Сарненський педагогічний коледж

Рівненського державного гуманітарного університету

Циклова комісія фізико-математичних дисциплін

Курсова робота

з методики навчання інформатики

на тему**:** «Методичні основи ознайомлення молодших школярів з побудовою скриптів на базі циклічних алгоритмів»

Студентки **IV** курсу **А-42** групи

Напрям підготовки **0101**

Педагогічна освіта

спеціальність 013

Початкова освіта

Потапчук Наталії Валеріївни

Керівник: викладач методики

навчання інформатики

Гожий Володимир Миколайович

Члени комісії:

_____ **В.В. Онищук**

_____ **В. М. Гожий**

_____ **К. Г. Михальська**

_____ **О. А. Лавор**

м. Сарни 2020

ЗМІСТ

Вступ

Процес інформатизації суспільства стає все більш динамічним і висуває нові вимоги до виховання і навчання учнів.

Сучасне суспільство зацікавлене в тому, щоб його громадяни були здатні самостійно, активно діяти, приймати рішення, гнучко адаптуватися до мінливих умов життя. Сучасна школа повинна створити умови для формування такої особистості.

Нова українська школа – це зовсім нова ключова реформа Міністерства освіти та науки України, і працювати в ній – це обов’язок кожної особистості, яка бере безпосередню чи опосередковану участь у даному процесі змін. Головна мета НУШ – створити школу, в якій буде приємно навчатися і яка даватиме учням не тільки знання, як це відбувається зараз, а й уміння застосовувати їх у повсякденному житті.

І це завдання не стільки змісту освіти, скільки використовуваних технологій навчання, до яких і відноситься знання та вміння алгоритмізації.

Дуже мало людей стане професійними письменниками, але писати й читати повинен вміти кожен. Те ж саме можна сказати і про програмування.

Вже зараз з'являється все більше професій на стику програмування, математики, фізики, біоінженерії, економіки. Школа має на меті підготувати дітей до майбутнього, в якому вони будуть жити. Учням подобається проводити час з новими технологіями, але частіше їх можна застати за такими заняттями, як ігри, спілкування в чатах з друзями, пошук в інтернеті та іншим активним користуванням. При цьому вони зовсім не орієнтовані на свідоме використання цих технологій та самореалізацію з їх допомогою.

Актуальність. Програмування - один з найважливіших навичок для сучасної людини. Заняття програмуванням не тільки забезпечить дитині високооплачувану роботу в майбутньому, а й підвищить його успіхи в математиці зараз, адже результати розроблених ним алгоритмів можна буде побачити на практиці, що ймовірно розвине інтерес до точних наук.

На чому ґрунтується програмування? Якщо задуматися про те, щоб почати у школі навчати дітей програмування, то напевно потрібно шукати відповідь на наступне питання: яку мову краще всього почати вивчати?

Але, тим не менш, є невелика кількість мов програмування та програм для роботи з ними, що можна застосовувати у навчальному процесі. Звичайно все залежить від вікової категорії учнів, профілю навчання та технічних можливостей школи. Але змістова лінія «Алгоритмізація та програмування» пронизаний весь шкільний курс вивчення інформатики.

У шкільних підручниках приділяється значна увага основним прийомам роботи в середовищі Scratch, а саме складанню алгоритмів для виконавця [6]. Більш детально можливості програмування в Scratch висвітлюють у своїх роботах такі вчені, як Є. Д. Патаракін, Т. Є. Сорокіна [9]. В літературі представлено різні напрямки роботи у середовищі Scratch [10], [11].

Аналіз публікацій в іноземних виданнях свідчить про накопичення значного досвіду щодо впровадження середовища програмування Scratch у навчальний процес, а саме з цієї тематики розроблені курси в Гарвардському університеті, Каліфорнійському університеті у Берклі, коледжі Нью-Джерсі інших [1, 2, 3]. Автори статті показують як зацікавити студентів, які вивчають комп'ютерні науки, та досвідчених програмістів до роботи в Scratch. До речі, учасникам навчання курсу CS50 Гарвардського університету, який набув широкого розголосу в інтернет-спільноті, на початковому етапі пропонувалось виконати завдання, реалізувавши його саме в середовищі Scratch. Проходження цього курсу є безкоштовним для всіх бажаючих отримати ґрунтовні знання з основ програмування

Метою даної роботи є: вивчення теоретичних та методичних основ створення циклічних алгоритмів на базі скриптів.

Відповідно до поставленої мети, виокремлено наступні **завдання:**

1. Опрацювати літературу та систематизувати теоретичні основи вивчення алгоритмів у шкільному курсі інформатики;

2. Здійснити класифікацію форм представлення, видів та властивостей алгоритмів;
3. Здійснити аналіз пропедевтики ознайомлення з циклічними алгоритмами та методами їх реалізації;
4. Вибрати та проаналізувати основні програмні середовища для візуалізації алгоритмів та кращого засвоєння у молодшій школі;
5. Розробити дидактичні матеріали для організації практичних робіт при вивченні циклічних алгоритмів.

Об'єктом розгляду даної роботи є вивчення методів побудови скриптів на базі циклічного алгоритму у середовищі програмування Scratch. Саме це середовище вважається одним з найлегших мов програмування для дітей, починаючи з 2 класу.

Предметом дослідження є методика використання та виконання скриптів при вивченні циклічних алгоритмів та їх побудови.

У процесі дослідження використано такі методи науково-педагогічного дослідження: аналіз, синтез, узагальнення, систематизація методичних матеріалів з проблеми використання скриптів для вивчення та побудови циклічних алгоритмів і педагогічне спостереження.

Курсова робота складається із вступу, двох розділів, висновку, переліку літератури та додатків.

РОЗДІЛ 1. Теоретичні основи вивчення циклічних алгоритмів у шкільному курсі інформатики.

1.1. Введення основних понять розділу «Алгоритми та виконавці».

Інформатизація початкового утворення на сучасному етапі є актуальним соціально-затребуваним процесом, найважливішим елементом парадигми початкової освіти, що змінюється. Навчальна програма Нової Української школи не декларує ідею початку вивчення інформатики 1 вересня у 1 класі, але тенденції зниження стартового віку в навчанні інформатиці школярів реалізуються сьогодні не лише в численних наукових дослідженнях, але і в керівних методичних та адміністративних документах.

Сучасні діти з раннього віку зустрічаються з новітніми технологіями і вже в дошкільному етапі можуть опановувати деякі з них – вони з раннього дитинства «дружать з комп'ютером», до школи вже досить впевнено запускають ігри. Але ті психологічні особливості, які властиві цьому віку дозволяють зробити великий крок в розвитку логіко-алгоритмічного мислення при навчанні у початкових класах [12].

Оволодівши необхідними знаннями та навичками в області алгоритмізації та програмування, у людини з'являється можливість створювати власні та вдосконалювати існуючи доробки у сфері інформаційних технологій. Виходячи з означення інноваційного потенціалу, алгоритмізація та програмування може використовуватись у навчально-виховному процесі як один із засобів для підвищення інноваційного потенціалу особистості як вчителя, так і учня. За новою програмою вивчення алгоритмізації та програмування повертається у шкільний курс інформатики і розглядається у кожній паралелі, починаючи з другого класу.

Згідно з новою програмою НУШ, змістова лінія алгоритми вивчається у кожному класі початкової ланки і має на меті розвиток логічного, алгоритмічного, творчого та об'єктно-орієнтованого мислення учнів [15].

Уже в 2 класі учень визначає послідовність кроків для виконавців, знаходить помилки у алгоритмах та визначає результат виконання лінійного алгоритму побудови простого геометричного зображення, створює малюнок за лінійним алгоритмом.

У 3 та 4 класах змістова лінія «Алгоритми» спрямована на розвиток розуміння поняття виконавця, його середовища, команди, системи команд виконавця алгоритму, основних алгоритмічних структур, зокрема, слідування, розгалуження та повторення; умінь виконувати готові алгоритми, а також складати прості алгоритми для виконавців, які працюють у певному зрозумілому для відповідної вікової категорії середовищі, використовуючи просту систему їхніх команд; навичок шукати помилки в послідовності команд, аналізувати зміст завдань на складання алгоритму для виконавців; вміння розв’язувати задачі з повсякденного життя, застосовуючи алгоритмічний підхід: уміння планувати послідовність дій для досягнення мети, передбачати можливі наслідки [16].

У третьому класі вивчаються команди та виконавці, алгоритми, способи подання алгоритму, а також логічні висловлювання.

А у четвертому середовище виконання алгоритму, алгоритми з розгалуженням, складання алгоритмів з повторенням. Також тут створюються програмовані проекти, зокрема анімаційні історії, ігри та стратегії перемоги [5].

Тому у даному пункті спробуємо визначити основні поняття, що потрібні дитині. У другому класі йде вивчення наступних понять:

Команди – це речення, які спонукають до дії.

Виконавець – той, хто виконує команди. За виконавця уже беруть Рудого кота, який виконує різні команди.

Система команд виконавця – це команди, які може виконати виконавець.

Після цього йде вивчення поняття алгоритму.

Алгоритм – це послідовність команд.

У третьому класі розглядається поняття *комп'ютерної програми* – це інструкції для комп'ютера, що забезпечують розв'язання певних задач.

Середовище – це комплекс програм, який містить виконавців зі своїм набором команд.

У програмі рекомендують використовувати навчальне середовище Скретч, тому у підручнику 3 класу описано наступні означення:

Скретч – це середовище програмування, де можна створювати власні програми, ігри, цікаві історії, мультфільми та багато іншого.

Спрайт – це виконавець у середовищі Скретч.

Скрипт – набір команд, що повинен виконати виконавець.

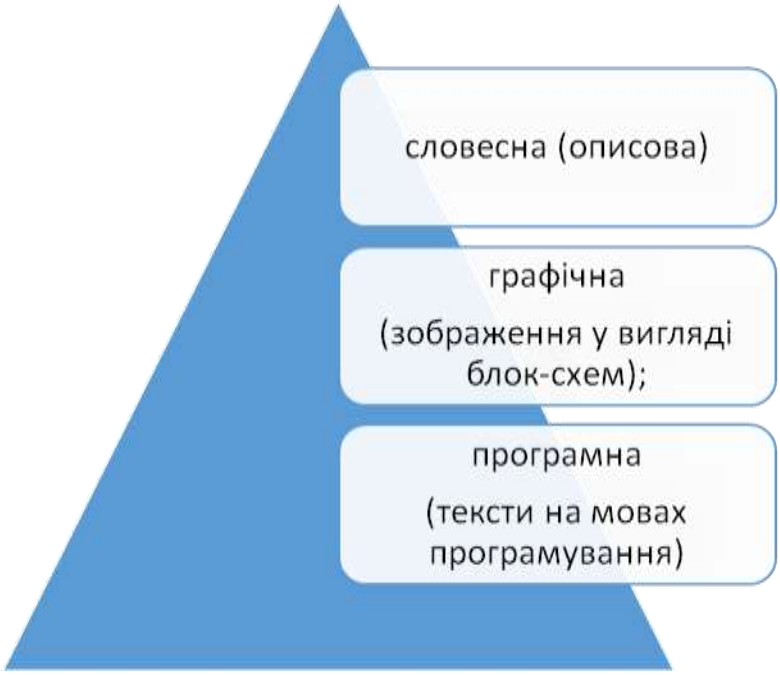
Пізніше вводиться поняття видів алгоритму, форм представлення алгоритму та інші поняття, пов'язані з даною темою.

Отже, при вивченні основ алгоритмізації формується системно-інформаційна картина світу, формуються навички виділення об'єктів, процесів та явищ, розуміння їх структури, і що саме головне, виробляється уміння самостійно ставити і вирішувати завдання. Тому вибір середовища програмування для молодших школярів та методу вивчення алгоритмізації та основ програмування є відповідальним кроком, від якого залежить якість майбутніх інформатичних знань у випускників як молодшої, так і старшої школи.

1.2. Форми представлення алгоритму.

Розглянемо наступні основні поняття вивчення теми алгоритмізації.

На практиці найбільш поширені такі форми представлення алгоритмів:



Смарт-схема 1.1

У шкільному курсі вивчення технологій використовуються усі три форми для відображення алгоритмів. Діти розпочинають вивчення з словесної форми, складають алгоритми з допомогою малюнків. Але для пояснення безпосередньо видів алгоритмів, використовуються блок-схеми. Щодо мов програмування, то безпосередньо кодів дитина не пише, тому що пропонується блочне візуальне середовище [17].

Словесний спосіб – є опис послідовних етапів обробки даних і задається в довільному викладі на природній мові.

Приклад. Записати в словесній формі алгоритм знаходження найбільшого спільного дільника (НСД) двох натуральних чисел. Алгоритм може бути таким:

1) Задати два натуральні числа;

2) Якщо числа рівні, то взяти будь-яке з них за відповідь і зупинитися, інакше продовжувати виконання алгоритму;

3) Визначити більше з чисел;

4) Замінити більше з чисел різницею більшого і меншого числа;

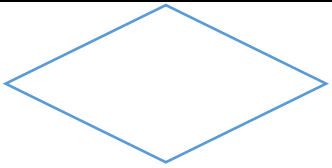
5) Продовжувати алгоритм з кроку 2).

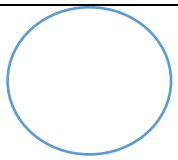
Частіше застосовується **графічний спосіб** запису алгоритмів як наочніший і компактніший. Алгоритм в цьому випадку зображується у вигляді послідовності зв'язаних між собою функціональних блоків, кожен з яких відповідає виконанню одного або декількох елементарних дій (операцій). Таке графічне представлення алгоритму називається його блок-схемою.

У блок-схемі кожній типовій дії відповідає геометрична фігура у вигляді блокового символу. Форми блоків, їх призначення, розміри і способи з'єднання між собою регламентовані стандартом.

Графічні символи, що найбільш часто вживаються, представлені в таблиці 1.1:

Таблиця 1.1

Назва	Блок-схема	Призначення
Початок Кінець		Початок/кінець алгоритму, переривання процесу обробки даних (для початку — тільки вихід, для кінця — тільки вхід)
Дані		Визначення даних
Процес		Виконання операції або групи операцій, у результаті яких змінюється значення, форма подання або розташування даних (один вхід, один вихід)
Розгалуження		Вибір напрямку виконання алгоритму залежно від деяких змінних умов (один вхід та тільки два виходи — так і ні)

Визначений процес		Програма, стандартна підпрограма
З’єднувачі		Вказівка зв'язку між перерваними лініями, що з'єднують блоки
Міжсторінковий з’єднувач		Вказівка зв'язку між перерваними лініями, що з'єднують блоки, розміщені на різних сторінках

Представлення алгоритму у вигляді блок-схеми є проміжним, оскільки у такому вигляді алгоритм не може бути виконаний комп’ютером. Складання блок-схеми алгоритму є важливим і в більшості випадків необхідним етапом рішення задачі, що значно полегшує процес складання програм.

Алгоритм, записаний на мові програмування, називається **програмою** для ПК. У таблиці 1.2 надано приклади написання програм у різних мовах програмування.

Таблиця 1.2

Назва мови програмування	Приклад тексту
Pascal	program HelloWorld; begin writeln('Привіт світ!'); end.
Python	>>> print('Hello, world!') Hello, world! >>>
C++	// first_program.cpp: определяет точку входа для консольного приложения. // <DEN><программа печатающая текст><15:40><23.07.2010> #include "stdafx.h" #include <iostream>

Отже, як бачимо, у процесі вивчення інформатики, учень вивчає усі форми представлення алгоритму, починаючи з словесного запису і середніх класах розпочинає вивчення чистих мов програмування.

1.3. Види алгоритмів.

Щодо видів алгоритмів, то вони розглядаються у 4 класі. Тобто другий та третій класи вивчають лінійні алгоритми, а у четвертому класі учні знайомляться зі структурою розгалуження та повторення.

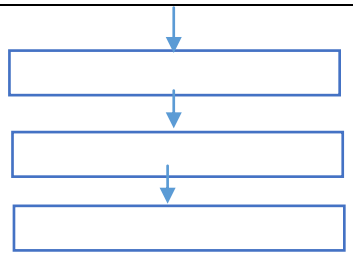
Отже, у даному пункті доцільно розглянути види алгоритмів.

Смарт-схема 1.2

Види алгоритмів розрізняють зазвичай не за складністю виконуваних дій, не за їхньою кількістю, а за складністю організації (або управлінням, за логічною конструкцією) алгоритмічного процесу.

Алгоритми найпростішого виду - **лінійні**. Це такі алгоритми, в яких дії виконуються послідовно, одна за одною. Кожна дія лінійного алгоритму обов'язково виконується, і виконується тільки один раз [14].

Таблиця 1.3

Словесна форма	Блок-схема
1. Колобок втік від бабусі з дідусем. 2. Колобок зустрів зайця. Оминув його. 3. Колобок зустрів вовка. І його оминув. 4. Колобок зустрів ведмедя. Не повірите - і ведмедя здихався. 5. Колобок зустрів лисицю. Вона його з'їла.	

Складнішими за організацією є алгоритми, в яких треба не просто виконувати всі підряд задані дії, а приймати рішення, які саме дії виконувати.

Отже, алгоритм, в якому та чи інша серія команд реалізується в залежності від виконання заданої умови, називається алгоритмом з **розгалуженням**.

Розрізняють повну і коротку форму розгалуження. В короткій формі при невиконанні умови ніякі дії не передбачаються. Повну форму розгалуження можна прочитати так: "Якщо умова виконується, то виконати дію 1, інакше виконати дію 2". А коротку - так: "Якщо умова виконується, то виконати дію".

Таблиця 1.4

Словесна форма	Блок-схема
ЯКЩО прагнеш бути здоровий, ТО загартовуйся, ІНАКШЕ валяйся весь день на дивані; ЯКЩО низько ластівки літають, ТО буде дощ, ІНАКШЕ дощу не буде; ЯКЩО уроки виучені, ТО йди гуляти, ІНАКШЕ вчи уроки.	

Третій вид алгоритмів - такі, котрі передбачають неодноразове (але скінченне) виконання певної дії (або кількох дій). Це циклічні алгоритми. Дії, які мають повторюватись, називаються тілом циклу. Умова, яка визначає кількість повторень циклу, називається умовою циклу. Зазначена команда/команди виконується Доки наведений логічний вираз справджується.

Розглянемо приклад з математики.

Натуральне число називають простим, якщо воно має тільки два дільники: одиницю й саме це число .

2, 3, 5, 7 — прості числа; 4, 6, 8 — ні.

В III столітті до нашої ери грецький математик Ератосфен запропонував наступний алгоритм для знаходження всіх простих чисел, менших заданого числа *n;*

1) виписати всі натуральні числа від 1 до n;

2) викреслити 1;

3) підкреслити найменше з невідмічених чисел;

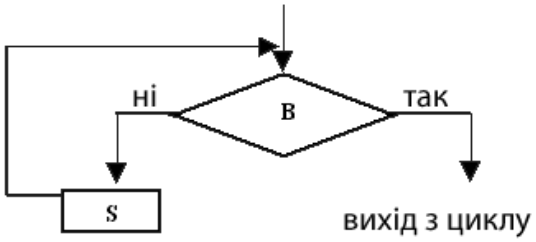
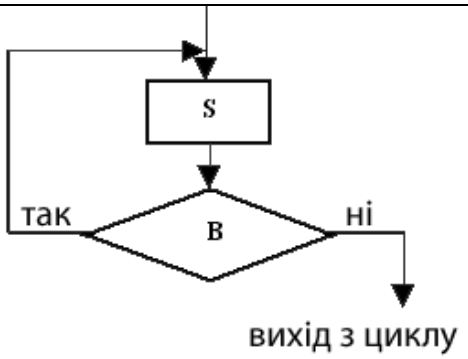
4) викреслити всі числа, кратні підкресленому на попередньому кроці;

5) якщо в списку є невідмічені числа, то перейти до кроку 3, а якщо ні, то всі підкреслені числа — прості.

Це циклічний алгоритм. При його виконанні повторення кроків 3-5 відбувається, поки у вихідному списку залишаються невідмічені числа.

Розглянемо таблицю подання словесної форми циклу з передумовою та післяумовою.

Таблиця 1.5

Словесна форма	Блок-схема
Перевіряємо умову - повне відро картоплі – НІ – беремо картоплю Чистимо, ставимо у чисту каструлю.	
Беремо картоплю. Чистимо. Ставимо у каструлю. Перевіряємо умову – не повне відро картоплі – ТАК.	

Отже, усі види алгоритмів у відповідній мірі вивчаються у початкових класах. Прийшовши у середню школу, учень вже має базу візуального програмування складання різних видів алгоритмів, яка йому потім знадобиться для вивчення справжньої мови програмування.

1.4. Властивості алгоритмів.

Властивості алгоритмів розглядають у середніх класах, коли учень має базове розуміння поняття алгоритму, його видів та форм представлення [4].

Розглянемо основні властивості алгоритмів.

Смарт-схема 1.1

Будь-який алгоритм подає розв'язування задачі як послідовність відокремлених простих дій. Вони виконуються почергово, одна за одною, в усталеному порядку. Тільки після виконання однієї дії можна перейти до наступної. Така розчленованість процесу виконання алгоритму називається **дискретністю** (від лат. *discretus* - розділений, переривчатий).

За скінченну кількість кроків алгоритм має приводити до здобуття потрібного результату або до досягнення потрібної мети. Цю властивість алгоритму називають **скінченністю** або **результативністю**.

Алгоритм має складатися з команд, які входять до системи команд його виконавця, адже це забезпечує **зрозумілість** алгоритму.

Разом із заданою сукупністю вхідних даних, алгоритм цілком визначає дії виконавця та їх послідовність при розв'язанні задачі. Кожна команда алгоритму однозначно встановлює, що треба робити на даному кроці та до

якої команди перейти на наступному. Будь-які розбіжності в тлумаченні дій, приписуваних командою, виключаються. Не може виникнути й потреби у будь-якій додатковій інформації ззовні або у прийнятті рішень самим виконавцем. Цю властивість алгоритму називають **визначеністю** або **детермінованістю** (від лат. *Determinare* - визначати).

Переважно один і той самий алгоритм має змогу виконати не один, а декілька виконавців - і отримати однакові результати. Ця властивість називається **формальністю** або певністю.

Зазвичай алгоритм складається так, щоб його можна було застосовувати до розв'язання не однієї конкретної задачі, а всіх задач певного типу, які відрізняються між собою вхідними даними. Цю властивість алгоритму називається **масовістю**. Вона не означає, що вхідним даним можна надавати довільних значень. Вони звичайно обмежені змістом задачі, застосованим способом її розв'язання тощо. Для кожного алгоритму встановлюється відповідна область припустимих значень вхідних даних, або, інакше кажучи, область його застосування.

Учень повинен знати та розуміти властивості алгоритму, але це засвоєння відбувається уже у середній школі, де учень розуміє поняття «властивість», а тоді уже поняття «властивість алгоритму».

1.5. Реалізація алгоритмів у різних програмних середовищах.

Вивчення алгоритмізації у шкільній інформатиці може мати два цільових напрями:

- ✓ розвивальний напрям: розвиток алгоритмічного мислення учнів;
- ✓ програмістський напрям: вивчення технології створення програм.

Останній напрям можна розділити на два цільових аспекти:

Перший аспект пов'язаний з посиленням фундаментальної компоненти курсу інформатики. Учням дається уявлення про те, що таке мови програмування, що представляє собою програма мовою програмування високого рівня, та як вона створюється в середовищі системи програмування.

Другий аспект носить профорієнтаційний характер. Вивчення програмування в рамках шкільного курсу дозволяє учням випробувати свої здібності до такого роду діяльності і, при бажанні, вибрати у майбутньому відповідний професійний шлях.

Алгоритмізація у школі відповідає методу структурного програмування і є підготовчим етапом до вивчення об’єктно-орієнтованого програмування, актуального на сучасному етапі розвитку програмування.

Стандартна програма вивчення основ алгоритмізації передбачає наступну послідовність тем:

- ✓ складання лінійних алгоритмів;
- ✓ складання циклічних алгоритмів;
- ✓ використання розгалужень в алгоритмах;
- ✓ опис і використання допоміжних алгоритмів.

На початковому етапі вивчення певної структури алгоритму доцільно використовувати блок-схеми, які ми згадували раніше, які наочно демонструють базові структури алгоритмів та дають можливість сформувати правильну уяву про механізм роботи кожної із них. Ефективним засобом підвищення рівня сприйняття теорії алгоритмізації є використання

середовищ з виконавцями, що наочно представляють механізм виконання алгоритмів та їх базових структур.

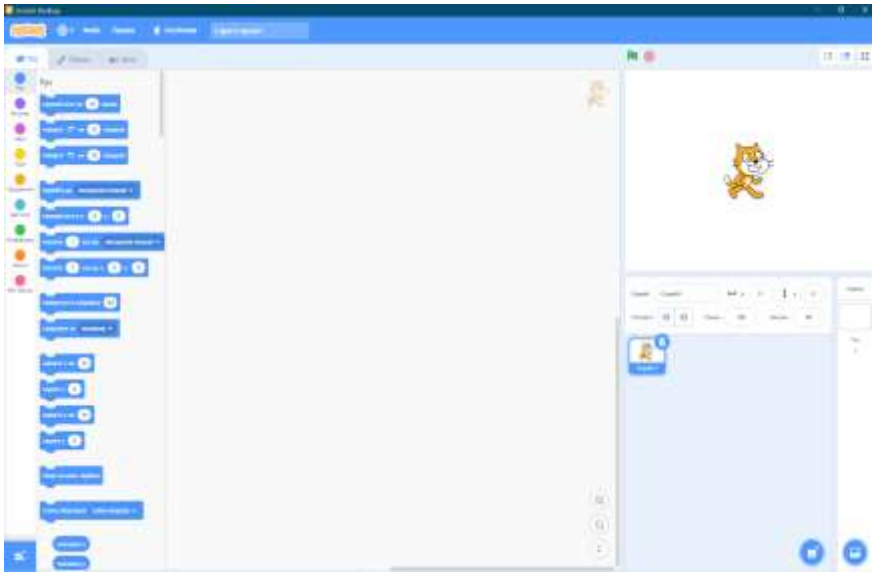
Навчання програмуванню можна розпочати ще з дошкільного навчального закладу, звісно з відповідним обладнанням. Для маленьких діток підійде цікавий додаток на планшетах **ScratchJr**. Він зрозумілий, яскравий, цікавий. У ньому можна реалізувати багато проектів. Виконавцями є різні герої, а команди, які вони виконують реалізують усі форми та види алгоритмів [18].

Мал.1.1. Інтерфейс програми ScratchJR

Програма шкільного курсу інформатики в якості навчального середовища програмуванню пропонує використовувати **Scratch** [8]. За програмою НУШ, навчання у цій програмі розпочинається уже з 2 класу. Це середовище об’єктно-орієнтованого візуального програмування, яке надає можливості створювати комп’ютерні анімації, мультимедійні презентації, інтерактивні матеріали у вигляді історій та ігор, моделі та ін. Scratch є вільно розповсюджуваною в навчальних цілях програмою, яку можна завантажити з офіційного сайту розробників.

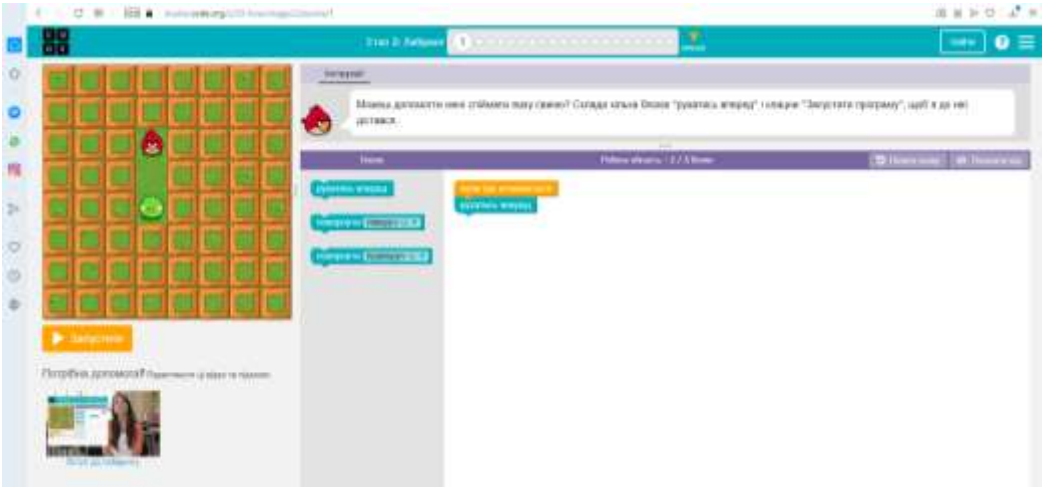
Програмування у зазначеному середовищі відбувається наступним чином: учні «збирають» програму (скрипт) із блоків, також деякі блоки

можна вбудовувати один в одного, виключаючи можливість виникнення синтаксичних помилок. У середовищі Scratch виконавцями є спрайти і сцена. Об’єкт, який пов’язують із певним зображенням та набором змінних і скриптів для визначення його поведінки називають спрайтом.



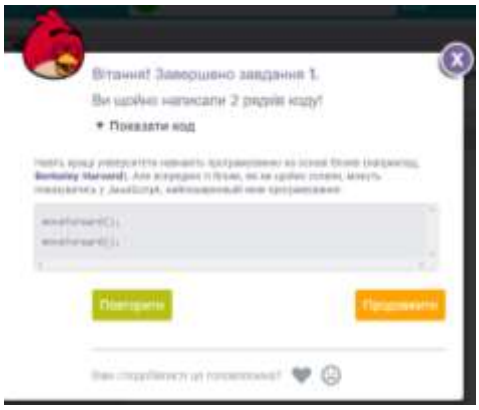
Мал.1.2. Інтерфейс програми Scratch

Багато педагогів схильні до думки, що більш ефективним є використання сайту **code.org**, який створений саме з навчальною метою. На сайті пропонуються завдання, розбиті на курси. Кожен курс розрахований на певний рівень підготовки учня та його вік [19].



Мал. 1.3. Онлайн сервіс Code

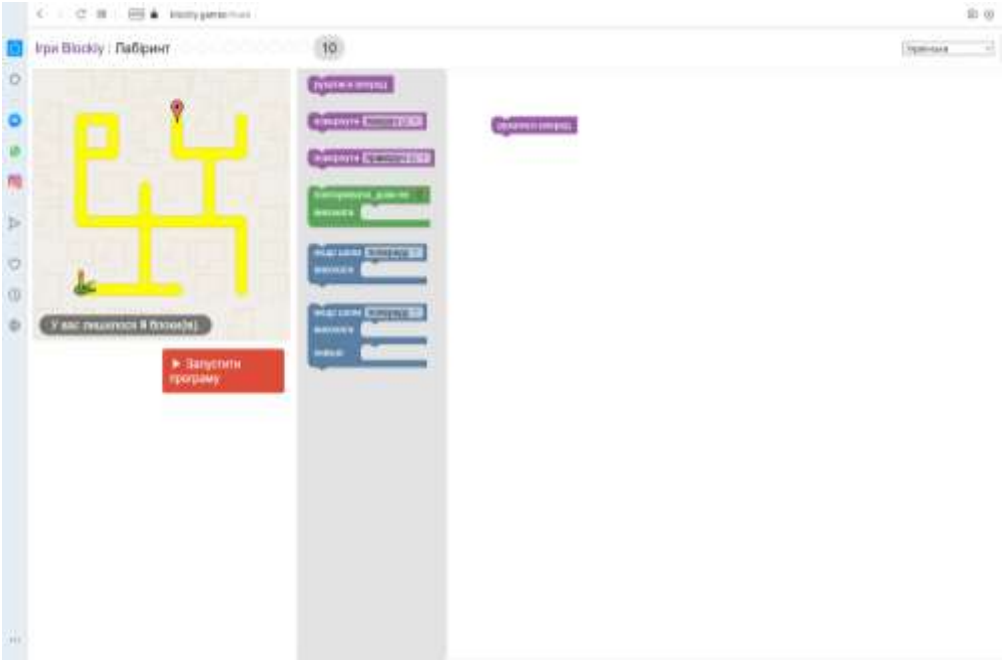
Зручним є те, що після виконання кожної програми можна переглянути код на JavaScript або Python.



Мал.1.4. Код програми

Дана спільнота проводить кожного року захід «Година коду», який має на меті об’єднати учнів усіх куточків світу та заохотити у вивченні програмування.

Не менш цікавим для молодших школярів є сервіс **Blockly**, основою якого є теж блочна форма складання команд. Зайшовши на сайт, можна побачити так звану стежку, де проходивши від однієї станції до іншої, можна отримати ґрунтовні початкові знання для майбутнього програмування. Тут теж відображаються готові коди JavaScript [21].



Мал.1.5.Онлайн-сервіс Blockly

Досить цікава програма для вивчення алгоритмізації і програмування є Tuxbot.

Мал.1.6. Інтерфейс програми TuxBot

Дана програма має англійський інтерфейс, тут є готові рівні. Цікавинкою є те, що пінгвін шукає рибку, яку хоче з'їсти. Щодо циклічних алгоритмів, то при певних налаштуваннях з'являється кнопка Repeat, яка дає можливість зекономити комірки для стрілок і одночасно повторити дію після неї [20].

Мал.1.7. Один із рівнів програми TuxBot

Цікавим для учнів також є створення власних рівнів, які потім вона може зберегти та експортувати у програму.

Отже, навчальних середовищ для програмування молодших школярів є безліч, головне учителеві вміло обрати ту чи іншу програму, розробити під неї методичні матеріали та правильно організувати роботу по опануванні. А це важливо, адже початкова ланка є базою для вивчення в майбутньому справжніх мов програмування.

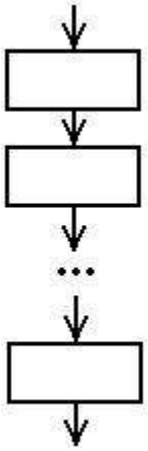
РОЗДІЛ 2. Методичні основи вивчення циклічних алгоритмів та побудови скриптів у початковій школі.

2.1.Методика ознайомлення з циклічним алгоритмом, методами його реалізації.

2.1.1. Методика побудови циклу з передумовою

Повернемося до видів алгоритмів, а саме до циклічного алгоритму або алгоритму з повторенням.

Раніше ми працювали з програмами, у яких команди виконувались послідовно.

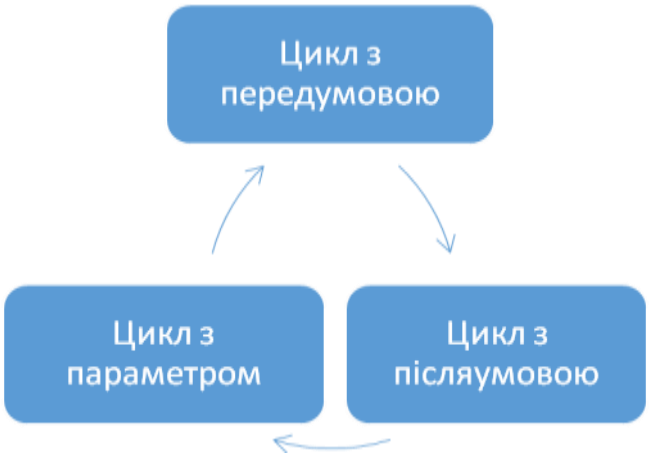


Мал.2.1. Схема лінійного алгоритму

Це так звані ***лінійні програми***. Проте досить часто певні команди слід повторювати багато разів. Зрозуміло, що було б дуже незручно писати одні й ті ж фрагменти програми кілька разів. Саме для цього використовують цикли.

Цикл - це фрагмент програми, що може виконуватись деяку кількість разів.

Існує три типи циклів або алгоритмів з повторенням:

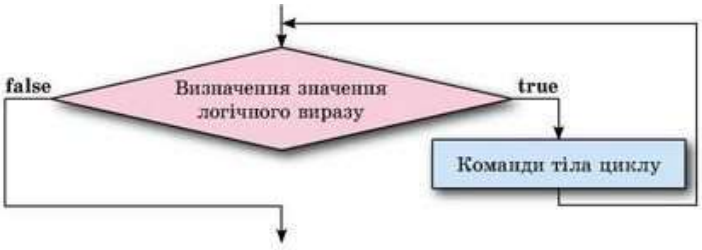


Смарт-схема 2.1. Види циклів

Цикл з передумовою як фрагмент алгоритму починається з команди перевірки умови й результатом виконання цієї команди може бути або істина (Так, true), або хиба (Ні, false). І залежно від результату виконання цієї команди виконуватимуться команди тіла циклу або команда алгоритму, наступна за циклом.

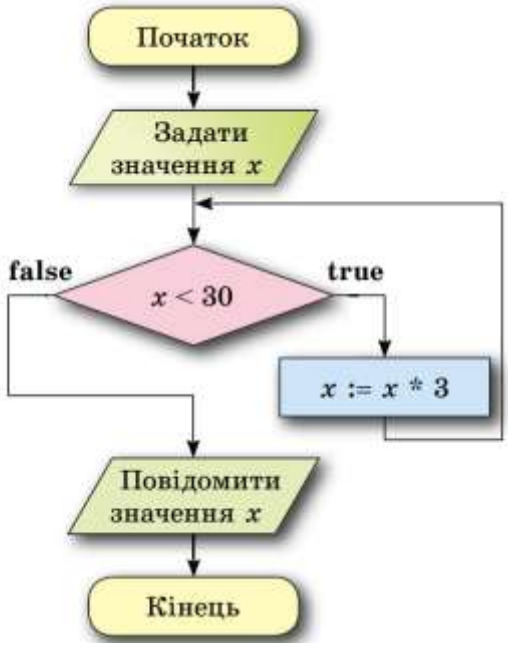
У загальному випадку у цій команді визначається значення певного логічного виразу, яке може бути або true, або false .

Загальний вигляд блок-схеми циклу з передумовою такий (Мал.2.1):



Мал.2.1. Цикл з передумовою

Для кращого розуміння розглянемо задачу. Задається число користувачу. Якщо воно менше 30, то множиться на три. Дія відбувається доти, доки число не стане більше за 30. На малюнку 2.2. показано блок-схему до задачі.



Мал.2.2. Блок-схема до задачі

Цикл з передумовою використовується у багатьох випадках, тому розуміння скласти блок-схему потрібно дітям ще з молодших класів.

Розглянемо ще таку задачу.

Є діжка, відро і колодязь з водою. Використовуючи відро, потрібно наповнити діжку водою.

Оскільки в цій задачі невідомо, чи є вода в діжці, чи діжка порожня, ні ємність діжки, ні ємність відра, то визначити, скільки разів потрібно виконати команди тіла циклу, не можливо.

Розглянемо виконавця з такою системою команд:

1.Наповнити відро водою.

2.Вилити воду з відра в діжку.

3.Перевірити умову «Діжка не повна?».

Алгоритм розв'язування цієї задачі для розглянутого виконавця у словесній формі виглядатиме так:

1.Перевірити умову «Діжка не повна?»

2.Якщо істина, то виконати команду 3, інакше (якщо хиба) виконати команду 6.

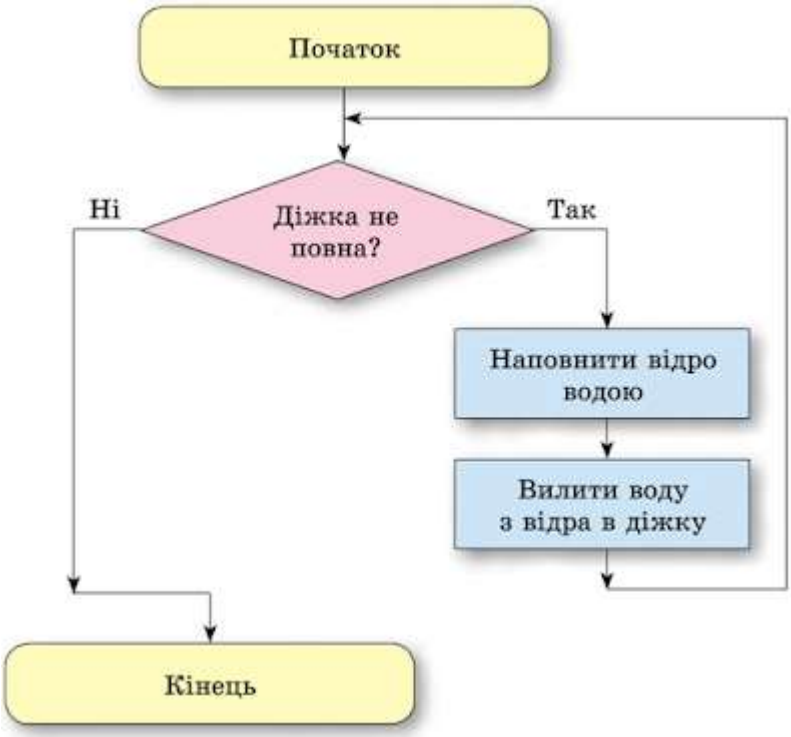
3.Наповнити відро водою.

4.Вилити воду з відра в діжку.

5.Виконати команду 1.

6.Закінчити виконання алгоритму.

Блок-схема цього алгоритму



Мал.2.3 Блок-схема до задачі

2.1.2. Методика побудови циклу з післяумовою

Якщо значення обчислюються доти, поки вони не задовольняють деякої умови, то потрібен **цикл із післяумовою**.

Мал.2.4. Блок-схема з післяумовою

Для прикладу розглянемо алгоритм прибирання своїх речей з парти після уроків.

Розглянемо блок-схему

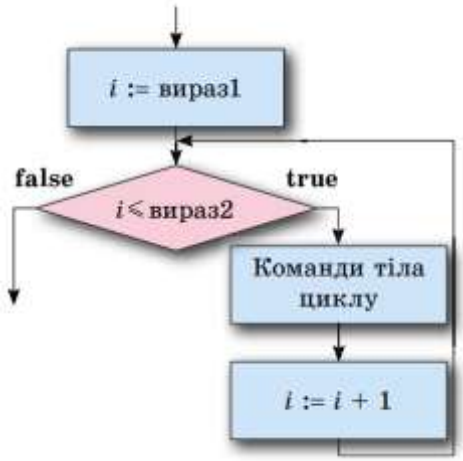
Мал.2.5. Блок-схема з післяумовою

Якщо на парті є хоча б один предмет, то треба взяти предмет з парти і покласти його в портфель. Після цього повернутися до команди перевірки умови. Знову перевірити, чи лишився на парті хоча б один предмет, і, якщо лишився, повторити дії з предметом.

Коли жодного предмета на парті не лишиться, цикл закінчиться.

2.1.3. Методика побудови циклу з переліком.

У алгоритмах з повторенням є команда циклу з лічильником. Її доцільно використовувати в тих випадках, коли кількість повторень команд тіла циклу відома ще до початку виконання команди.



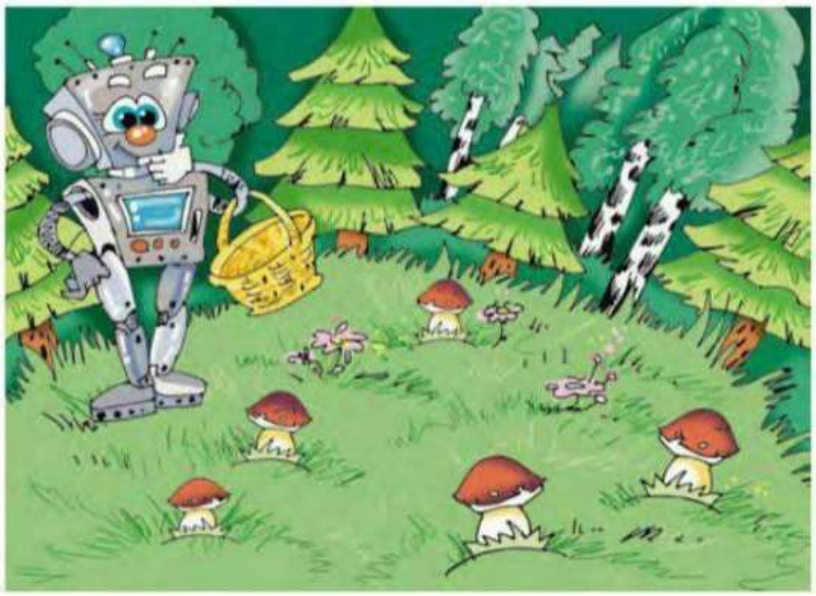
Мал.2.6. Блок-схема циклу з лічильником

Для прикладу можна розглянути просту життєву задачу набирання води у діжку.

Мал.2.7. Блок-схема реалізації циклу з лічильником

Також, типовою буде така задача.

Склади інструкцію роботу-лісовику, щоб він міг зібрати всі гриби на галявині.



Мал.2.8. Ілюстрація до задачі

Мал.2.9. Блок-схема до задачі

2.2. Реалізація алгоритмів з повторенням у середовищі Скретч.

Scratch - середовище програмування, що з'явилася відносно недавно, дає можливість учням молодшого та середнього шкільного віку створювати ігри, фільми, анімовані історії та багато іншого. Програма Scratch в об'єктно-орієнтованому середовищі «збирається» з різнокольорових блоків команд так само, як збираються з різнокольорових цеглинок в конструкторах Лего різні об'єкти. Є можливість внесення змін в програму навіть тоді, коли вона запущена, що дозволяє експериментувати з новими ідеями по ходу виконання завдання. В результаті виконання простих команд створюється складна модель, в якій взаємодіють безліч об'єктів, наділених різними властивостями. Після того як проект створений в Scratch, є можливість його розмістити на сайті.

Одним із принципових переваг даного середовища є те, що вона є вільно поширюваним програмним продуктом, таким чином, будь-який навчальний заклад може завантажити програму з інтернету і приступити до безпосереднього вивчення і роботі в новому середовищі програмування.

Уже в початковій школі діти легко можуть освоїти такі поняття як «паралельність» і «синхронізація». При цьому важливим є не «знання» термінології, але розуміння взаємної зв'язку виконуються потоків.

Scratch бере все найкраще від обчислювальної техніки і дизайну інтерфейсів для того, щоб зробити процес програмування більш привабливим і доступним для дітей, підлітків і тих, хто хоче навчитися програмуванню. Основні особливості Scratch:

• *Блочне програмування*. Для створення програм в Scratch, ви просто поєднуєте графічні блоки разом в стеках. Блоки зроблені так, щоб їх можна було зібрати тільки в синтаксично вірних конструкціях, що виключає помилки. Різні типи даних мають різні форми, підкреслюючи несумісність. Ви можете зробити зміни в стеках, навіть коли програма запущена, що дозволяє більше експериментувати з новими ідеями знову і знову.

• *Маніпуляції даними*. З Scratch ви можете створити програми, які керують і змішують графіку, анімацію, музику та звуки. Scratch розширює можливості управління візуальними даними, які популярні в сьогоднішній культурі.

• *Спільна робота і обмін*. Сайт проекту Scratch пропонує натхнення і аудиторію: ви можете подивитися проекти інших людей, використовувати і змінити їх картинки і скрипти, і додати ваш власний проект. Найбільше досягнення - це загальне середовище і культура, створена навколо Scratch.

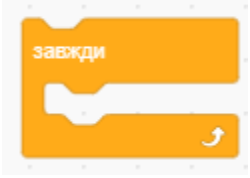
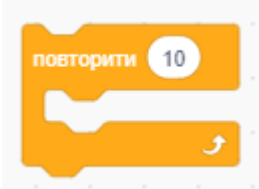
Scratch пропонує низьку підлогу (легко почати), висока стеля (можливість створювати складні проекти) і широкі стіни (підтримка великого різноманіття проектів). У роботі зі Scratch приділяється особлива увага простоті, іноді навіть на шкоду функціональності, для кращого розуміння.

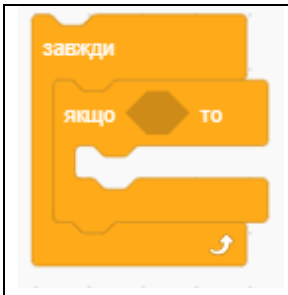
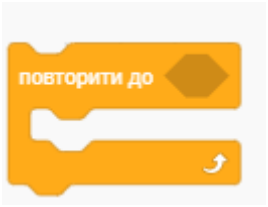
Коли учні працюють над проектом в Scratch, вони мають шанс вивчити важливі обчислювальні концепції, такі як повторення, умови, змінні, типи даних, події і процеси.

У даній роботі буде описано безпосередньо реалізація алгоритмів з повторенням.

Для початку визначимо основні блоки, для цього будемо аналізувати найновішу версію програми Scratch 3.6.

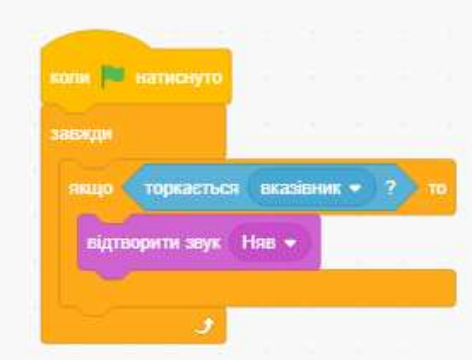
Таблиця 2.1

Блок у Scratch	Назва	Пояснення
	Безумовний цикл	Виконання дії постійно, поки не буде натиснута кнопка зупинити.
	Цикл з лічильником	Повторення команди або групи команд певну кількість разів

	З передумовою	Виконання команди з перевіркою умови
	З післяумовою	Йде виконання команд, доки не виконається умова

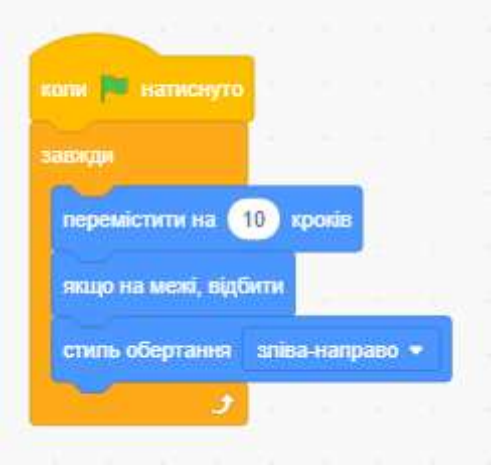
Розглянемо прості приклади для учнів 4 класу.

Приклад 1. Якщо показником миші торкнутися до котика, він нявкає.



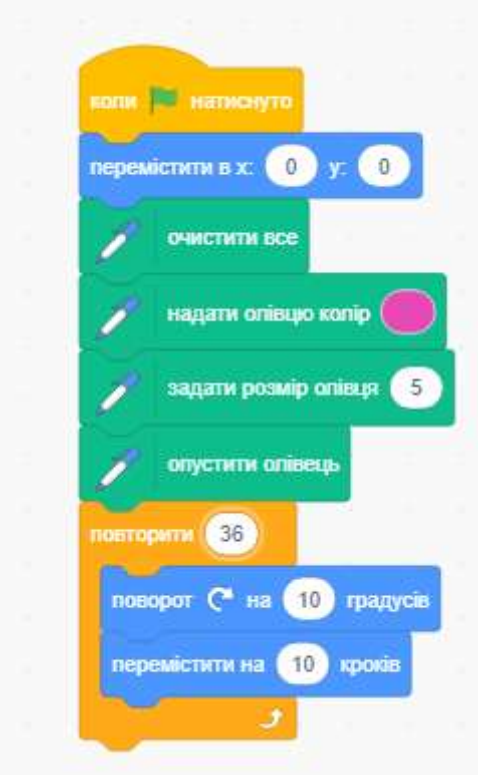
Лістинг 1.1. Реалізація циклу з передумовою

Приклад 2. Відповідний спрайт рухатиметься вправо до межі і повертатиметься вліво, продовжуючи рух. Це відбуватиметься завжди.



Лістинг 1.2. Реалізація безумовного циклу

Приклад 3. Зобразити невеличке коло



Лістинг 1.3. Реалізація циклу з лічильником

Приклад 4. Намалювати лінію до межі.

Лістинг 1.4. Реалізація циклу з післяумовою

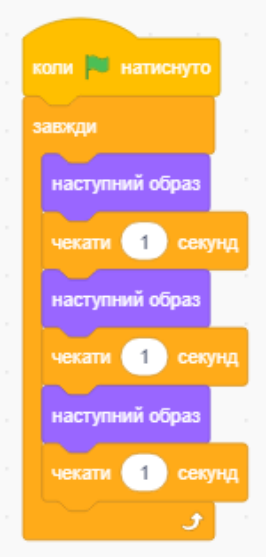
Відзначимо одну особливість роботи в середовищі Скретч, характерну для сучасних програмних середовищ. Так само, як в MS Word ми працюємо з документом, а в MS Excel ─ з книгою, так в Scratch ми завжди працюємо над проектом. Це пов'язано з тим, що створення будь-якого, навіть самого простого продукту в Scratch ─ анімації, мелодії, презентації і т. п., завжди вимагає наявності цілком визначеної мети діяльності, постійної звірки

отриманого результату з вихідним задумом і виправлення помилок. При роботі над складним проектом, що складається з великого кількості об'єктів, які містять складний програмний код, виникає необхідність розбиття вихідного проекту на підзавдання. При цьому вирішувати кожну таку підзадачу в принципі можуть різні учасники єдиного проекту.

Отже, як бачимо дуже легко використовувати цикли у середовищі Scratch. Дитина легко сприймає та засвоює основні види алгоритмів з повторенням через блоки.

2.3. Методика побудови скриптів з анімацією за допомогою циклу.

Для реалізації анімації об’єкта використовується цикл «завжди», в тілі якого відбувається послідовна зміна образів з затримкою в 1 секунду. Лістинг 1.5. представляє скрипт, що відображає послідовність команд, зібраних із певних блоків (категорії «Вигляд» і «Керувати»).

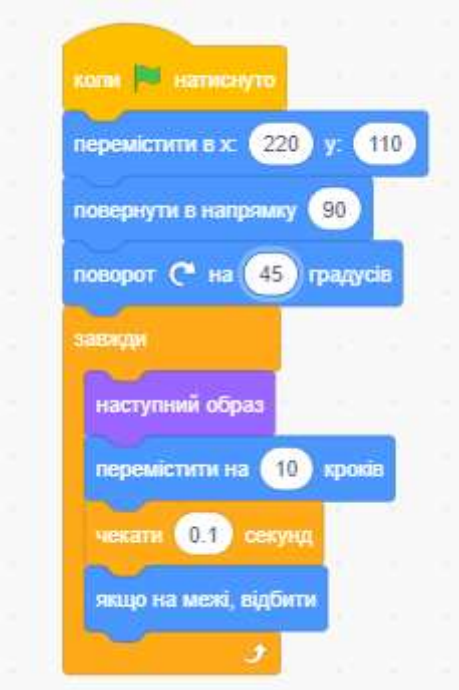


Лістинг 1.5. Вигляд скрипта, що демонструє циклічність зміни образів спрайта

Звичайно, ми можемо створювати анімації й для інших наших героїв, наприклад, метелика, якого ми додали на сцену. Спочатку потрібно імпортувати ще один костюм для метелика із бібліотеки.

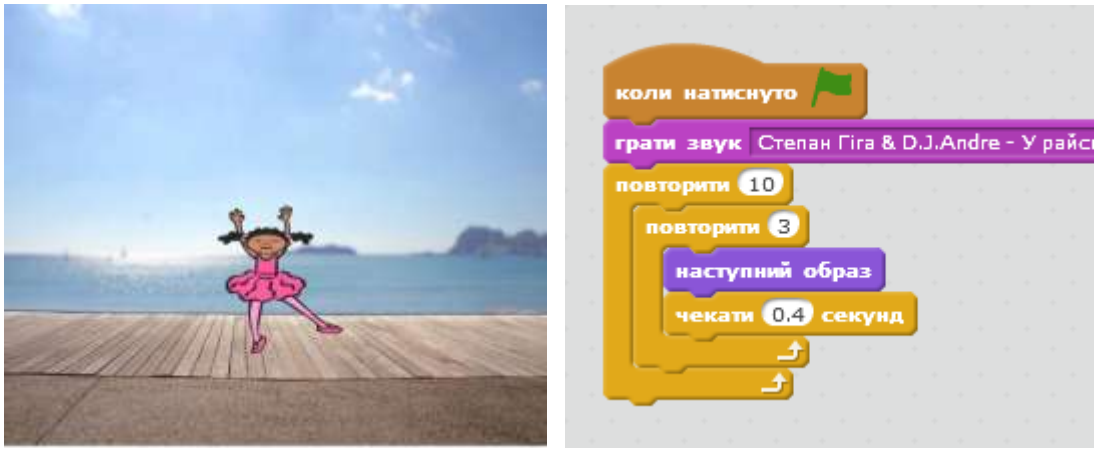
Мал.2.8. Метелик з бібліотеки спрайтів

А далі створити наступну програму:



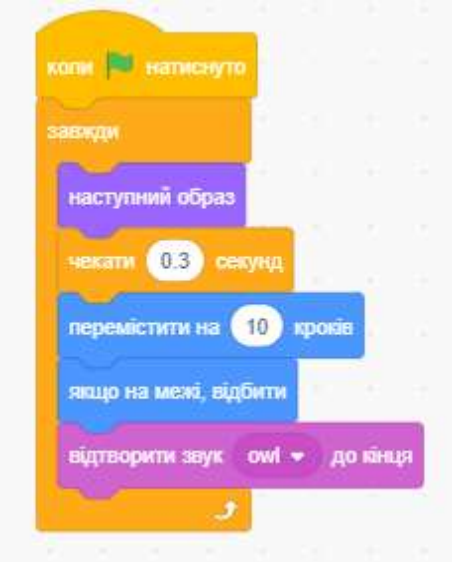
Лістинг 1.6. Програма для метелика

Можна створити анімацію для будь якого героя. Наприклад, герой балерина.



Лістинг 1.7. Анімація для балерини

Або для героя Кажан. Добре брати героїв, які мають безліч костюмів.



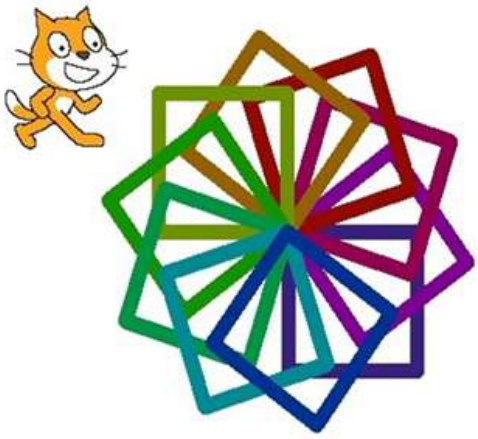
Лістинг 1.8. Анімація для спрайта Кажан

Дітям дуже подобається створювати анімації. Саме на зміні образів побудовані цикли. І дитина уже у 4 класі може легко створювати анімовані сюжети, ігри та мультфільми.

2.4. Методика побудови графічних композицій.

Учні також можуть створювати скрипти для анімації послідовно змінюючи образи об’єктів з певною затримкою в часі. Для цього використовують вбудований графічний редактор програми і можуть створити різні образи.

Наприклад, створюємо зірку.



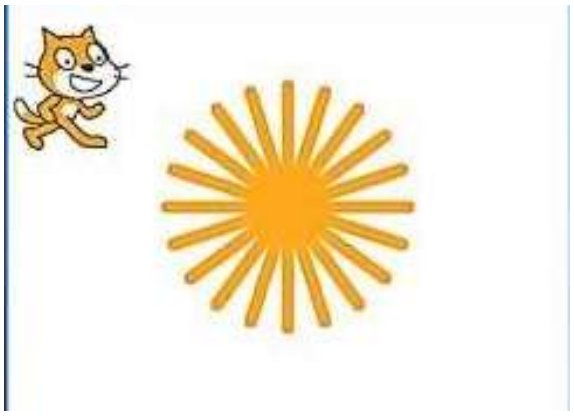
Мал.3.1.Графічна композиція

Реалізувати таку композицію можна за допомогою команди «Повторити». Для цього необхідно ввести такий код.

Мал.3.2. Код спрайту

Прикладів таких композицій може бути багато, варто лише залучати до цього фантазію.

Наприклад, потрібно створити сонце.

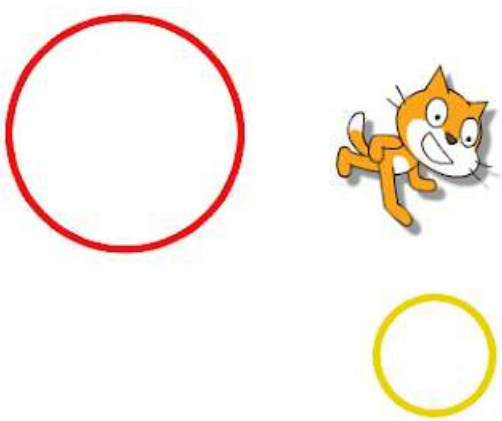


Мал.3.3. Графічна композиція

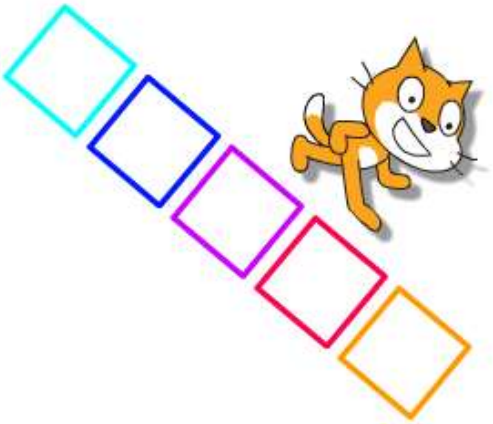
Щоб створити таку композицію необхідно ввести такий код

Мал.3.4.Код спрайту

Також до даної теми є доцільним таке завдання. Скласти та виконати алгоритм, щоб отримати такі зображення. Малювання квадрату та кола. (Мал.3.5.,3.6.)



Мал.3.5.



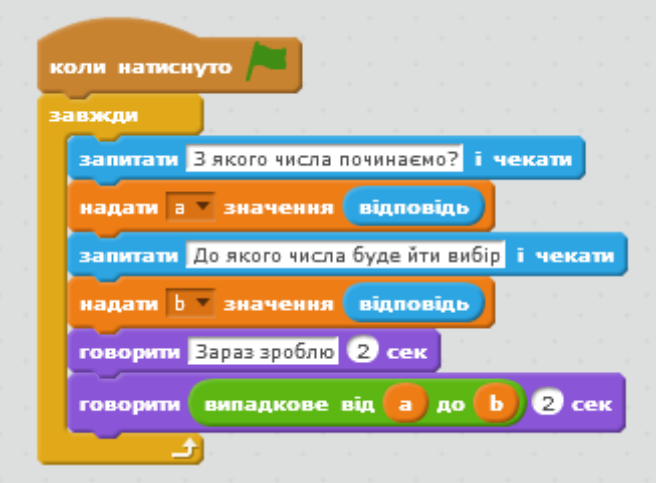
Мал.3.6.


Наприклад, щоб у проекті Вітраж, реалізованому в середовищі Скретч, отримати зображення з 15 різнокольорових квадратів, використовують два цикли.
15 разів забезпечує побудову об'єкта різними кольорами в деякому місці
повторити 15
перемістити в x: випадкове від -100 до 100 y: випадкове від -100 до 100
задати колір олівця випадкове від 1 до 100
опустити олівець
повторити 4
перемістити на 100 кроків
поворот 90 градусів
Забезпечує побудову сторін квадрата
підняти олівець
чекати 1 секунд
У програмі, що містить вкладені цикли, обов'язково спочатку виконується «внутрішній» цикл, а потім — «зовнішній».
© Вивчаємо інформатику teach-inf.at.ua

2.5. Побудова обчислювальних скриптів з циклом.

Для створення різноманітних проектів математичного спрямування, використовують змінні, випадкові числа. Але ця тема вивчається у середніх класах, коли в учнів є сформовані математичні здібності, тобто вони розуміють, що створюють і що буде результатом.

Наприклад, програма для вибору випадкового числа.



Лістинг 1.12. Програма для визначення випадкового числа

Дану програму можна використовувати при різноманітних лотереях. Коли треба вибрати випадкову людину. Наприклад, пронумерувати учнів класу, тоді ввести діапазон чисел і програма дасть випадкове число, під яким знаходиться відповідний учень, ну і він повинен для прикладу щось виконати.

Щодо циклу, то тут йде використання циклу ЗАВЖДИ.

Інший приклад, калькулятор для перевірки таблички множення. Дуже доречний проект на уроці математики у 4 класі. Тобто учні старших класів можуть виконати цей проект і надати вчителеві початківцю, який здійснить перевірку. Для кожної дитини вчитель може сам вибрати кількість прикладів.

Лістинг 1.13. Програма для перевірки таблиці множення

У програмі використано цикл з лічильником.

Обчислювальні цикли використовуються теж у середній школі, коли йде вивчення змінних. Тоді учні зможуть з легкістю створювати різноманітні програми з математики і не тільки.

2.6. Створення дидактичних матеріалів для організації практичних робіт з циклічними алгоритмами.

У середовищі Scratch вдало поєднуються ігровий інтерфейс та елементі програмування. Діти завжди охоче сприймають навчання, якщо воно містить елементи гри. Тому приклади, які містять створення малюнків цікаві та зрозумілі, вважаю найбільш доречними при вивченні теми.

Кожен може легко почати програмувати та створювати власні ігрові проекти.

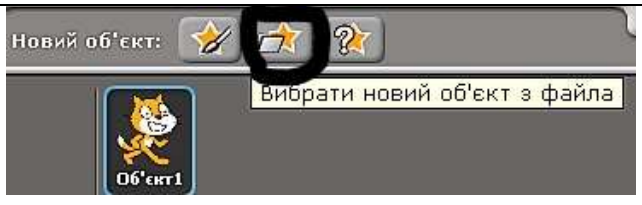
Для ефективного проведення уроку суттєвим є використання інструкційно-технологічних карток, тому що учні самостійно без додаткової допомоги вчителя виконують завдання, що сприяє:

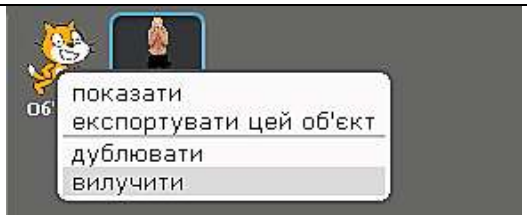
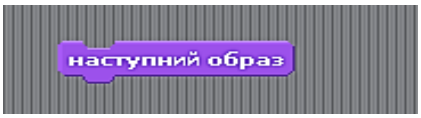
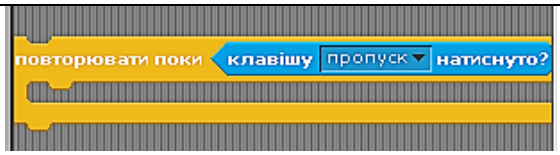
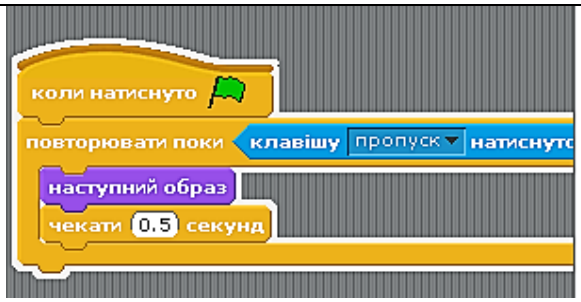
- ✓ закріпленню вміння та навичок учнів щодо певної теми, встановлення використання відповідних елементів, блоків, групи скриптів;
- ✓ розвитку логічного мислення та естетичного смаку при створенні та того чи іншого проекту;
- ✓ вихованню сучасної людини, що володіє інформаційною культурою.

За допомогою інструкційних карток кожен може виконати завдання, залежно від рівня, та визначити для себе оцінку за кожний виконаний етап роботи.

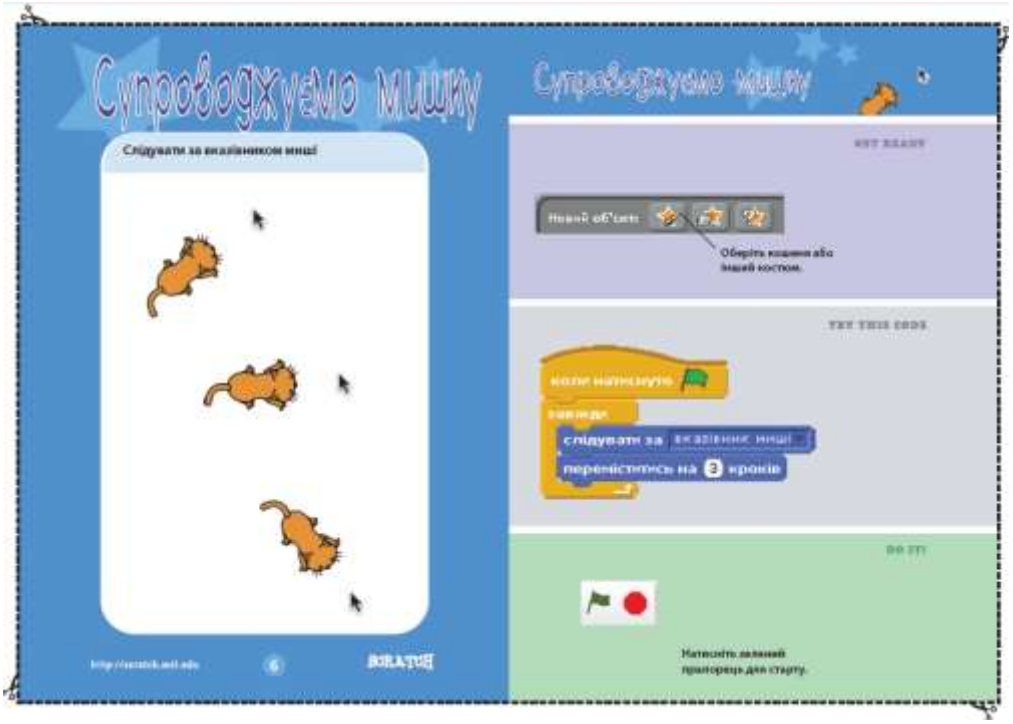
Наведемо приклади різного виду інструкційних карток.

Інструкційна картка №1

Порядок виконання	Вказівки по виконанню	Бали
1. Завантаж середовище Скретч. 2. Зміни виконавця Рудого кота на Танцюриста. Для цього у списку інструменті **Новий об'єкт** обери Вибрати новий об'єкт із файла.		

3. У вікні Новий об’єкт обери папку Люди(people), натисни кнопку Гаразд. У списку об’єктів обери перший вигляд виконавця(anjuli-1).		
4. У списку об’єктів вилучи Рудого кота (Об’єкт 1). Скористайся для цього відповідною вказівкою контекстного меню		
5. Перейди на вкладку **Образи** та додай ще три інші образи виконавця. Використай кнопку *імпортувати*. Клацни на першому образі – з нього почнеться танець.		
6. Утворюємо скрипт. 7. Із групи **Керувати** перетягуємо команду. 8. З групи **Вигляд** наступна команда. 9. З групи **Керувати** перетягуємо.	  	
10.З групи **Керувати** перетягуємо і об’єднуємо командою. 11. З групи **Датчики** вставляємо в цю команду таку умову		
12.Перевір виконання програми.		

Інструкційна картка №2

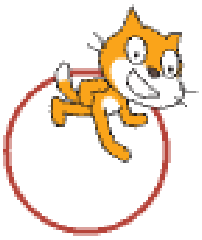


Інструкційна картка №3

Інструкційна картка №4

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм. (*Інструктаж із правил техніки безпеки.)*

Завдання. Створить програму малювання різнокольорових кілець. Перевірте роботу програми, запустивши скрипта на виконання.

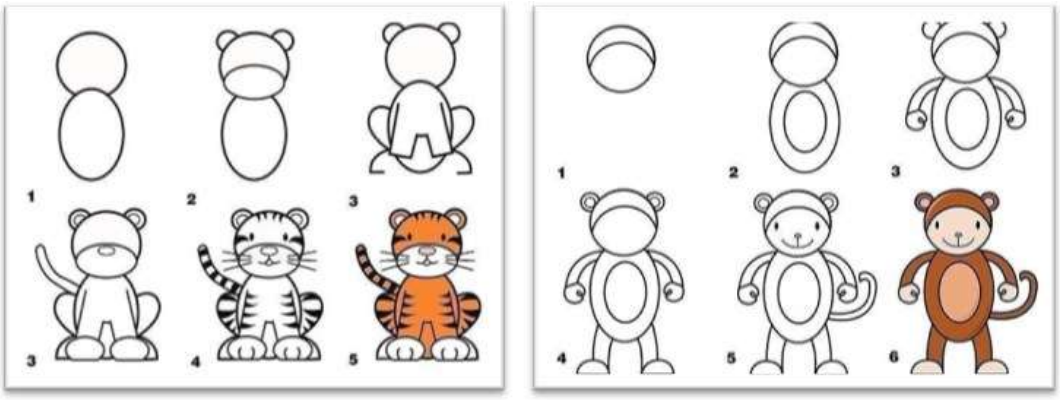
	

Можна також створювати карти із завданнями. Для прикладу:

Картка завдань

1. Створити в середовищі Скретч проект, у якому на сцені танцює виконавець, виконуючи різні рухи з деяким інтервалом.
2. Створити в середовищі Скретч проект, у якому рух м'яча припинявся при наведені вказівника миші на нього.
3. Створити в середовищі Скретч проект, орнамент з намальованих квітів.
4. Створи в середовищі Скретч проект, щоб краб змінював колір, коли натиснуто клавішу пропуск.

Можна також використати графічні інструкції, де дитина зможе намалювати відповідні образи.



Мал.2.10. Інструкція малювання образів

2.7. Досвід проведення занять інформатики з розділу «Алгоритми та виконавці»

Педагогічну практику я проходила у Яринівській філії в період з 05 лютого по 27 березня 2020 року.

Я проходила практику у 2 класі. Класний керівник цього класу – Трофимчук Ольга Валеріївна. У 2 класі 21 школяр. У перший день я познайомилася з усіма здобувачами освіти 2 класу під час проведення ранкової зустрічі.

Здобувачі освіти 2 класу розподілені на 2 підгрупи. За розкладом урок з інформатики у 2 класі був у середу та четвер п'ятим уроком. Учні не мали підручників з інформатики, проте були зошити з друкованою основою та роздатковий матеріал, підготовлений мною або ж вчителем.

Вчителем інформатики є Слудчик Василь Степанович.

Під час практики я проводила всі уроки з інформатики в 2 класі та ще в 3 і 4 класах.

У школі було створено всі умови для проходження практики. Кабінет інформатики достатньо обладнаний технічними засобами навчання, забезпечений наочними посібниками, науковою, навчальною та методичною літературою. Протягом усього перебування в школі я мала змогу отримати консультації від вчителя інформатики.

Під час практики провела 18 уроків з інформатики, з них 4 уроки, які стосувалися теми мого курсового дослідження. Усі уроки були обговорені та затверджені вчителем.

Коли я проводила уроки я використовувала наочність для успішного їх проведення :

- картки із завданнями та інструкційні картки;
- ребуси;
- кросворди.

Які ж труднощі були під час практики?
-Важко встановлювати зв'язок уже вивченого учнями матеріалу з тим, що вивчається, а також показувати перспективу практичного його використання.
-Не завжди діти прагнули дізнаватися щось нове та удосконалювати свої знання, через це необхідно було знаходити окремий підхід до кожного учня, зацікавлюючи школярів.

Діти дуже любили працювати у середовищі Scratch. На перших уроках у 2 класі панувало неабияке захоплення такою діяльністю. Діти все швидше хотіли навчитися самостійно працювати в такому середовищі. Навіть були діти, яким легко давалося виконувати всі поставлені мною завдання та навіть додаткові. Здобувачі освіти 4 класу також захоплено виконували всі завдання, з нетерпінням чекали уроків інформатики.
Для мене були цікаві всі аспекти практики. Під час спостереження за вчителем і учнями відкривається завіса таємниці шкільного життя. Позиція студентки на час змінюється на позицію вчителя.

Висновки

В наші дні інформатика – це галузь знань, що швидко розвивається; постійно росте масштаб використання наукових і практичних розробок інформатики в усіх сферах людської діяльності. Концентруючи знання багатьох наукових областей, інформатика виробляє методологію рішення завдань з використанням можливостей техніки, яка стає сьогодні методологією сучасного пізнання. Проблема пошуку шляхів викладання програмування для молодших школярів, системного рівня діяльності учнів школи, визначення змісту навчання інформатиці учнів початкової школи, в умовах розвитку інформаційного освітнього простору актуальна і вимагає подальшого дослідження.

Я вважаю, що можна бути стовідсотково впевненим щодо доцільності введення програмування в середовищі Scratch в початковій школі, що відповідає цілям розвитку учня і його підготовки до подальшої діяльності в інформаційному суспільстві. Scratch розроблявся як нове навчальне середовище для навчання школярів програмуванню. В той же час в цьому середовищі учні повною мірою можуть розкрити свої творчі таланти, оскільки в Scratch можна легко створювати фільми, ігри, анімовані листівки і презентації; придумувати і реалізовувати різні об’єкти, визначати, як вони виглядають в різних умовах, переміщати по екрану, встановлювати способи взаємодії між ними.

Отже, мета, яка була поставлена на початку написання роботи була досягнена і усі поставлені завдання виконані.

Маю надію, що дана робота допоможе майбутнім педагогам у якісній методичній підготовці щодо пропедевтики такої важливої змістової лінії «Алгоритмізація та програмування», адже база – це початкова ланка і якщо вона буде на належному рівні сформована, це велика основа для майбутнього випускника.

Список використаної літератури

1. Malan, D., and Leitner, H., Scratch for Budding Computer Scientists, SIGCSE Procedings, 2007.

2. Shafer, R. Oh! The Beauty and Joy of Computing, Innovations Volume 3, Issue 10, UC Berkeley College of Engineering, December 2009. See also http://inst.eecs.berkeley.edu/~cs10.

3. Wolz, U., Maloney, J., and Pulimood, S.J., “Scratch” Your Way to Introductory CS, ACM SIGCSE Bulletin, Volume 40 , Issue 1, March 2008.

4. Властивості алгоритмів [Електронний ресурс] – Режим доступу до ресурсу: https://sites.google.com/site/algoriymsitecorp/Home/ponatta-algoritmu/vlastivosti-algoritmiv.

5. Інформатика: підруч. для 4 класу загальноосвіт. навч. закл./ М. М. Корнієнко, С. М. Крамаровська, І. Т. Зарецька. – Х.: Вид-во «Ранок», 2015. – 160 с. : іл.

6. Ломаковська Г. В. Сходинки до інформатики : підруч. для 2 кл. загальноосвіт. навч. закладів / Г. В. Ломаковська, Г. О. Проценко, Й. Я. Ривкінд, Ф. М. Рівкінд. – К. : Видавничий дім «Освіта», 2012. – 160 с. 8.

7. Ломаковська Г. В. Сходинки до інформатики : підруч. для 3 кл. загальноосвіт. навч. закладів / Г. В. Ломаковська, Г. О. Проценко, Й. Я. Ривкінд, Ф. М. Рівкінд. – К. : Видавничий дім «Освіта», 2013. – 160 с.

8. Офіційний сайт Scratch [Електронний ресурс] – Режим доступу до ресурсу: https://scratch.mit.edu.

9. Патаракин Е.Д. Учимся готовить в среде Скретч / Е.Д. Патарикин. – М. : 2007. – Режим доступу: http://umr.rcokoit.ru/dld/metodsupport/scratch1.pdf.

10. Патаракин Е.Д. Школа Scratch / Е.Д. Патарикин // Школьные технологии. 2010. - № 4. - С. 132 – 135. пр. Кам.-Поділ. нац. ун-ту. Серія

педагогічна. – Кам'янець-Подільський: Кам.-Поділ. природничих спеціальностей педагогічних університетів / О. І. Теплицький // Зб. наук.

11. Сорокина Т.Е. Визуальная среда Scratch как средство мотивации учащихся основной школы к изучению программирования / Т.Е. Сорокина // Информатика и образование. 2015. - №5 (264). - С. 30 – 34.

12. С. О. Шлянчак // Інформаційні технології в освіті. 2016. - № 3 (28). С. 84-93.– Режим доступу: http://ite.kspu.edu/webfm_send/901.

13. Сходинки до інформатики: підруч. для 3-го кл. загальноосвіт. Навч. Закл. / О. В. Коршунова. – К.: Генеза, 2014. – 176 с. : іл.

14. Типи алгоритмів [Електронний ресурс] – Режим доступу до ресурсу: https://sites.google.com/site/infosite44e/Home/vstup/pravila-zobrazenna-blok-shem

15. Типова освітня програма, розроблена під керівництвом Савченко О. Я.

16. Типова освітня програма, розроблена під керівництвом Савченко О. Я. 3- 4 клас.

17. Форми представлення алгоритмів [Електронний ресурс] – Режим доступу до ресурсу: https://dl.sumdu.edu.ua/textbooks/108989/459146/index.html.

18. Офіційний сайт ScratchJr [Електронний ресурс] – Режим доступу до ресурсу: https://www.scratchjr.org.

19. Офіційний сайт Code [Електронний ресурс] – Режим доступу до ресурсу: https://code.org.

20. Офіційний сайт TuxBot [Електронний ресурс] – Режим доступу до ресурсу: http://appli-etna.ac-nantes.fr:8080/ia53/tice/ressources/tuxbot/index.php?fbclid=IwAR3ZljOQPPIU6LQWtQJWbVXqFxxMCD-38L06U_fzZawV-_LasNl2ynaLNgg#download.

21. Офіційний сайт Blockly [Електронний ресурс] – Режим доступу до ресурсу: https://blockly.games.

ДОДАТКИ

Додаток 1

Конспект уроку

Тема: Цикли з лічильником

Мета:

навчальна: ознайомити з поняттям «цикл з лічильником», продовжувати навчати створювати, записувати, виконувати алгоритми з циклами та реалізовувати у середовищі Scratch;

розвивальна: розвивати уважність, пам’ять, мовлення, логічне мислення, навички самостійної роботи на комп’ютері;

виховна: виховувати самостійність та відповідальність, працелюбність, акуратність у роботі.

Обладнання та наочність: комп’ютери, підручник, робочий зошит з друкованою основою, інструктивна картка.

Програмне забезпечення: Scratch

Тип уроку: комбінований

Клас: 4

План уроку

I. Організаційний момент (1хв.)

II. Актуалізація опорних знань (4 хв.)

III. Мотивація навчальної діяльності. Повідомлення теми і мети уроку (3 хв.)

IV. Вивчення нового матеріалу (13 хв.)

V. Застосування набутих знань (10 хв.)

Фізкультхвилинка

VI. Формування вмінь та навичок (10 хв.)

VII. Підбиття підсумків (3 хв.)

Хід уроку

I. Організаційний момент

Підготовка учнів до уроку

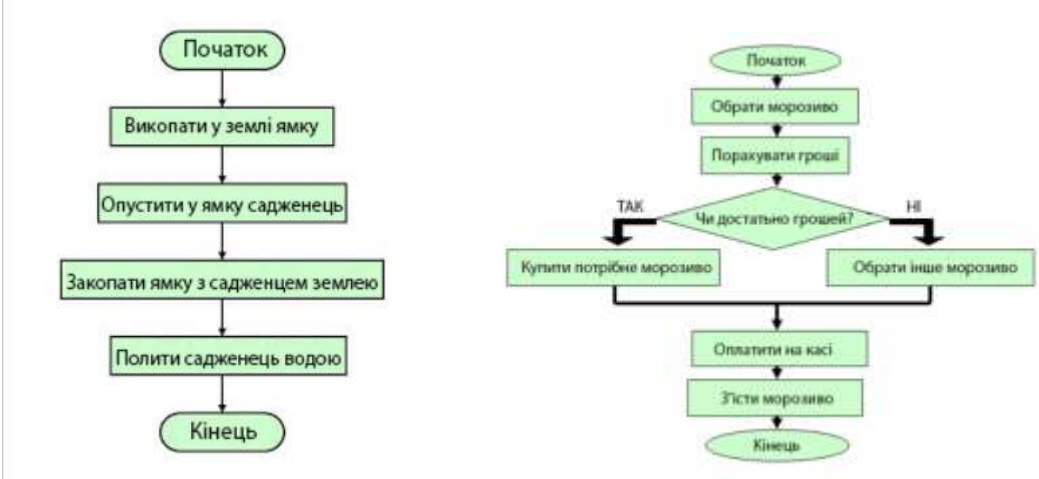
Привітання з класом

II. Актуалізація опорних знань.

- Давайте пригадаємо, що ви вивчали на минулому уроці?

- Які алгоритми можна назвати алгоритмами з розгалуженням?

- Який алгоритм є алгоритмом з розгалуженням: посадки дерева чи алгоритм купівлі морозива?



III. Мотивація навчальної діяльності. Повідомлення теми уроку.

Розгадайте ребус:

(Цикл)

- Отже сьогодні на уроці ми поговоримо про цикли, а тема нашого уроку – цикли з лічильником, тобто цикли які повторюються задану кількість разів.

IV. Вивчення нового матеріалу

- Як ви думаєте, що таке цикл? (багаторазове повторення певної послідовності дій)

- Цикл - це фрагмент алгоритму який може виконуватись більше одного разу.

- Ви вже знаєте, що таке цикл, а що ж таке циклічний процес?

Циклічними називаються процеси, в яких дії повторюються в одній і тій самій послідовності.

- Як ви думаєте, можемо ми спостерігати в природі циклічні процеси?

Планета Земля кожного року проходить один і той самий шлях навколо Сонця. Кожного року відбувається зміна пір року: зима, весна, літо, осінь, а потім знову зима. Кожну добу день змінюється ніччю, а ніч – днем, тобто виконується певний цикл.

- А чи можуть циклічні процеси відбуватися в нашому житті ви дізнаєтесь з підручника на с. 119

- Отож, які циклічні процеси відбуваються в нашому житті?

В алгоритмах розв’язування багатьох задач потрібно виконати одну або кілька команд більше ніж один раз. Для цього такі алгоритми мають містити параметри, які визначають кількість повторення певних команд.

Розглянемо таку задачу.

Задача. У дворі є порожні діжка і відро ємністю 50 л і 10 л відповідно та колодязь (мал. 3.4). Потрібно наповнити діжку водою

Мал. 3.4. Діжка, відро та колодязь

Очевидно, для розв’язування цієї задачі потрібно виконати такий алгоритм:

1. Узяти відро.

2. Повторити 5 разів

1. Підійти до колодязя.

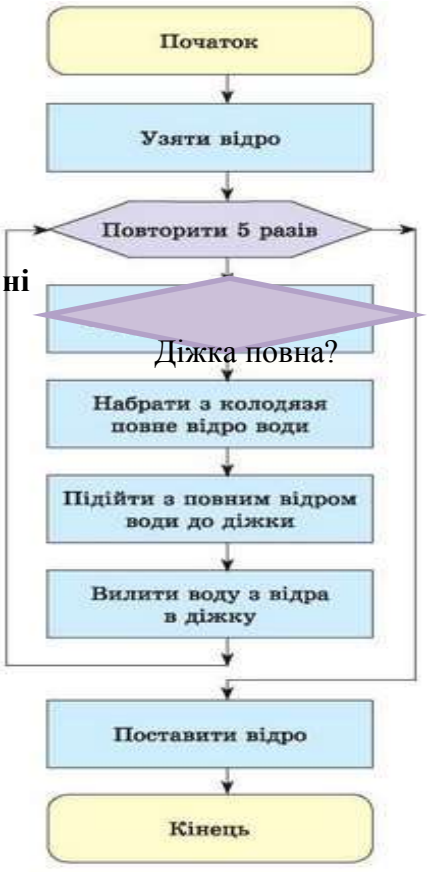
2. Набрати з колодязя повне відро води.

3. Підійти з повним відром води до діжки.

4. Вилити воду з відра в діжку.

3. Поставити відро.

Цей алгоритм містить команду 2:



Мал. 3.5. Блок-схема алгоритму наповнення діжки водою

Повторити 5 разів

1. Підійти до колодязя.

2. Набрати з колодязя повне відро води.

3. Підійти з повним відром води до діжки.

4. Вилити воду з відра в діжку.

Таку команду називають командою циклу з лічильником. Вона визначає, що під час виконання алгоритму команди:

1. Підійти до колодязя.

2. Набрати з колодязя повне відро води.

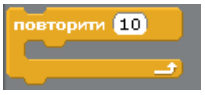
3. Підійти з повним відром води до діжки.

4. Вилити воду з відра в діжку - повинні виконатися 5 разів поспіль.

Вони утворюють тіло циклу.

Сама команда «Повторити 5 разів» задає кількість повторень тіла циклу. Її називають заголовком циклу.

У середовищі Scratch також можна складати алгоритми із циклами. Для цього в системі команд виконавців є спеціальні команди. Зокрема, для організації в алгоритмі циклу з лічильником можна використати команду «повторити», яка розміщена в групі Керувати.



Її вибір приводить до виконання вказану кількість разів команд, які містяться всередині цього блока. Зрозуміло, що кількість повторень команд тіла циклу

можна змінювати. Наприклад давайте разом виконаємо наведений алгоритм, де Рудий кіт намалює нам орнамент. Для цього:

V. Застосування набутих знань

Робота в зошиті з друкованою основою (1-4 завдання)

Завдання 1. Доповни алгоритм збирання солодощів для Елзіка.

Взяти цукерку.

Вправо □ клітинки.

Взяти цукерку.

Вправо □ клітинка.

Вгору □ клітинка.

Взяти цукерку.

Вліво □ клітинки.

Вгору □ клітинки.

Взяти цукерку.

Вправо □ клітинки.

Вгору □ клітинка.

Взяти цукерку.

Завдання 2. Запиши, скільки разів на тиждень Олесь виконує даний алгоритм. Чи є цей алгоритм циклом? Поясни свою відповідь.

Прокинутися о 7 год.

Зібратися до школи.

Піти в школу.

Навчатися.

Повернутися додому

Зробити уроки.

_____

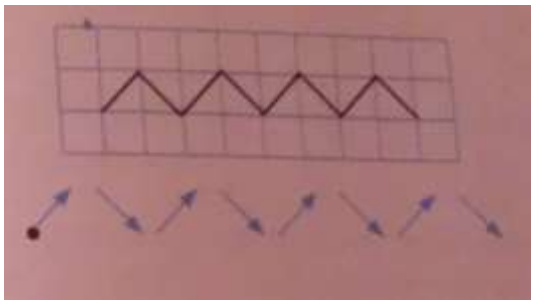
—

_____

—

Завдання 3. Запиши для Олівця алгоритм зображення малюнка. Виконай алгоритм двічі.

Завдання 4. В алгоритмі виділити групу команд, що повторюються. Полічи та запиши кількість повторень.



Фізкультхвилинка

Руки в сторони та вгору —

На носочки піднялись.

Підніми голівку вгору —

Й на долоньки подивись.

Присідати ми почнемо,

Добре ноги розімнемо.

Раз — присіли, руки прямо.

Встали — знову все так само.

Повертаємося вправно,

Все виконуємо гарно.

Вліво-вправо повернулись

І сусіду усміхнулись.

VI. Формування вмінь і навичок.

Робота за комп’ютером. Виконання практичного завдання. Повторення правил поведінки за комп’ютером

Тема: Цикл з лічильником.

Мета: навчити складати та реалізовувати вміння виконатувати алгоритми з циклами у середовищі Scratch.

Завдання: створити в середовищі Scratch програму, за якою виконавець малюватиме зірку.

Порядок виконання:

Завантажте середовище Scratch.

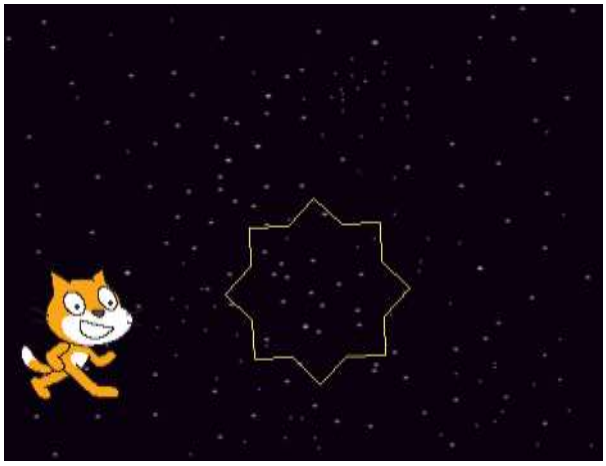
Змініть фон сцени.

Налаштуйте олівець для малювання

Складіть скрипт, використовуючи команду «повторити»

Перевірте роботу скрипта.

Завершіть роботу із середовищем Scratch.



VII. Підбиття підсумків уроку

Бесіда за питаннями

- Що таке цикл? (це фрагмент алгоритму який може виконуватись більше одного разу)

- Які процеси називаються циклічними? (в яких дії повторюються в одній і тій самій послідовності)

- Як називаються цикли які повторюються на задану кількість разів? (цикли з лічильником)

Додаток 2

Конспект уроку

Тема: Цикли з лічильником.

Мета:

- *навчальна*: сформувати свідомі знання з даної теми, закріпити правила поведінки в школі, вдосконалити навички створення проектів з використанням алгоритмів ;
- *розвивальна*: розвивати мислительні процеси дітей та моторику рук, сприяти всебічному розвитку;
- *виховна*: виховувати інтерес до вивчення інформатики.

Тип уроку: засвоєння нових знань, формування вмінь.

Обладнання та наочність: комп'ютери, підручники, презентація, проектор.

Програмне забезпечення: Scratch

Хід уроку

I. Організаційний етап

II. Актуалізація опорних знань

Повторення раніше вивченого матеріалу

1. Що таке цикл?
2. Наведіть приклад з реального життя на своєму прикладі.
3. Який блок відповідає за циклічність?

Повторення правил поведінки в комп'ютерному класі

III. Мотивація навчальної діяльності

Учитель. Сьогодні ми продовжимо працювати з циклами із лічильником

IV. Вивчення нового матеріалу

Розповідь учителя (*з демонструванням презентації на екрані*)

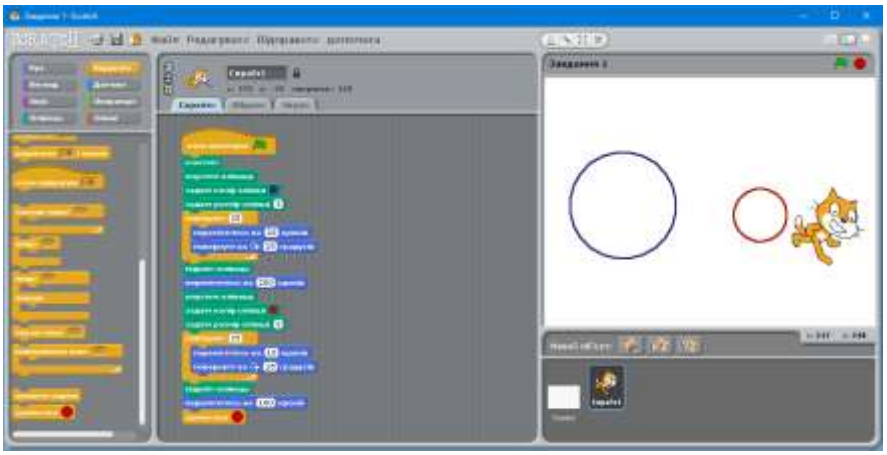
V. Фізкультхвилинка

VI. Усвідомлення набутих знань

VII. Формування вмінь та навичок

Практичне завдання

Завдання 1 - Створити опрацьований проект



Релаксація

Вправа для профілактики короткозорості та порушення зору

VIII. Підбиття підсумків уроку

1. Як в блоці циклічності збільшити кількість повторень.
2. При малюванні круга як збільшити його радіус?

Додаток 3. Завдання для практичної роботи

Завдання 1.“Додавання нового спрайту до проекту. Створення анімації для нього”

Мета: навчитися додавати до проекту новий спрайт і створювати для нього анімацію

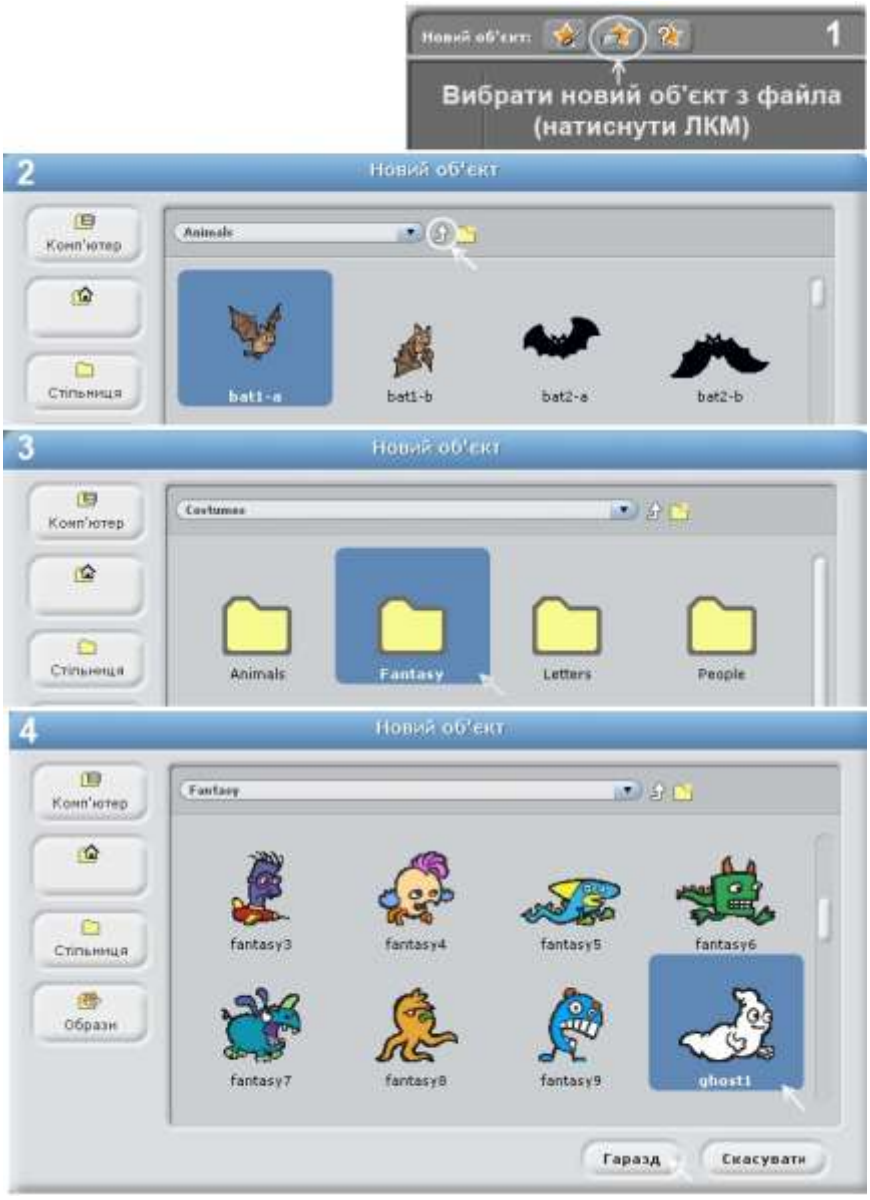
Завдання. Необхідно створити анімацію літаючого привида (якщо натиснута кнопка “стрілка вправо” – привид летить право, якщо натиснута кнопка “стрілка вліво” – привид летить вліво).

Технологія виконання завдання.

1. Запустіть програму Scratch.

2. Видаліть з проекту Спрайт 1, він нам не знадобиться для роботи.

3. Додайте до проекту новий спрайт, натиснувши кнопку вибрати новий об’єкт з файлу. Спрайт вибрати з папки Fantasy, ім’я спрайту Ghost1.



4. Для обраного спрайту складіть наступний скрипт.

5. Переведіть проект в режим демонстрації.

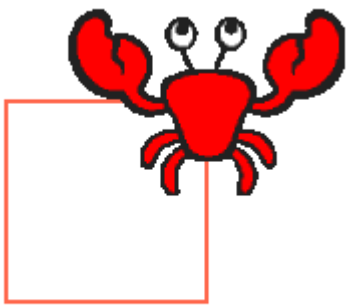
6. Для запуску проекту натискайте або на “стрілку вліво” з клавіатури, або на “стрілку вправо”. Переконайтеся, що проект працює правильно.

Додаткове завдання 1. Додайте можливість здійснювати рух привида при натисканні на кнопки “стрілка вгору” і “стрілка вниз”.

Додаткове завдання 2. Створіть анімацію крокуючої людини, з кнопками керування; стиль руху, фон і звук вибрати довільно.

Завдання 2.“Створення та виконання алгоритмів з повторенням”

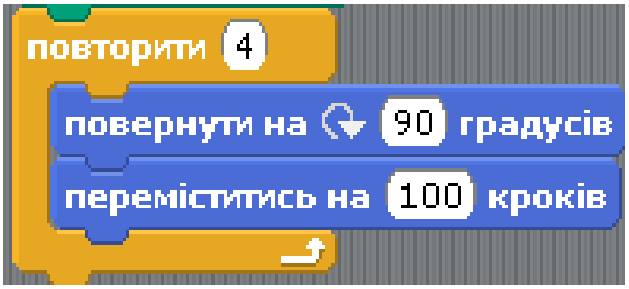
Вправа 1. Створення циклічного алгоритму у середовищі Scratch



1. Запусти програму Scratch.
2. Оберіть героя із папки Animals на свій розсуд.
3. Створи алгоритм побудови узору:

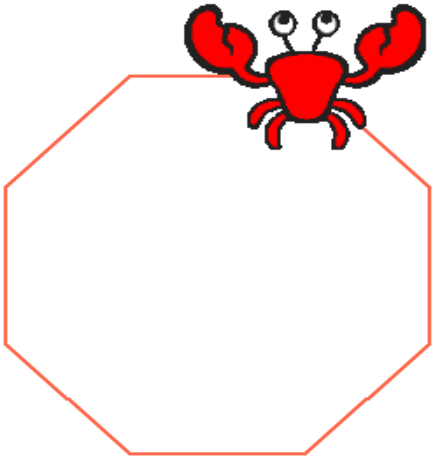
В блоці команд Керувати оберіть команду 

1. Для того, щоб олівець почав малювати потрібно задати його колір, розмір та опустити його, для цього:
   - перейдіть в блок Олівець та оберіть команди Задати колір олівця (колір обирається на свій розсуд);
   - оберіть команду Задати розмір олівця (наприклад, 2 розмір), - наступна команда Опустити олівець.
2. Для того, щоб герой малював фігуру, потрібно обрати кут на який герой буде повертатися. Якщо це квадрат, то потрібно 360 градусів поділити на кількість сторін. Ми отримаємо 90^0.
3. У блоці Рух обираємо команду Повернутися на ___ градусів (обираємо на 90^0).
4. У блоці Рух обираємо команду Переміститись на __ кроків (обираємо на 100 кроків).
5. Для того, щоб герой намалював чотири сторони, застосовуємо циклічний алгоритм, в блоці Керувати обираємо команду Повторити__ (повторити 4 рази, бо наш квадрат має 4 сторони). Та вставляємо в середину цієї команди блок команд Повернутися на 90^0 градусів та Переміститись на 100 кроків.

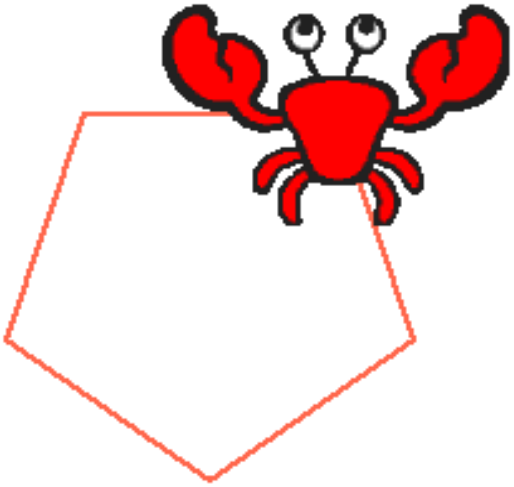


Вправа 2. Створення циклічного алгоритму у середовищі Scratch

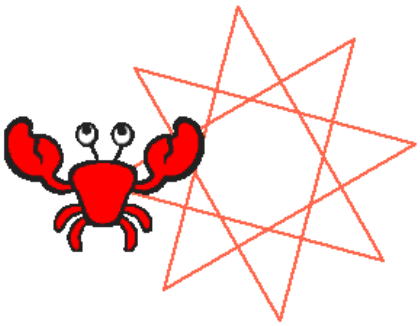
Завдання. За даним зразком створи узори, які показані на малюнку 1 та 2.



Малюнок 1.



Малюнок 2.



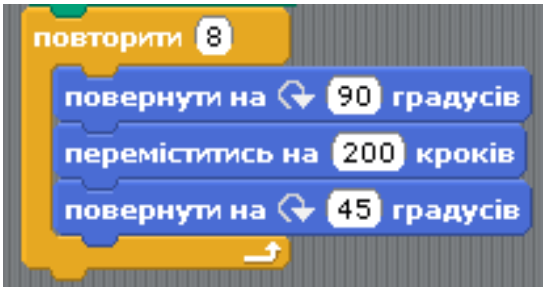
Вправа 3. Створення циклічного алгоритму у середовищі Scratch

Завдання. Створення узорів

1. Запустіть програму Scratch.
2. Оберіть героя із папки Animals на свій розсуд.
3. Створіть алгоритм побудови узору:

1. В блоці команд Керувати оберіть команду
2. Для того, щоб олівець почав малювати потрібно задати його колір, розмір та опустити його, для цього:
   - перейдіть в блок Олівець та оберіть команди Задати колір олівця (колір обирається на власний розсуд),
   - оберіть команду Задати розмір олівця (наприклад, 2 розмір), - далі команду Опустити олівець.
3. У блоці Рух обираємо команду Повернутися на ___ градусів (обираємо на 90^0).
4. У блоці Рух обираємо команду Переміститись на __ кроків (обираємо на 200 кроків).

5. У блоці Рух обираємо команду Повернутися на ___ градусів (обираємо на 45^0).
6. Застосовуємо циклічний алгоритм, в блоці Керування обираємо команду Повторити__ (повторити 8 раз). Та вставляємо в середину цієї команди, команди Повернутися на 90^0 градусів та Перемістити на 200 кроків та Повернутися на 45^0 градусів.



Додаток 4

Інформатичний диктант

Середовище Scratch

1.Послідовність команд або дій, що призводять до результату - це...

А) Висловлювання

Б)Скрипт

В)Алгоритм

Г)Програмування

2.Виконавець команд у середовищі Scratch - це...

А) Спрайт

Б) Персонаж

В) Скрипт

3.Місце, де відбуваються події в середовищі Scratch - це...

А) Фон

Б) Програма

В) Сцена

Г) Арена

4.Файл, створений у середовищі Scratch, називають...

А) Папкою

Б) Проектом

В) Рефератом

Г) Програмою

5.Для виконання команд на сцені потрібно вибрати кнопку...

А) Червоний кружечок

Б) Синій прапорець

В) Зелений кружечок

Г) Зелений прапорець

6.Команди, об’єднані в групи, в середовищі Scratch називають...

А) Контейнер

Б) Скрипт

В) Проект

Г) Програма