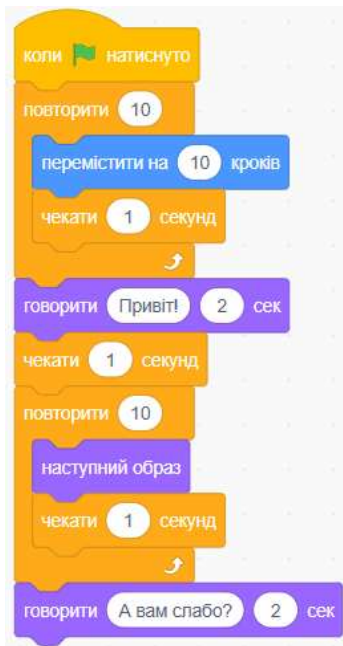
1. Виділіть перший спрайт та додайте до нього ще кілька образів.



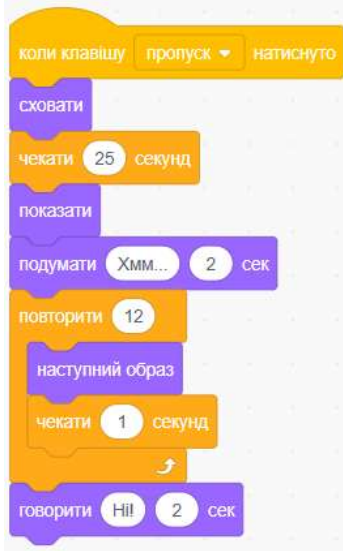
2. Додайте другий спрайт та ще кілька його образів.



3. Для Спрайта 1 складіть такий скрипт.



4. Для Спрайта 2 складіть довільний скрипт, наприклад



5. Перевірте роботу вашого проекту: перейдіть в режим презентації та натисніть на зелений прапорець.

ЗВЕРНІТЬ УВАГУ! *Кожен спрайт має власний набір команд. Пауза є простим способом синхронізації дій без використання складних конструкцій.*

3. Створення програми власної складної анімації з кількома спрайтами.

1. Використати щонайменше два спрайти (допускається три).
2. Для кожного спрайта створити окремий програмний код.
3. Забезпечити узгодженість дій спрайтів у часі:
 - одночасний початок дій, або
 - послідовний початок з використанням пауз.
4. Використати не менше двох типів команд для кожного спрайта, наприклад:
 - рух;
 - зміна костюмів;
 - пауза;
 - звук.
5. Анімація має мати логічний початок і завершення..
6. Простежте за діями всіх спрайтів.
7. За потреби змініть час пауз або напрям руху.

ЗВЕРНІТЬ УВАГУ! *Синхронізація є ключовим елементом складної анімації.*

Орієнтовні варіанти сюжетів (на вибір студента):

- два персонажі рухаються назустріч один одному;
- один спрайт з'являється, а другий – реагує на його появу;
- персонажі по черзі виконують рухи;
- один спрайт «запрошує» іншого рухом або звуком.

Сюжет має бути простим і зрозумілим, орієнтованим на можливості учнів початкової школи.

НАПРИКЛАД. *Я створив(ла) проєкт із двома спрайтами. Перший спрайт після запуску рухається до центру сцени, а другий починає рух через одну секунду. Дії спрайтів не заважають одна одній і утворюють послідовну анімацію.*

ЗВЕРНІТЬ УВАГУ!

- не потрібно ускладнювати програму великою кількістю команд;
- головне – узгодженість дій кількох спрайтів;
- кожен спрайт має виконувати зрозумілу й логічну роль;
- програма має бути такою, щоб її міг зрозуміти учень початкової школи.

4. Методичний аналіз завдання

1. Визначте, які навчальні дії виконує дитина.
2. Які навчальні дії виконує дитина, працюючи з такою програмою.
3. Які уміння та навички формуються під час створення анімації з кількома спрайтами.
4. Зробіть методичний висновок.

НАПРИКЛАД. Завдання розвиває вміння планувати дії кількох об'єктів і розуміти взаємозв'язок між ними.

ЗВЕРНІТЬ УВАГУ! Складні анімації формують основу для командної логіки програмування.

Форма звітності

1. Створити файл звіту `LP_Scratch_7_Прізвище_Група.doc`
2. Структура:
 - опис спрайтів;
 - опис програмних кодів;
 - аналіз взаємодії;
 - методичний висновок.

Вимоги до оформлення

- 3–4 сторінки;
- Times New Roman, 14;
- міжрядковий інтервал 1,5;
- обов'язкові скріншоти:
 - списку спрайтів;
 - скриптів кожного спрайта;
 - сцени з анімацією.
- скріншоти мають бути підписані;
- після скріншотів додайте покликання на створені скрипти.

Критерії оцінювання (5 балів)

- коректність роботи кількох спрайтів – 2 бала;
- узгодженість анімації – 1 бал;
- методичний аналіз – 1 бал;
- оформлення – 1 бал.

■ ЛАБОРАТОРНА РОБОТА №8

Створення та виконання алгоритмів з повторенням. Малювання геометричних фігур за допомогою олівця

Мета. Сформувати у студентів практичні вміння створювати алгоритми з повторенням у Scratch, використовувати розширення «Олівець»

для малювання фігур, аналізувати навчальні дії молодших школярів під час виконання подібних завдань.

Завдання лабораторної роботи:

- ознайомитися з розширенням «Олівець» у Scratch;
- навчитися використовувати блок «повторити»;
- створити алгоритм малювання геометричної фігури;
- поєднати команди руху, повороту та повторення;
- здійснити методичний аналіз завдання.

Матеріали та засоби:

- середовище Scratch (онлайн або офлайн-версія);
- персональний комп'ютер або ноутбук;
- текстовий редактор;
- засіб створення скріншотів.

Хід виконання лабораторної роботи

1. Підготовка середовища до малювання

1. Запустіть Scratch і створіть новий проєкт.
2. Натисніть кнопку «Додати розширення» (іконка пазла).
3. Оберіть розширення «Олівець».
4. Переконайтеся, що в списку блоків з'явилася категорія «Олівець».
5. Знайдіть блок повторити (). Зверніть увагу, що всередину цього блоку можна вкладати інші команди.

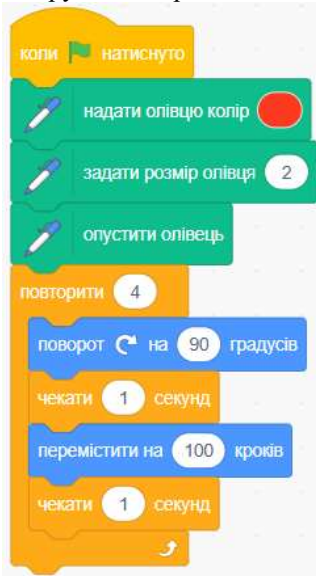
НАПРИКЛАД. Я додав(ла) розширення «Олівець», після чого в програмі з'явилися нові команди для малювання.

***ЗВЕРНІТЬ УВАГУ!** Без підключення розширення «Олівець» малювання в Scratch неможливе. Повторення – базовий елемент алгоритмічного мислення.*

2. Створення алгоритму малювання квадрата

1. Запустіть програму Scratch.
2. Оберіть героя із папки Animals.
3. Створіть алгоритм побудови геометричної фігури.
4. Перейдіть до блоку «Олівець», задайте його колір, розмір та опустіть його:
 - оберіть команду «Задати колір олівця» (колір обираєте на свій розсуд);
 - беріть команду «Задати розмір олівця» (наприклад, 2 розмір);
 - оберіть команду «Опустити олівець».
5. Оберіть кут на який герой буде повертатися (для того щоб намалювати квадрат потрібно 360^0 градусів поділити на кількість сторін. Ми отримаємо 90^0).
6. У блоці Рух оберіть команду Повернутися на 90^0 градусів.

- Обираємо команду Переміститись на 100 кроків.
- Щоб наш спрайт намалював квадрат треба застосувати циклічний алгоритм, в блоці Керувати обираємо команду Повторити 4 рази.



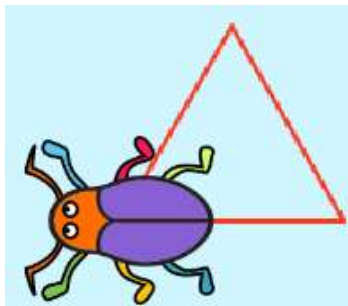
- Перевірте роботу вашого проєкту: перейдіть в режим презентації та натисніть на зелений прапорець.

ЗВЕРНІТЬ УВАГУ! Кількість повторень відповідає кількості сторін фігури.

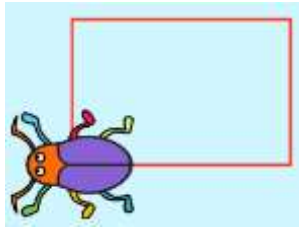
3. Створення власної програми з використанням алгоритму малювання геометричних фігур

Створіть три замкнені геометричні фігури (на вибір):

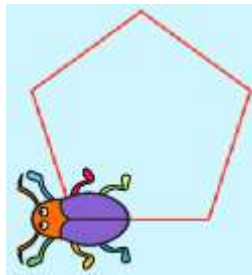
- рівносторонній трикутник;



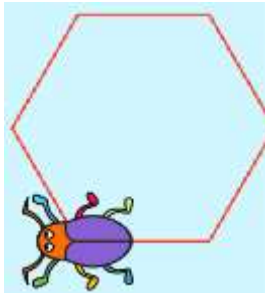
- прямокутник;



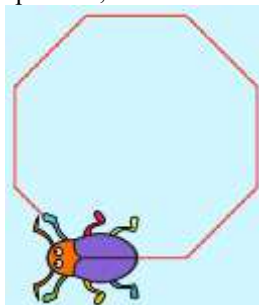
- п'ятикутник;



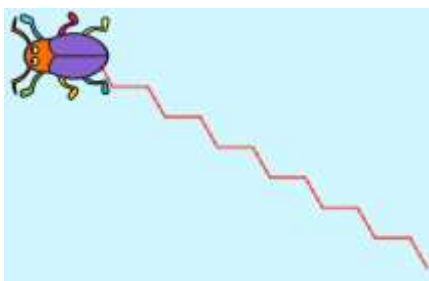
- шестикутник;



- ламана лінія з повторенням;



- простий орнамент із повторюваних елементів.



НАПРИКЛАД. Я обрав(ла) завдання з малювання рівностороннього трикутника. Для цього використав(ла) повторення трьох однакових дій: рух уперед і поворот на відповідний кут. У результаті спрайт намалював замкнену фігуру.

ЗВЕРНІТЬ УВАГУ!

- кількість повторень відповідає кількості сторін фігури;
- кут повороту залежить від форми фігури;
- довжина сторін визначається кількістю кроків руху;
- для прямокутника можна використати два вкладені повторення або послідовність різних сторін.
- не використовуйте зайвих команд, не пов'язаних із малюванням;
- алгоритм має бути простим і зрозумілим для учня початкової школи;
- головне – логіка повторення, а не складність фігури;
- такі експерименти розвивають уміння прогнозувати результат алгоритму.

4. Методичний аналіз завдання

1. Визначте, які навчальні дії виконує дитина.
2. Проаналізуйте, які уміння формуються.
3. Як учень може здогадатися, скільки разів потрібно повторити дію;
4. Як змінюється результат при зміні кута повороту або довжини сторони;
5. Які математичні уявлення формуються під час виконання завдання.

НАПРИКЛАД. Завдання формує уявлення про повторення, послідовність дій та зв'язок між алгоритмом і результатом.

ЗВЕРНІТЬ УВАГУ! Малювання фігур є ефективним способом інтеграції інформатики з математикою.

Форма звітності

1. Створити файл звіту ЛР_Scratch_8_Прізвище_Група.doc
2. Структура:
 - опис створеного алгоритму;

- скрипт малювання фігури;
- аналіз результату;
- методичний висновок.

Вимоги до оформлення

- 3–4 сторінки;
- Times New Roman, 14;
- міжрядковий інтервал 1,5;
- обов'язкові скріншоти:
 - підключеного розширення «Олівець»;
 - зібраного циклу;
 - намальованої фігури;
- скріншоти мають бути підписані;
- після скріншотів додайте покликання на створені скрипти.

Критерії оцінювання (5 балів)

- коректність алгоритму з повторенням – 2 бали;
- правильність малювання фігури – 1 бал;
- методичний аналіз – 1 бал;
- оформлення – 1 бал.

■ ЛАБОРАТОРНА РОБОТА №9

Створення та виконання алгоритмів з розгалуженням

Мета. Сформувані у студентів практичні вміння створювати алгоритми з розгалуженням у Scratch, аналізувати навчальні дії молодших школярів під час виконання завдань, що передбачають вибір дій залежно від умови.

Завдання лабораторної роботи:

- ознайомитися з поняттям розгалуження алгоритму;
- навчитися використовувати умовні блоки в Scratch;
- створити програму, у якій дії спрайта залежать від умови;
- поєднати умови з рухом, виглядом або звуком;
- здійснити методичний аналіз навчального завдання.

Матеріали та засоби:

- середовище Scratch (онлайн або офлайн-версія);
- персональний комп'ютер або ноутбук;
- текстовий редактор;
- засіб створення скріншотів.

Хід виконання лабораторної роботи

1. Ознайомлення з поняттям розгалуження

1. Згадайте, що означає ситуація вибору дії («якщо ... то ...»).
2. Запустіть Scratch і відкрийте новий проєкт.
3. Перейдіть до категорії «Керування».

НАПРИКЛАД. *Я зрозумів(ла), що розгалуження – це ситуація, коли програма виконує різні дії залежно від умови.*

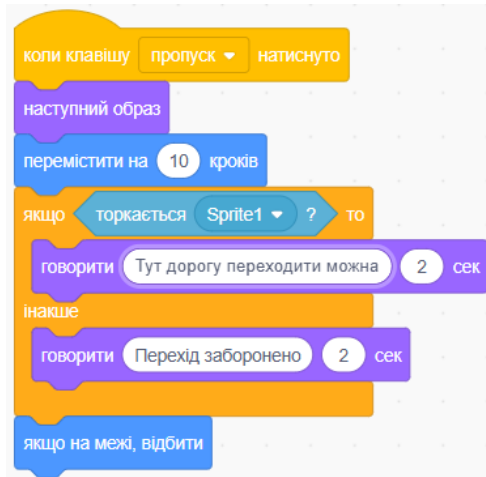
ЗВЕРНІТЬ УВАГУ! *Розгалуження – це ключовий елемент алгоритмічного мислення поряд із повторенням.*

2. Створення програмного коду використовуючи алгоритми з розгалуженням

1. Запустіть програму Scratch.
2. Намалуйте новий об'єкт (спрайт) зебру за допомогою вбудованого графічного редактора.
3. Оберіть нове тло нічного міста.
4. Відредагуйте тло доповнивши його дорогою використовуючи вбудований графічний редактор.
5. Розмістіть основний спрайт (за вашим вибором) і новий спрайт (зебра) на фоні нічного міста відповідно до малюнка (за допомогою їхнього перетягування ЛКМ)



6. Створюємо алгоритм з розгалуженням для вашого спрайта (Sprite1 в кодї це новий створений спрайт зебра).



7. Перевірте роботу вашого проекту: перейдіть в режим презентації та натисніть на зелений прапорець.

ЗВЕРНІТЬ УВАГУ! Форма блоку підказує його призначення – це важлива дидактична особливість Scratch.

3. Створення власного програмного коду

1. Запустіть програму.
2. Відкрийте попередній проект, або створіть новий.
3. Намалюйте та додайте до проекту третій спрайт (світлофор) у вбудованому графічному редакторі та розташуйте його біля зебри.
4. Відредагуйте скрипт таким чином, щоб основний спрайт доторкаючись до світлофора видавав повідомлення: «Можна перейти тільки на зелений».



6. Методичний аналіз завдання

1. Визначте, які навчальні дії виконує дитина.
2. Проаналізуйте, які уміння формуються.
3. Зробіть методичний висновок.

НАПРИКЛАД. Завдання формує розуміння умов, логічного вибору та причинно-наслідкових зв'язків.

ЗВЕРНІТЬ УВАГУ! Розгалуження є основою для створення ігор та інтерактивних проєктів.

7. Практичне завдання для самостійного виконання

Створіть власну програму в Scratch, у якій спрайт:

- реагує на **дотик до краю, або**
- реагує на **натискання клавіші, або**
- змінює поведінку залежно від умови (наприклад, положення на сцені).

У програмі обов'язково використайте блок **якщо** або **якщо – інакше**.

Форма звітності

1. Створити файл звіту **ЛР_Scratch_9_Прізвище_Група.doc**
2. Структура:
 - опис спрайтів;
 - опис програмних кодів;
 - аналіз взаємодії;
 - методичний висновок.

Вимоги до оформлення

- 3–4 сторінки;
- Times New Roman, 14;
- міжрядковий інтервал 1,5;
- обов'язкові скріншоти:
 - списку спрайтів;
 - скриптів кожного спрайта;
 - сцени з анімацією.
- скріншоти мають бути підписані;
- після скріншотів додайте покликання на створені скрипти.

Критерії оцінювання (5 балів)

- коректність розгалуження – 2 бала;
- логічність поведінки спрайта – 1 бал;
- методичний аналіз – 1 бал;
- оформлення – 1 бал.

■ ЛАБОРАТОРНА РОБОТА №10

Створення навчального мініпроєкту в Scratch

Мета. Узагальнити та систематизувати практичні вміння студентів зі створення програм у Scratch; сформувати здатність проєктувати цілісний навчальний продукт для початкової школи та здійснювати методичний аналіз створеного проєкту.

Завдання лабораторної роботи:

- створити завершений проєкт у середовищі Scratch;
- використати основні алгоритмічні конструкції (послідовність, повторення, розгалуження);
- забезпечити взаємодію кількох об'єктів;
- використати сцени, звук та/або олівець;
- здійснити методичний аналіз створеного проєкту.

Матеріали та засоби:

- середовище Scratch (онлайн або офлайн-версія);
- персональний комп'ютер або ноутбук;
- текстовий редактор;
- засіб створення скріншотів.

Загальні вимоги до підсумкового проєкту

У створеному проєкті обов'язково має бути:

1. Не менше двох спрайтів.
2. Щонайменше два фони (сцени).
3. Алгоритм з повторенням (повторити або завжди).
4. Алгоритм з розгалуженням (якщо, якщо – інакше).
5. Подієве керування (зелений прапорець, клавіша, дотик, повідомлення).
6. Елемент мультимедіа:
 - звук або
 - малювання олівцем.

Орієнтовні теми мініпроєкту (на вибір студента):

- «Подорож персонажів різними сценами»
- «Навчальна анімація з геометричними фігурами»
- «Інтерактивна історія для молодших школярів»
- «Гра з умовами та реакціями спрайтів»
- «Орнамент або малюнок з керуванням»

ЗВЕРНІТЬ УВАГУ! Тема має бути зрозумілою та дидактично доцільною для початкової школи.

Хід виконання лабораторної роботи

1. Проектування мініпроєкту

1. Оберіть тему проєкту.
2. Визначте:
 - кількість спрайтів;
 - кількість сцен;
 - основні дії кожного об'єкта.
3. Продумайте логіку роботи програми.

НАПРИКЛАД. Я запланував(ла) мініпроєкт з двома спрайтами та двома сценами. На першій сцені персонажі рухаються, а на другій – реагують на дії користувача.

ЗВЕРНІТЬ УВАГУ! Планування є ключовим етапом підсумкової роботи.

2. Реалізація проєкту в Scratch

1. Створіть новий проєкт.
2. Додайте спрайти та сцени.
3. Реалізуйте:
 - повторення;
 - розгалуження;
 - взаємодію об'єктів.
4. Перевірте коректність роботи програми.
5. За потреби відкоригуйте параметри.

3. Методичний аналіз створеного проєкту

У звіті проаналізуйте:

- які навчальні дії виконує дитина;
- які алгоритмічні уявлення формуються;
- на якому етапі уроку доцільно використати проєкт.

Форма звітності

1. Створити файл ЛР_Scratch_10_Прізвище_Група.doc
2. Орієнтовна структура звіту:
 - тема і короткий опис проєкту;
 - опис використаних алгоритмів;
 - скріншоти проєкту;
 - методичний висновок.

Вимоги до оформлення

- 3–4 сторінки;
- Times New Roman, 14;
- міжрядковий інтервал 1,5;
- обов'язкові скріншоти:
 - списку спрайтів;

- скриптів кожного спрайта;
 - сцени з анімацією.
- скріншоти мають бути підписані;
- після скріншотів додайте покликання на створені скрипти.

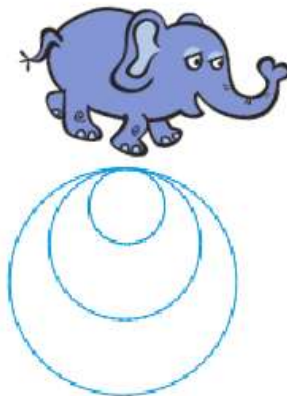
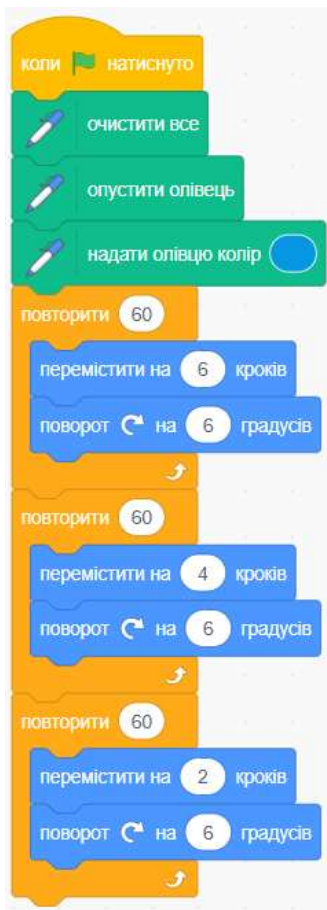
Критерії оцінювання

- повнота реалізації вимог – 2 бала;
- коректність алгоритмів – 1 бал;
- логічність і цілісність проекту – 1 бал;
- методичний аналіз і оформлення – 1 бал.

РОЗДІЛ III. ЗРАЗКИ ВПРАВ ДЛЯ ПОГЛИБЛЕНОГО ВИВЧЕННЯ

Вправа 1. «Вкладені кола»

Намалюйте три вкладених кола різних діаметрів певного кольору

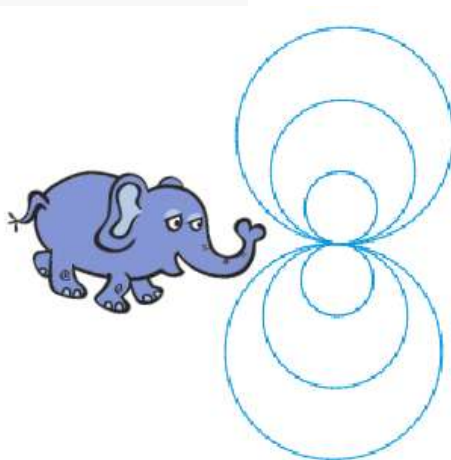


Вправа 2. «Симетричні вкладені кола»

Намалюйте три вкладених кола різних діаметрів певного кольору та відобразіть цей малюнок симетрично.

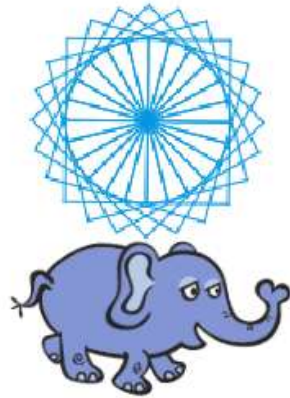


The image shows two columns of Scratch code blocks. The left column starts with a 'коли натиснуто' (when clicked) event block, followed by 'очистити все' (clear all), 'опустити олівця' (erase), and 'надати олівцю колір' (set pen color to blue). It then contains three 'повторити' (repeat) loops, each with a count of 60. The first loop has 'перемістити на 6 кроків' (move 6 steps) and 'поверот на 6 градусів' (turn 6 degrees). The second loop has 'перемістити на 4 кроків' (move 4 steps) and 'поверот на 6 градусів' (turn 6 degrees). The third loop has 'перемістити на 2 кроків' (move 2 steps) and 'поверот на 6 градусів' (turn 6 degrees). The right column starts with a 'коли я отримую повідомлення' (when I receive a message) event block, followed by three 'повторити' (repeat) loops, each with a count of 60. The first loop has 'поверот на 6 градусів' (turn 6 degrees) and 'перемістити на 6 кроків' (move 6 steps). The second loop has 'поверот на 6 градусів' (turn 6 degrees) and 'перемістити на 4 кроків' (move 4 steps). The third loop has 'поверот на 6 градусів' (turn 6 degrees) and 'перемістити на 2 кроків' (move 2 steps). Both columns end with 'оновити повідомлення' (update message) and 'чекати' (wait).



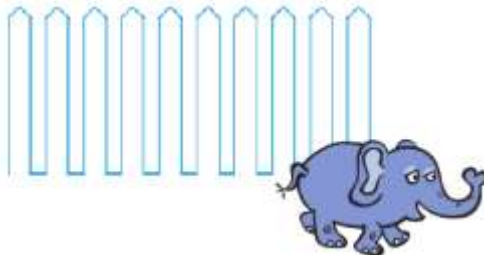
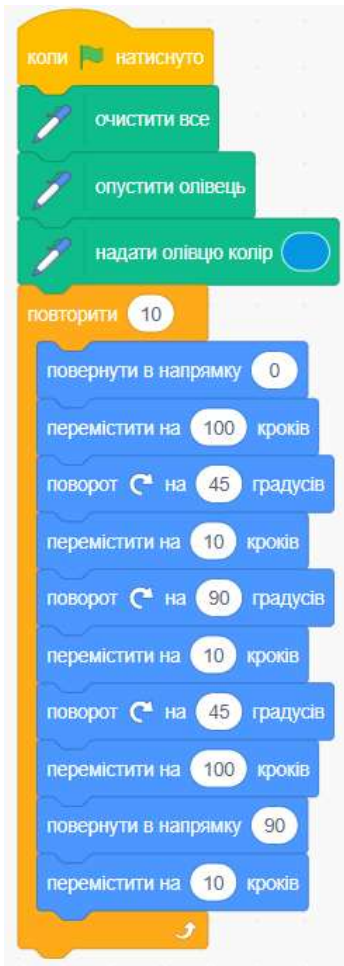
Вправа 3. «24 квадрати»

Намалюйте 24 квадрати із стороною у 40 кроків. При малюванні кожного наступного квадрата потрібно повернути вправо на 15° градусів.



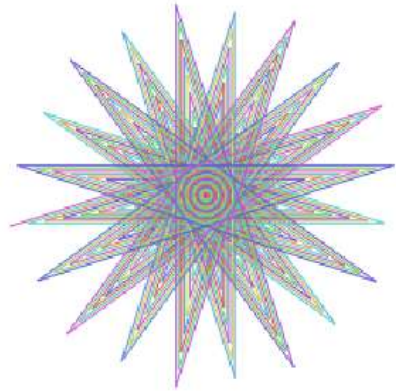
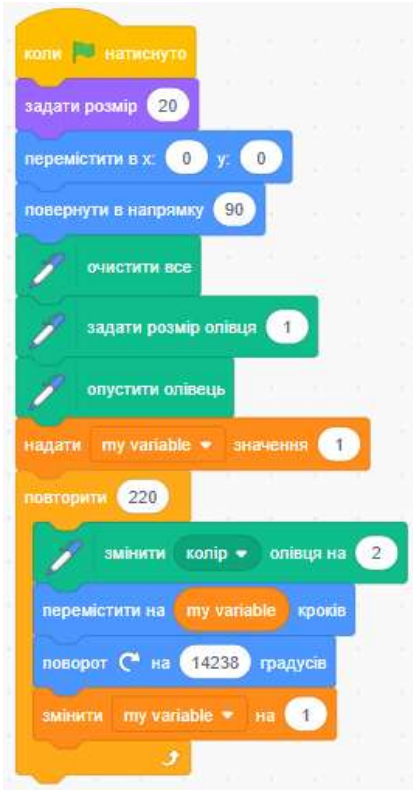
Вправа 4. «Паркан»

Намалюйте паркан з 10 вузьких довгих дощечок (100 кроків).

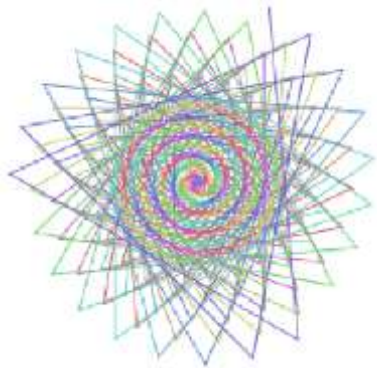


Вправа 5. «Райдужна зірка»

Намалюйте візерунок, яка складається з певної кількості відрізків, розташованих під кутом 140° .



Проексperimentуйте з кількістю повторень та градусу кута повороту.



Вправа 6. «Акваріум»

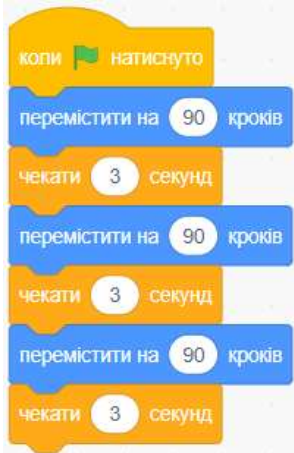
У цьому завданні основним є забезпечення спонтанного руху риб у акваріумі. Анімацію того як риба пливе можна реалізувати створивши для неї два образи. Послідовна зміна образів дає ефект того, що риба пливе. Спонтанність руху зможе забезпечити такий скрипт.



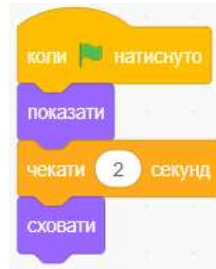
Вправа 7. «Полювання на гриби»

Намалюйте об'єкт Гриб і скопіюйте його двічі. Отримуємо чотири об'єкта: Кіт і три гриба. Створити анімацію, під час якої Кіт рухається зліва на право по 100 кроків. Підходячи до кожного гриба Кіт зупиняється, чекає 5 секунд і збирає гриба (гриба сховати). Далі рухається до слідуячого гриба і дії повторюються.

Скрипт для спрайта (Кіт)



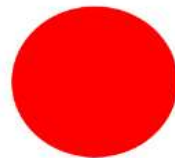
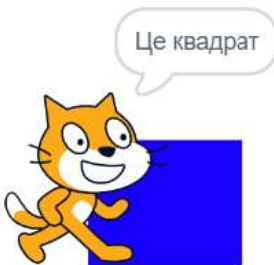
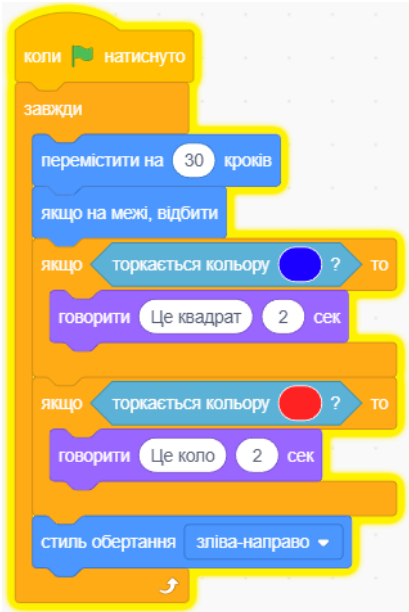
Скрипт для гриба



Вправа 8. «Геометричні фігури»

Намалюйте на тлі два об'єкти: синій квадрат та червоне коло. Складіть проект, у якому виконавець Кит рухається по сцені зліва-направо та промовляє слово «Це квадрат» або «Це коло», в залежності від того, до якого кольору він доторкнеться.

Зверніть увагу! Параметри кольорів в скрипті та тлі повинні бути однаковими.



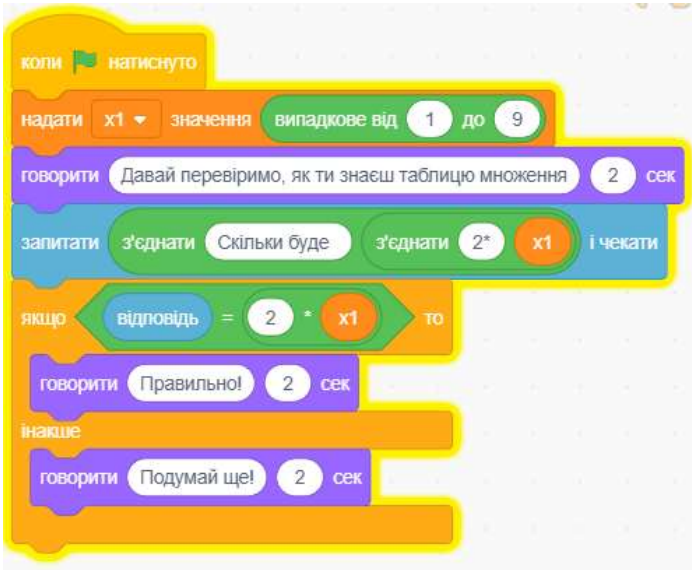
Вправа 9. «Скільки буде 2+3?»

Створіть проект, у якому виконавець запитує «Скільки буде 2+3?». Якщо користувач вводить вірну відповідь, то Спрайт відповідає «Молодець!», якщо ні – то говорить «Не вірно, подумай ще!».



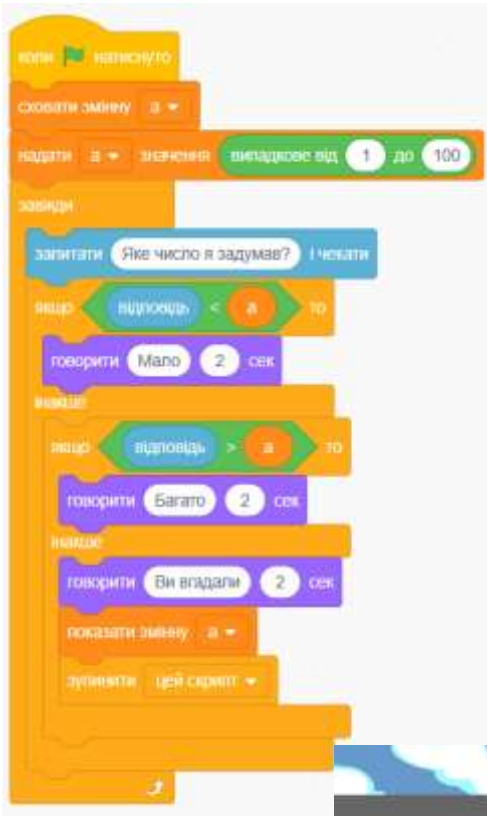
Вправа 10. «Таблиця множення»

Створіть проект, у якому спрайт говорить «Давай перевіримо, як ти знаєш таблицю множення» і з'являється випадковий приклад. Користувач вводить відповідь і отримує схвальне «Правильно», якщо відповідь вірна, і навпаки, якщо не вірна, то «Подумай ще!».



Вправа 11. «Вгадай число»

Спрайт випадковим чином задумує число у межах від 0 до 100, а гравець вгадує його. На пропозиції гравця Спрайт повідомляє: «Мало», «Багато» чи «Ви вгадали!!!» залежно від взаємного розташування числа-згадки і випадкового числа. Маємо циклічний алгоритм доки не буде отримано правильну відповідь.



СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бондарчук Ж.А. Збірка творчих завдань та вправ в середовищі програмування Scratch. Методична розробка. Луцьк, 2017. 44 с.
2. Бурчак С.О., Прокопенко Ю.В., Тесленко М.Ю. Мова програмування SCRATCH навчально-методичний посібник. Суми : Видавничий дім «Ельдорадо», 2015. 128 с.
3. Середовище програмування Scratch 4. URL:www.scratch.ucoz.net (дата звернення 12.01.2026).
4. Спільнота Scratch. URL:www.scratch.mit.edu (дата звернення 12.01.2026).
5. Федорович Л. В. Програмування у середовищі Scrath. Сновидовичі, 2021. 84 с.
6. Чоп С. Ф. Збірник практичних робіт у середовищі програмування Скретч з інформатики для 5 - 7 класів (збірник практичних завдань). Бузакі: ЗЗСО «Бузаківський ліцей», 2022. 104 с.

Електронне видання

Литвинов Андрій Сергійович

**ВИКОРИСТАННЯ СЕРЕДОВИЩА SCRATCH У ПРОЦЕСІ
НАВЧАННЯ ІНФОРМАТИЧНОЇ ОСВІТНЬОЇ ГАЛУЗІ**

навчально-методичний посібник
для здобувачів вищої освіти

Підп. до розповсюдження 25.02.2026.
Формат 60x84/16. Умов. друк. арк. 4,01. Зам. №3549
Облік.-вид. арк. 2,49. Папір офсетний. Гарнітура Таймс.
Видавництво Глухівського національного педагогічного
університету імені Олександра Довженка
41400, м. Глухів, Сумська обл., вул. Київська, 24
тел/факс (05444) 2-33-06.
Свідоцтво суб'єкта видавничої справи СМв №046 від 16 червня 2014 року