

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 розгалужених алгоритмів»**

Студента **IV** курсу **A-42** групи

Напрямок підготовки **0101**

Педагогічна освіта

спеціальність 013

Початкова освіта

Котика Василя Леонтійовича

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	3
РОЗДІЛ I. Теоретичні основи вивчення основ алгоритмізації в початкових класах.	
1.1 Основні поняття теорії алгоритмів	5
1.2 Властивості алгоритмів.....	6
1.3 Форми представлення алгоритмів.....	7
РОЗДІЛ II. Методичні основи вивчення розгалужених алгоритмів та побудови скриптів на їх основі.	
2.1 Методика ознайомлення з розгалуженими алгоритмами	10
2.2 Методика побудови алгоритмів за схемою логічного слідування	12
2.3 Реалізація розгалуження в середовищі Scratch.....	16
2.4 Методика створення дидактичних матеріалів для теми «Алгоритми з розгалуженням»	24
ВИСНОВКИ	29
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	30
ДОДАТКИ	31

Вступ

Актуальність теми. У сучасному житті розробка ефективних алгоритмів займає значну роль при вирішенні різного роду задач за допомогою електронних обчислювальних машин (ЕОМ). У відсотковому відношенні етап розробки алгоритмів займає приблизно 20% всього часу повного вирішення завдання. Цей етап є найбільш відповідальним, оскільки алгоритмічні помилки дуже часто важко виявити і виправити, в окремих випадках набагато простіше написати новий алгоритм і програму, ніж виправити існуючу. Зараз за допомогою алгоритмів вирішуються не тільки обчислювальні, а й багато прикладних завдань.

Алгоритм - це точний припис, що визначає обчислювальний процес, що веде від варійованих вихідних даних до шуканого результату. Саме слово «алгоритм» походить від латинської форми написання імені великого математика IX століття Аль Хорезмі (Мухаммеда ібн Муса аль Ногесмі), який сформулював правила виконання арифметичних дій. Спочатку під алгоритмом і розуміли тільки правила виконання чотирьох арифметичних дій над багатозначними числами, а в подальшому це поняття стало використовуватися для позначення послідовності дій, що призводять до вирішення поставленого завдання. Призначення алгоритмів полягає в поясненні послідовності ходу виконання кроків для вирішення конкретного завдання.

За типом розрізняють лінійні, розгалужені, циклічні й змішані алгоритми.

В даній курсовій роботі мова йтиметься про розгалужені алгоритми.

Алгоритм з розгалуженням – це алгоритм, який приписує виконання тих чи інших дій у залежності від результату перевірки умови [1].

Мета курсової роботи полягає в дослідженні методів створення алгоритмів з розгалуженням.

Для досягнення зазначеної мети необхідно вирішити наступні **завдання**:

1. Ознайомитися з основами теорії алгоритмів.
2. Ознайомитися з методикою та прикладами задач в середовищі Scratch.

3. Розглянути розв'язання типових задач на розгалуження в середовищі Скретч.

4. Ознайомитися з створенням ігрових задач в мережі Інтернет.

Об'єктом дослідження є алгоритми з розгалуженнями.

Предметом дослідження є сукупність теоретичних, методичних і практичних аспектів реалізації алгоритмів з розгалуженням.

Інформаційною базою роботи є отримані знання та навички протягом навчання та матеріали досліджень вітчизняних і зарубіжних вчених, опубліковані у періодичних виданнях та розміщені у мережі Інтернет.

Структура роботи. Основна частина даної курсової роботи складається з двох частин. В першій частині описується теоретичний матеріал, а в другій – методики побудови алгоритмів з розгалуженням. В курсовій роботі наведені висновки.

I. Теоретичні основи вивчення основ алгоритмізації в початкових класах.

1.1 Основні поняття теорії алгоритмів

Алгоритми - система правил, сформульована на зрозумілій виконавцеві мові, що визначає процес переходу від припустимих вихідних даних до деякого результату й має властивості масовості, кінцівки, визначеності, детермінованості.

Слово "алгоритм" походить від імені великого середньоазіатського вченого 8-9 ст. Аль-Хорезми (Хорезм - історична область на території сучасного Узбекистану). З математичних робіт Аль-Хорезми до нас дійшли тільки дві - алгебраїчна (від назви цієї книги народилося слово алгебра) і арифметична. Друга книга довгий час вважалася загубленою, але в 1857 у бібліотеці Кембриджського університету був знайдений її переклад на латинську мову. У ній описані чотири правила арифметичних дій, практично ті ж, що використовуються й зараз. Перші рядки цієї книги були переведені так: "Сказав Алгоритми. Віддамо належну хвалу Богу, нашому вождеві й захисникові". Так ім'я Аль-Хорезми перейшло в Алгоритми, звідки й з'явилося слово алгоритм. Термін алгоритм вживався для позначення чотирьох арифметичних операцій, саме в такому значенні він і ввійшов у деякі європейські мови. Наприклад, в авторитетному словнику англійської мови Webster's New World Dictionary, виданому в 1957, слово алгоритм позначене позначкою "застаріле" і пояснюється як виконання арифметичних дій за допомогою арабських цифр.

Слово "алгоритм" знову стало вживаним з появою електронних обчислювальних машин для позначення сукупності дій, що становлять деякий процес. Тут мається на увазі не тільки процес рішення деякого математичного завдання, але й багато інших послідовних правил, що не мають відносини до математики, - всі ці правила є алгоритмами [2].

Виконуючи те чи інше завдання за допомогою алгоритмів можна виділити властивості, які виконують будь-які алгоритми.

1.2 Властивості алгоритмів

До властивостей алгоритмів належать:

1. Зрозумілість. Щоб виконавець міг досягти поставленої перед ним мети, використовуючи даний алгоритм, він повинен уміти виконувати кожну його вказівку, тобто розуміти кожну з команд, що входять до алгоритму.

Наприклад: Дорослому сину-підлітку необхідно детальніше пояснити в яких магазинах можна придбати потрібний товар, що можна купити замість відсутнього товару і таке інше.

2. Визначеність (однозначність). Зрозумілий алгоритм все ж таки не повинен містити вказівки, зміст яких може сприйматися неоднозначно. Наприклад, вказівки "почистить картоплю", "посоли за смаком", "прибрати в квартирі" є неоднозначними, тому що в різних випадках можуть призвести до різних результатів.

Крім того, в алгоритмах неприпустимі такі ситуації, коли після виконання чергового розпорядження алгоритму виконавцю не зрозуміло, що потрібно робити на наступному кроці.

Отож, точність - це властивість алгоритму, що полягає в тім, що алгоритм повинен бути однозначно витлумачений і на кожному кроці виконавець повинен знати, що йому робити далі.

3. Дискретність. Як було згадано вище, алгоритм задає повну послідовність дій, які необхідно виконувати для розв'язання задачі. При цьому, для виконання цих дій їх розбивають у визначеній послідовності на прості кроки. Виконати дії наступного розпорядження можна лише виконавши дії попереднього. Ця розбивка алгоритму на окремі елементарні дії (команди), що легко виконуються даним виконавцем, і називається дискретністю.

4. Масовість. Дуже важливо, щоб складений алгоритм забезпечував розв'язання не однієї окремої задачі, а міг виконувати розв'язання широкого класу задач даного типу. Наприклад, алгоритм покупки якого-небудь товару буде завжди однаковий, незалежно від товару, що купується. Або алгоритм прання не залежить від білизни, що преться, і таке інше. Отож, під масовістю.

5. Результативність. Виконання будь-якого алгоритму повинне завершуватися одержанням кінцевих результатів. Тобто ситуації, що в деяких випадках можуть призвести до так званого "зациклення", повинні бути виключені при написанні алгоритму. Наприклад, можна розглянути таку ситуацію: роботу дано завдання залишити кімнату (замкнутий простір), не виконуючи руйнівних дій. У цьому випадку, якщо роботу не дати вказівки відкрити двері (що, можливо, закриті), то спроби залишити приміщення можуть бути безуспішними .

1.3 Форми представлення алгоритмів

Також важливим питанням є подання алгоритму виконавцю. Існує кілька методів запису алгоритмів, вибір яких залежить від виконавця та того, хто його задає.

Перший спосіб - це *словесний опис* алгоритму. Ці алгоритми подаються за допомогою словесного опису.

Як приклад можна розглянути таку задачу: Задача. Дано два вектори $A = ($

a_1 a_2

$, \dots, a_n)$ та $B = (b_1, b_2, \dots, b_n)$. Знайти вектор C , елементи якого

c_i

обчислюються за формулою $c_i = a_i + b_i$.

Приклад словесного опису алгоритму:

1. Покладемо i рівним 1.
2. Покладемо c_i рівним $a_i + b_i$.
3. Перевіримо, чи i дорівнює n . Якщо так, то обчислення припиняємо. Якщо ні, то збільшуємо i на 1 та переходимо до п. 2.

Словесно-формульна форма. Це поєднання формул перетворення інформації та словесного визначення послідовності їх виконання. Використовує загальноприйняті математичні позначення, коментарі до них, що пояснюють дії, та їх послідовність, яка визначається за допомогою міток.

Як приклад можна розглянути задачу з попереднього опису алгоритму.

Розв'язання буде таким:

$i = 1$

Q1 : $c_i =$

Якщо $i = n$, то перейти до Q3 , інакше — до Q2.

Q2 : $i = i + 1$. Перейти до Q1.

Q3 : Закінчити обчислювання.

Граф-схеми. Граф-схема — це представлення алгоритму у вигляді системи точок, кожна з яких визначає дію, та стрілок, які вказують перехід від однієї дії до іншої.

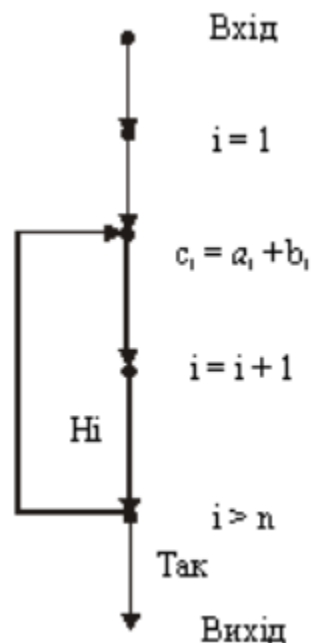


Рис 1.1. Граф-схема алгоритму прикладу

Блок-схеми. Це форма представлення, при якій процес розв'язання задачі поділяється на окремі етапи (або операції), які представляються у вигляді спеціальних блоків, конфігурація яких вказує тип дій. Зв'язки між блоками визначають послідовність цих дій.

Призначення кожного із таких блоків, а також правила їх застосування регламентовані Міждержавним стандартом Єдиної системи програмної документації (ЄСПД).

Розглянувши загальні відомості про алгоритми можна детальніше описати алгоритми з розгалуженнями, про які детальніше буде опис в курсовій роботі.

Команда розгалуження - це складена команда алгоритму, в якій залежно від умови P виконується або одне A, або інше B дія. З команд проходження і команд галуження складаються алгоритми, що розгалужуються. Прикладом алгоритму, що розгалужується, буде знаходження більшого з двох чисел, введених з клавіатури.

Алгоритми з розгалуженням бувають двох різновидів: розгалуження повної форми та розгалуження скорочене. (Рис. 1.2)

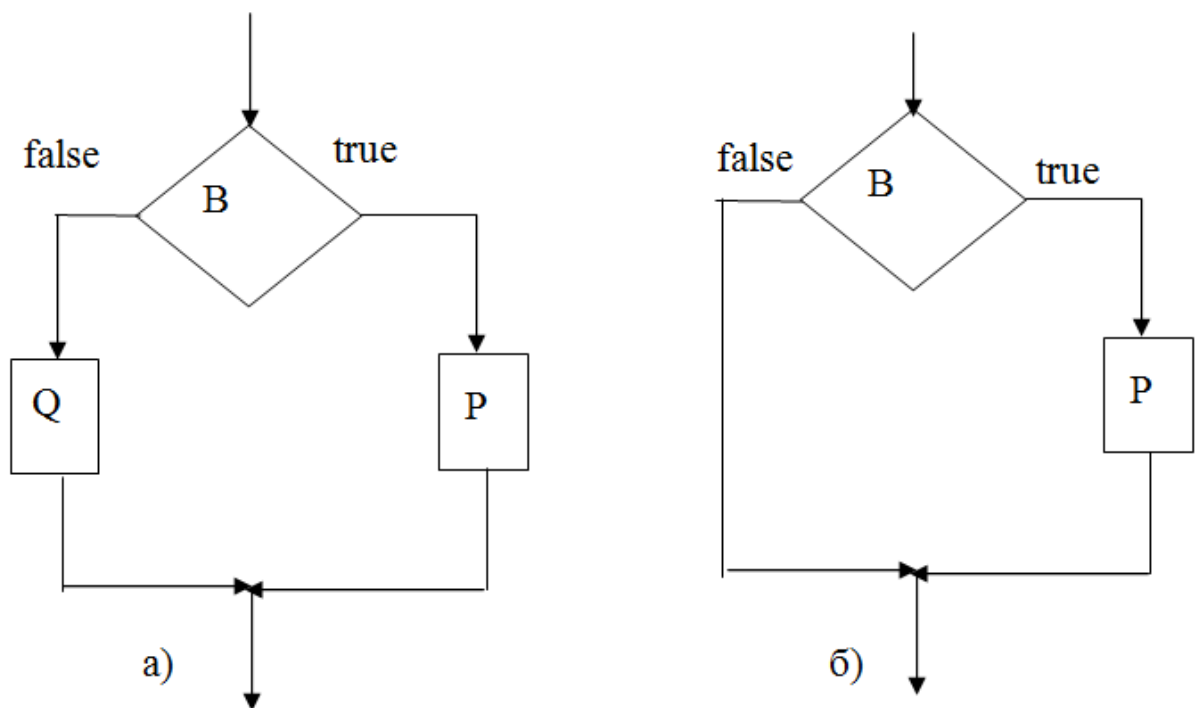


Рис. 1.2. Графічне зображення структур розгалуження:

а – повна форма

б – скорочена

Неповна форма команди галуження використовується тоді, коли необхідно виконувати дію P тільки у разі дотримання умови B. Якщо умова B не дотримується, то команда галуження завершує свою роботу без виконання дії. Прикладом команди галуження неповної форми буде зменшення в два рази тільки парного числа.

II. Методичні основи вивчення розгалужених алгоритмів та побудови скриптів на їх основі.

2.1. Методика ознайомлення з розгалуженими алгоритмами

Ознайомлення з кожною алгоритмічною структурою, в тому числі й з розгалуженням, здійснюється за схожою схемою:

- наведення прикладів алгоритмів відповідної структури,
- з'ясування ключових слів для запису структури та їх позначень на блок-схемах,
- реалізація системи вправ на виконання,
- відшукування та виправлення помилок,
- конструювання та зміна алгоритмів.

З алгоритмічною структурою розгалуження учні знайомляться у 4 класі.

Ситуації з різними умовами, коли залежно від того, чи виконується умова, чи ні, виконавцеві треба приймати різні рішення або виконувати різні команди, трапляються досить часто. Такі ситуації в алгоритмах називають розгалуженням. При графічному способі подання алгоритму, тобто на блок-схемах, команду перевірки умови записують у ромбі. Алгоритмічну структуру розгалуження вчитель може порівняти з деревом, вітки якого розгалужуються. Таке розгалуження відбувається і з командами алгоритму.

Далі можна навести приклади життєвих ситуацій, де потрібно приймати рішення і які можуть бути описані реченнями у формі "якщо ..., то" або "якщо ..., то ..., інакше ...", тобто, структурою розгалуження.

Яскравим прикладом такої структури є алгоритм переходу вулиці на перехресті зі світлофором. Підійшовши до переходу, потрібно зупинитися і подивитися на сигнал світлофора. Якщо сигнал світлофора зелений, то можна переходити вулицю. Інакше (якщо світло червоне), переходити вулицю не можна. Слід обов'язково дочекатися зеленого сигналу світлофора.

Можна навести інший приклад алгоритму з розгалуженням – перехід вулиці без світлофора: 1. Подивитися ліворуч. 2. Якщо немає перешкоди, то йти до середини вулиці, інакше пропустити машини, а потім йти до середини вулиці. 3. Подивитися праворуч. 4. Якщо немає перешкоди, то завершити перехід, інакше – пропустити машини, а потім завершити перехід.

Учням не наголошуються, але на цьому прикладі та на інших демонструються особливості повної та скороченої форм команди розгалуження.

Команди 1 та 3 у цьому алгоритмі називаються простими, а 2 і 4 – складеними.

При вивченні структури розгалуження варто навести приклади, на яких ще уроках учні зустрічаються з нею. Наприклад, на уроках української мови при перевірці написання префіксу с- або з-, написання ненаголошених е та и в корені слова тощо. Застосовуючи відповідне правило, учні кожен раз перевіряють умову. Від результату цієї перевірки залежить правопис слів, що перевіряються. Ось, як виглядатиме блок-схема цього алгоритму.

Ця алгоритмічна структура присутня в деяких казках, наприклад, «Іван-Царевич і сірий вовк», «Півник і двоє мишенят» (Колосок), у мультфільмі «Добриня Микитич і Змій Горинич» та інших. Аналогічно можна виявити, що більшість ігор молодших школярів мають алгоритмічну структуру розгалуження, адже там теж присутні умови які й визначають правила гри.

Лише один виконавець із комплексу навчально-розвивальних програм «Сходінки до інформатики» – Восьминіжка має в своїй системі команд структуру розгалуження, а тому забезпечує виконання алгоритмів з розгалуженням. Також програма у «Скарбниці знань» – Розгалуження із розділу Алгоритми, 4 клас.

У ході виконання вправ та завдань учні вчать аналізувати ситуації, синтезувати, порівнювати та оцінювати результати, що сприяє розвитку алгоритмічного та структурно-логічного мислення школярів.

Приклади завдань, які можна запропонувати виконати учням при вивченні структури розгалуження.

2.2. Методика побудови алгоритмів за схемою логічного слідування

Наступним етапом у вивченні алгоритмів є пояснення способів їх подання та видів. Доцільною є організація бесіди, під час якої вчитель разом з учнями з'ясовує, яким чином, у якому вигляді виконавець отримує алгоритм дій.

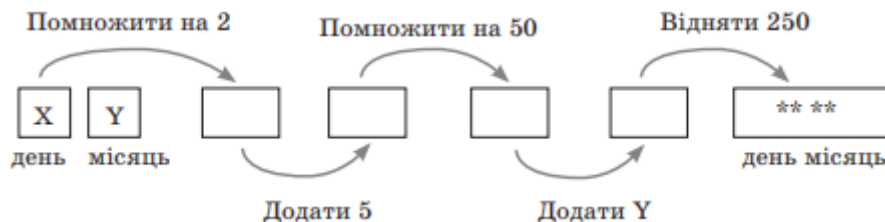
-Діти, давайте згадаємо правило написання великої літери у слові.

-Чи можна це правило назвати алгоритмом? Чому?

-Із яких джерел ви дізналися про цей алгоритм?

-Так, це правило є алгоритмом, оскільки його виконання дозволяє за певну кількість кроків отримати правильну відповідь і визначитися з написанням літери. А дізналися ви про це зі слів учителя та з підручника. У таких випадках алгоритм подано у словесній або текстовій формі.

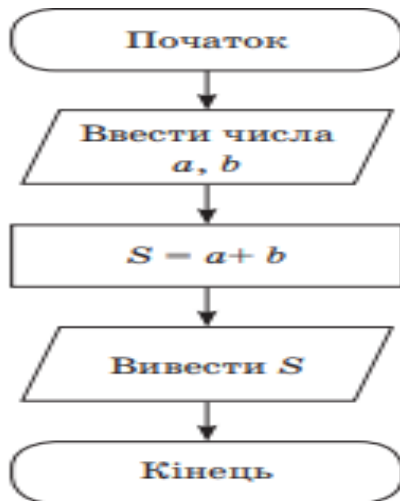
-На схемі зображено алгоритм відгадування дати народження. Цей алгоритм подано у графічній формі.



Графічний спосіб є зручним і зрозумілим. В інформатиці для графічного зображення алгоритмів прийнято використовувати блок-схеми.

Кожний блок має своє загально визначене призначення, зрозуміле людям будь-якої національності: блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних — прямокутником, блок прийняття рішень — ромбом, еліпсом — початок та кінець алгоритму.

Так, наприклад блок-схема алгоритму знаходження суми двох чисел має вигляд:



Саме за допомогою блок-схеми легко пояснити дітям, чому подібні алгоритми називаються лінійними (усі дії виконуються послідовно, одна за одною, наче по лінійці). Далі можна запропонувати учням скласти блок-схеми для різноманітних лінійних алгоритмів («Зліпити Сніговика», «Приготувати бутерброд», «Відправити СМСповідомлення» тощо.).

Уведення понять циклічного та розгалуженого алгоритмів легко сприймаються дітьми за допомогою відомих казок. Так, наприклад, звернення до казки «Колобок» дозволяє вилучити інформаційний блок, що повторюється декілька разів: Колобок котиться доріжкою, зустрічає різних звірів та співає їм пісеньку:

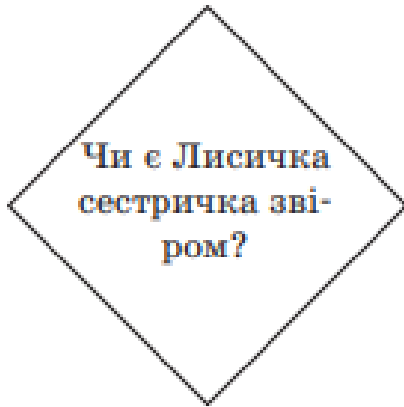
Я по засіку метений,
 Я із борошна спечений,—
 Я від баби втік,
 Я від діда втік,
 То й від тебе втечу!



Як відомо, казка закінчується, як тільки Колобок зустрічає Лисичку-сестричку.

Пояснюємо, що алгоритм, у якому дія повторюється декілька разів називається циклічним. Повторюваність закінчується, як тільки буде виконано умову. З'ясуємо, який саме блок повторюється:

А умовою є зустріч з Лисичкою-сестричкою. Якщо умова не виконана, Колобок зустрічає наступного звіра, співає йому пісеньку та котиться далі. Якщо ж умова виконується, то алгоритм закінчується. Умову позначаємо на блок-схемі у вигляді ромба:



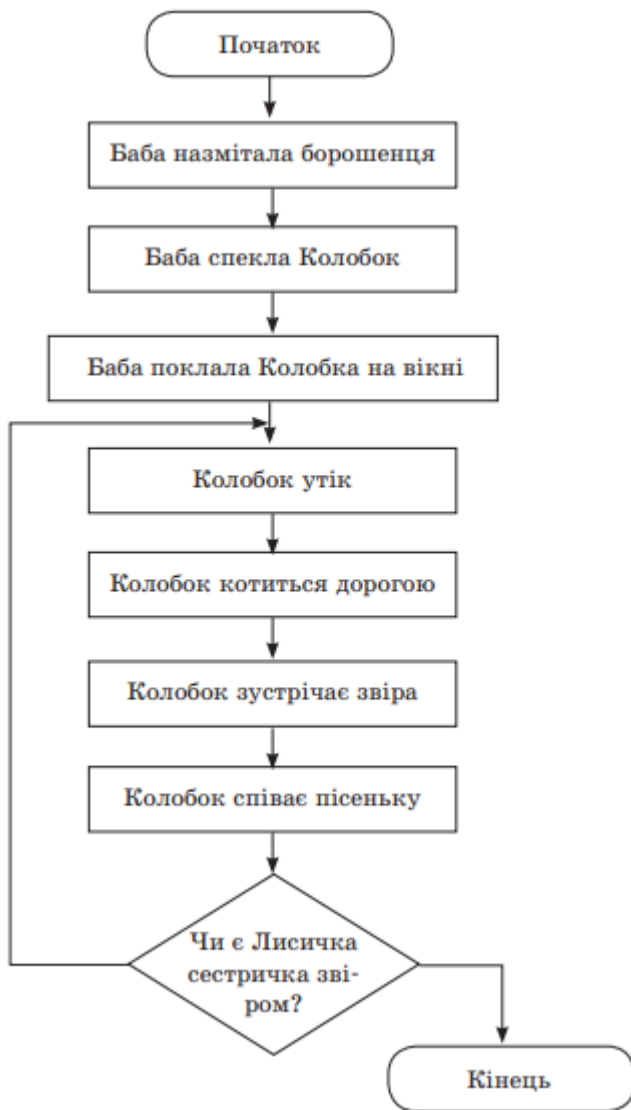
Спробуємо об'єднати всі блоки в єдину схему:

Пропонуємо учням згадати казки, у яких також дії повторюються, поки не виконається казкова умова. Такими казками є «Рукавичка», «Івасик-Телесик» та ін. Алгоритм можна подати як у текстовій формі, так і у графічній.

Пояснюємо учням, що відмінною ознакою циклічного алгоритму є повторюваність виконання дій, поки не буде виконана умова.

Для визначення розгалуженого алгоритму акцентуємо увагу учнів на блоці прийняття рішень, тобто на умові, від якої залежить або повторення циклу, або його завершення.

- Якою геометричною формою позначено на блок-схемі умову? (Ромб)
- А чи можете ви назвати випадки з власного життя, коли від вибору умови залежить наступна дія?
- Робота світлофора впливає на наш вибір переходити дорогу чи зачекати, шкільний дзвоник дозволяє нам виходити на перерву або заходити в клас, перевірка правилом допомагає правильно написати слово тощо.



У першому випадку блок-схема матиме вигляд:



Таким чином, алгоритм, у якому є вибір дії залежно від виконання умови, називається розгалуженим

Акцентуємо увагу учнів на відмінній ознаці розгалуженого алгоритму: якщо умова правильна, то виконується правильна дія. Із цією метою доцільно навести приклади з різних дисциплін з промовлянням саме конструкції якщо-то:

1. Якщо число ділиться на 2, то воно парне.
2. Якщо квітку не поливати, то вона зів'яне.
3. Якщо іменник у реченні відповідає на питання хто? або що?, то він є підметом.

Більшість прислів'їв можна перефразувати у вислів, побудований за конструкцією якщо-то. Це дає можливість використовувати завдання, у яких треба виконати таке перефразування. Наприклад, прислів'я «хто багато читає, той багато знає» перефразуємо в таке: «якщо багато читати, то будеш багато знати». Корисною є й обернена вправа: речення «якщо сила не візьме, то розум допоможе» є перефразуванням відомого прислів'я: «де сила не візьме, розум допоможе».

2.3. Реалізація розгалуження в середовищі Скретч

Скретч (Scratch) – нова мова та візуальне середовище об'єктно – орієнтованого програмування, яка дозволяє легко створювати власні проекти: анімаційні та інтерактивні історії, ігри, моделі та розміщувати їх в мережі Інтернет.

Виконацем цього алгоритму є Рудий кіт (також є й інші персонажі) , для якого існує власна система команд. З кожною командою пов'язана певна подія, яка відтворюється на сцені. Виконавців в середовищі Scratch називають спрайтами.

Команди в середовищі Scratch можна обрати в контейнері (Рис.2.2.1). Команди також об'єднані в групи: руху, зміни вигляду, малювання, відтворення звуку і інші [11].

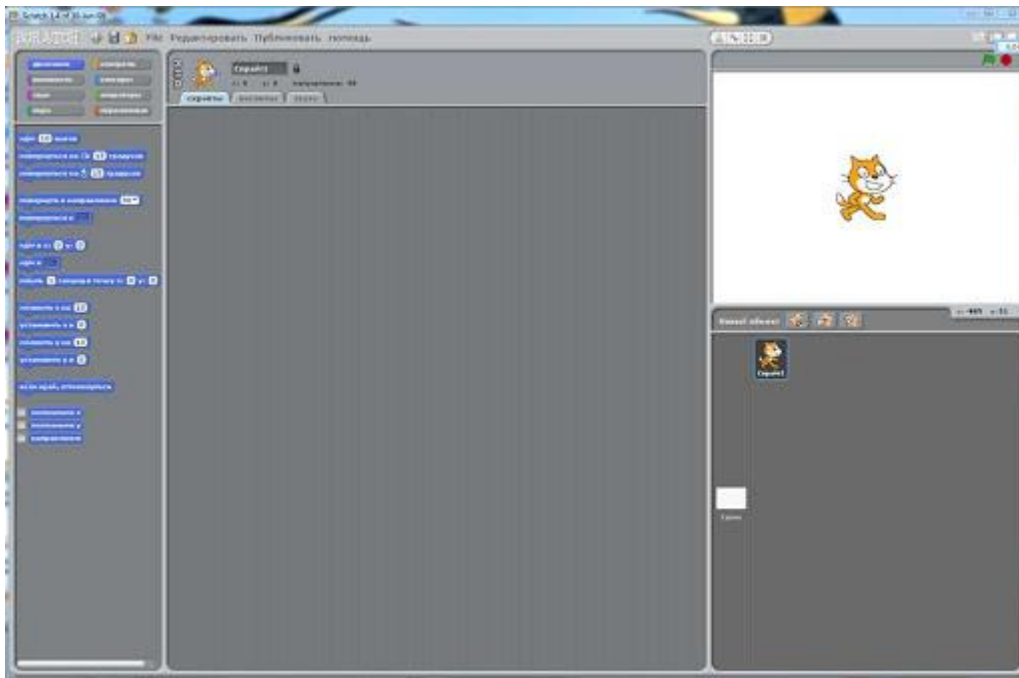


Рис 2.2.1. Вікно програмування програми Scratch

Рудий кіт на білому полі вікна – це виконавець програми. Він розташований у центрі сцени і може рухатися на 180 кроків угору або вниз і на 240 кроків вправо або вліво.

Цікавий факт про середовище програмування Scratch

Мову програмування Scratch створено в 2007 році в лабораторії Lifelong Kindergarten Массачусетського технологічного університету під керівництвом професора Мітчела Резника та з фінансовою підтримкою великих компаній таких як Science Foundation, Microsoft, LEGO Foundation, Google, Dell, Fastly та інші великі компанії. Scratch надає можливість дітям навчитися основ програмування та відкриває їм шлях до професії програміста. На сьогоднішній день багато дітей з всього світу із задоволенням створюють свої проекти в середовищі Scratch і розміщують їх в мережі Інтернет. Також часто проходять конкурси на кращу роботу [12].

Для того щоб детальніше ознайомитися з роботою в середовищі програмування Scratch можна розглянути декілька задач на розгалуження та порядок їх виконання [13].

Приклад 1. Вправа: Голодна акула



Акула рухається з відкритим ротом по сцені, слідує за показником миші на відстані 10 кроків. Рибки з невеликим інтервалом з'являються в лівій частині сцени і рухаються зліва направо, повертаються випадковим чином, відштовхуються від краю. Якщо рот акули торкається тіла рибки, то вона її з'їдає, при цьому видає звук «ам» і відкриває-закриває рот, а рибка ховається і після паузи з'являється у випадковій точці лівої частини сцени.

ВИКОНАННЯ

При виконанні завдання передбачається використання кінцевого циклу і датчика випадкових чисел. На рис. 7 пропонується послідовне виконання двох скриптів для акули, на рис. 8 - скрипт рибки. Скрипт створюється для однієї рибки, а потім дублюється спрайт рибки 2 рази разом з сценарієм. Фон сцени вибирається з бібліотеки.

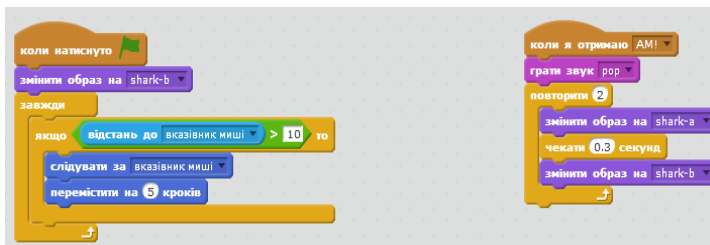


Рис. 7

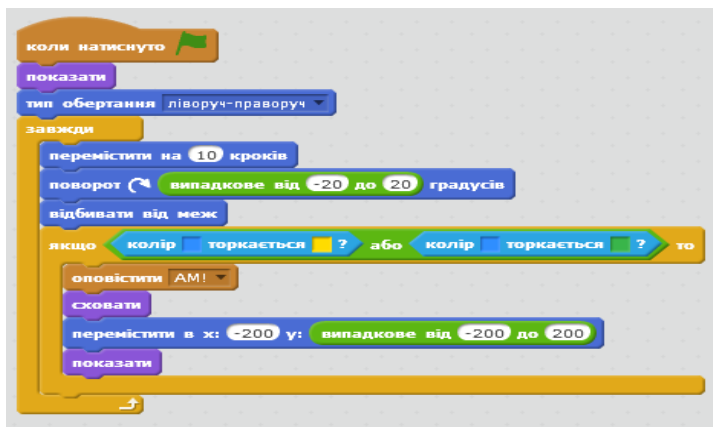
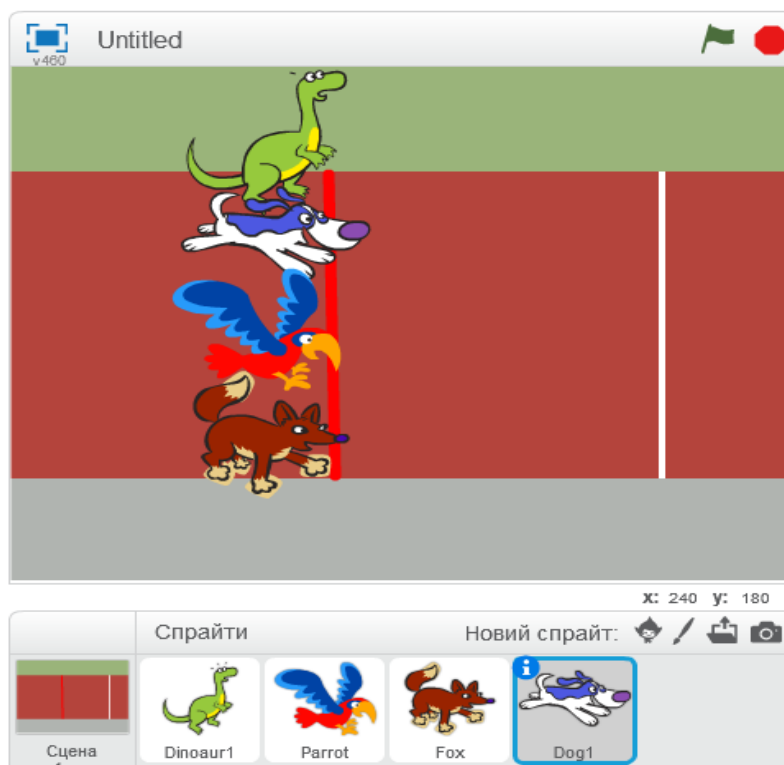


Рис. 8

Для того, щоб рибка рухалася зліва направо і поверталася випадковим чином виконується команда «рух» разом з оператором датчика випадкових чисел: "повернути за годинниковою стрілкою на випадкове від -20 до 20 градусів". Аналогічним чином для появи рибки після паузи у випадковій точці лівої частини сцени - "перейти в x: -200 у: видати випадкове від -200 до 200".

Приклад 2. Вправа: Хто перший?



Відстань від старту до фінішу 250 кроків. Всі учасники змагання: динозавр, папуга, лисиця і собака спочатку розташовані на лінії старту. У кожного спрайту виконується свій скрипт, під час дії команди "коли клавішу пропуск натиснуто".

При досягненні фінішу переможець каже: "Я - перший!" і всі учасники зупиняються. Хто (динозавр, папуга, лисиця або собака) швидше досягне фінішу?

ВИКОНАННЯ

Для вирішення завдання малюються лінія Старту і лінія Фінішу. На лінії старту розташовуються спрайти.

Пишеться скрипт для кожного учасника змагання (рис. 10-13). Дописуються скрипти з однаковими заголовками (подія почнеться "коли клавішу пропуск натиснуто") відповідно до умови завдання.

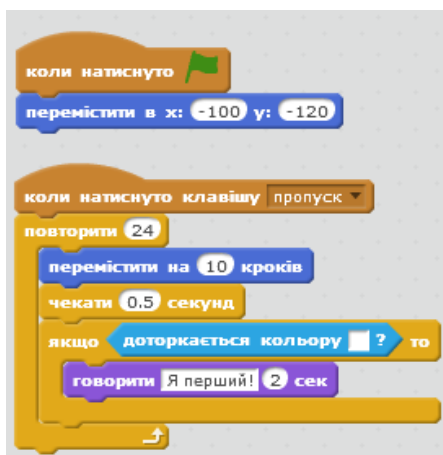


Рис. 10 (динозавр)

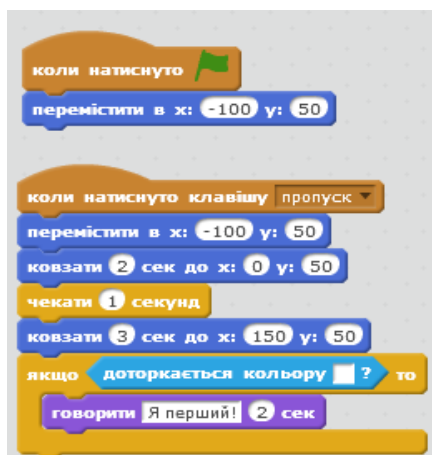


Рис. 11 (папуга)



Рис. 12 (лисиця)

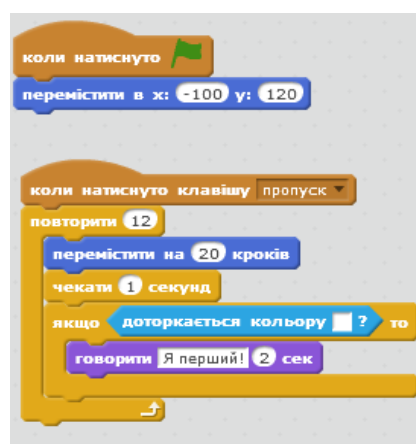


Рис. 13 (собака)

Приклад 3. Вправа: Квадрати, які обертаються

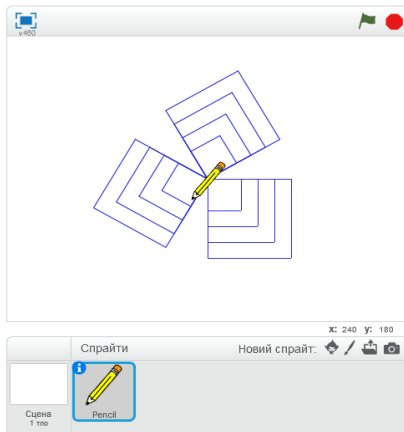


Рис. 24

Використовуючи процедури з параметрами, намалюйте квадрати, що обертаються.

ВИКОНАННЯ

Для вирішення завдання використовуються дві процедури з параметром і одна процедура без параметра. У процедурі Квадрат параметр "сторона" приймає значення: 40, 60, 80 і 100. Виклик процедури здійснюється в процедурі Квадрати, де всі значення передаються параметру послідовно (рис. 25-27).

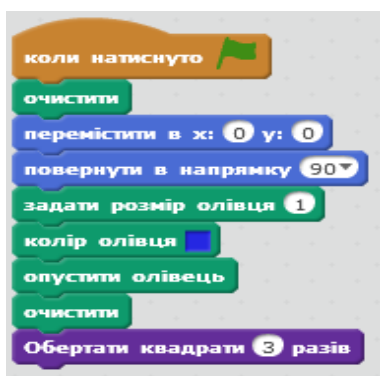


Рис. 25

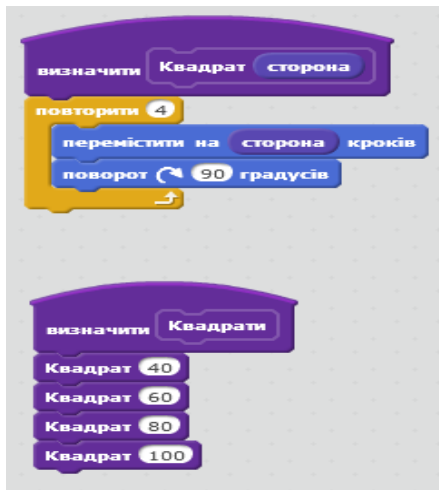


Рис. 26



Рис. 27

У процедурі *Обертати квадрати* значення параметра "число" використовується як дільник в операторі поділу команди "повернути за годинниковою стрілкою на $360 / \text{число}$ градусів" і як змінна кінцевого циклу.

Розглянувши детальніше теоретичні основи про розгалужені алгоритми, їх будову, властивості, застосування та розв'язання за допомогою їх задач можна перейти до створення ігрових задач за допомогою алгоритмів в середовищі Scratch.

Відкривши в мережі Інтернет сайт з програмою Scratch (в даному випадку перейти за посиланням <https://scratch.mit.edu/>) потрібно перейти на головну сторінку. Наступним кроком буде створення нового спрайту.

Прикладом буде гра «Чарівна куля», яка буде давати випадкову відповідь (так або ні).

Для цього додаємо новий спрайт «Куля»(рис.2.3.1)

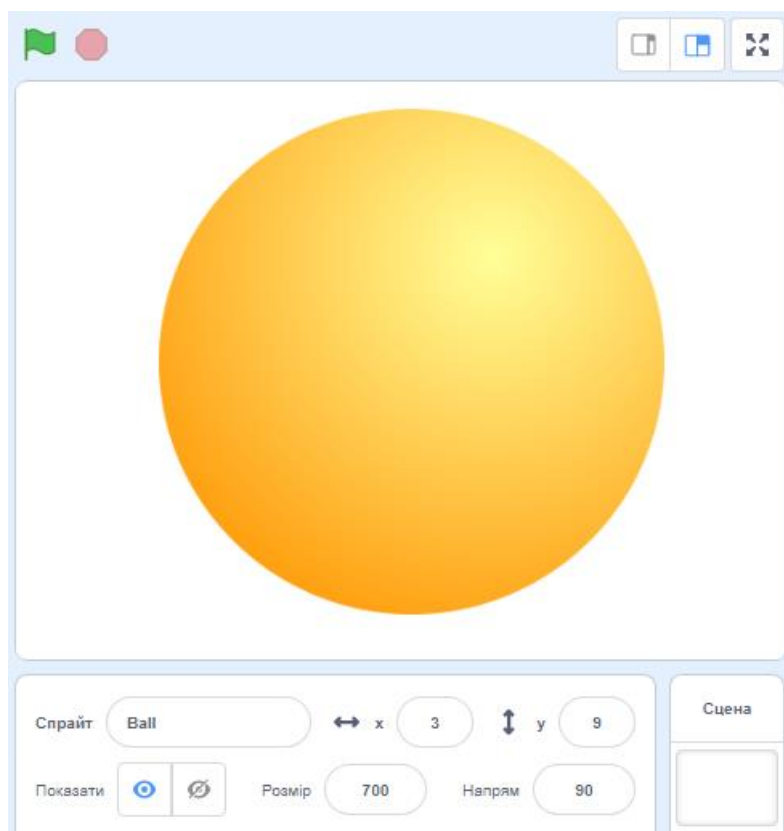


Рис.2.3.1. Спрайт «Куля»

Потім потрібно буде створити скрипт, який при кліканні на кулю сповщає подію „Update”, рухає її декілька разів вправо-вліво, а потім повертає в вихідне положення.

Наступним кроком буде намалювати на кулі новий спрайт – прямокутник(рис.2.3.2).

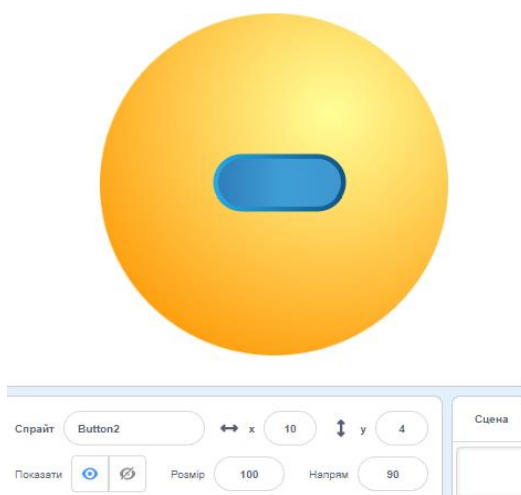


Рис.2.3.2. Куля з новим спрайтом

В нові сині спрайти необхідно внести два варіанти відповідей «Так» або «Ні».

Передостаннім кроком буде створення невидимої змінної „Answer”.

Останнім кроком буде створення нового скрипту, що починається коли отримано подію „Update”, який спочатку ховає кулю, потім чекає 3 секунди та надає змінній „Answer” випадкове значення 1 або 2, яке відповідає значенням «так» або «ні» відповідно і видає образ який випадково випав (Рис.2.3.3)

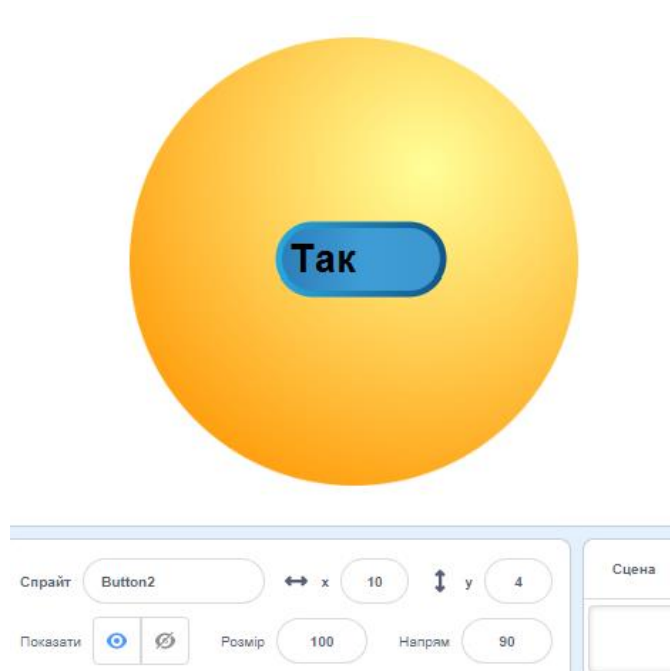


Рис.2.3.3 «Чарівна куля»

2.4. Методика створення дидактичних матеріалів для теми «Алгоритми з розгалуженням»

Дидактичні матеріали до уроку інформатики:

Вправа 1

Назва вправи: Розкодує прислів'я.

Завдання: Виконай алгоритм і розкодує прислів'я.



Зміст вправи: Скласти правильне прислів'я. (Карточки з завданням роздруковую і пропоную виконати у парах на швидкість).

Тип вправи: фрагмент зображення.

Методично-дидактичний аспект використання: дану вправу пропоную виконати учням на етапі узагальнення і систематизації набутих знань з метою відпрацювання навичок виконання алгоритмів з розгалуженням.

Практична цінність: перевіряє вміння учнів виконувати алгоритми, розвиває алгоритмічне та логічне мислення, швидкість, формує вміння працювати в парі.

Вправа 2

Назва вправи: Лабіринт.

Завдання: Допиши алгоритм проходження лабіринту.



Зміст вправи: Дописати алгоритм. (Карточки з завданням роздруковую і пропоную виконати самостійно)

Тип вправи: фрагмент зображення

Методично-дидактичний аспект використання: дану вправу пропоную виконати учням на етапі узагальнення і систематизації набутих знань з метою відпрацювання навичок виконання алгоритмів з розгалуженням.

Практична цінність: дана вправа перевіряє вміння учнів виконувати алгоритми, розвиває алгоритмічне та логічне мислення, прищеплює навички самостійної роботи.

Вправа 3

Назва вправи: Слідопит.

Завдання: Знайдіть терміни, яки ви вивчили з теми «Алгоритми з повтореннями» використовуючи підказки.



1. _____ Базова алгоритмічна, структура призначена для організації багаторазового виконання набору команд.
2. _____ Змінна, значення якої визначає кількість повторень команд тіла циклу.
3. _____ Послідовність команд.
4. _____ Одна або більше команд, які повторюються декілька разів.

Зміст вправи: Знайти терміни, які були вивчені з теми «Алгоритми з повтореннями» та записати на місці пропусків. (Картку з завданням роздруковуємо і пропонуємо виконати у парах)

Тип вправи: фрагмент зображення.

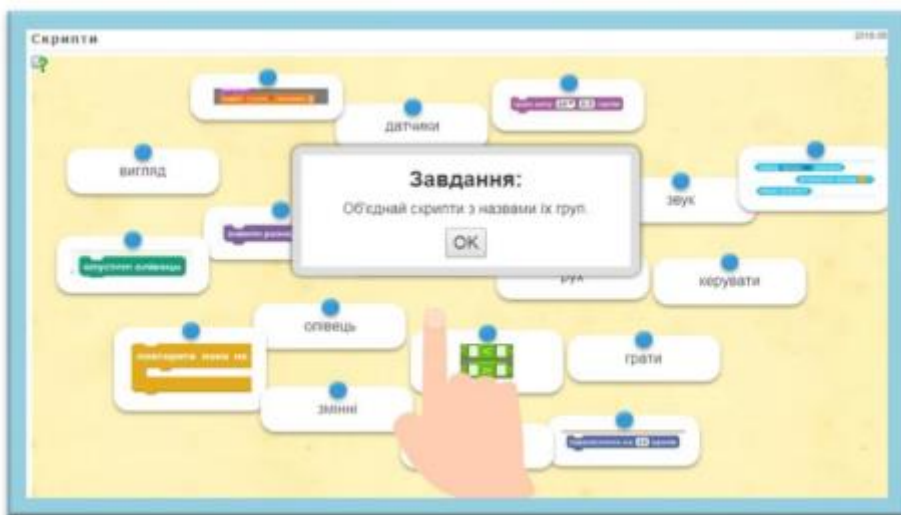
Методично-дидактичний аспект використання: дану вправу пропонуємо виконати учням на етапі узагальнення і систематизації набутих знань з теми «Алгоритми з повтореннями».

Практична цінність: перевірити знання учнів, розвивати пізнавальний інтерес, алгоритмічне мислення, пам'ять, уважність.

Вправа 4

Назва вправи: Скрипти

Завдання: Об'єднай скрипти з назвами їх груп.



Зміст вправи: об'єднати скрипти і назви їх груп

Тип вправи: знайди пару.

Методично-дидактичний аспект використання: дану вправу пропоную виконати учням на етапі актуалізації з метою перевірки та повторення призначення груп і скриптів.

Практична цінність: перевіряє та закріплює знання учнів.

Висновок

У процесі написання курсової роботи «Методичні основи ознайомлення молодших школярів з побудовою розгалужених алгоритмів» було виконано всі завдання, які були перелічені в вступі, а саме: було ознайомлено з основною теорією алгоритмів, їх властивостями, застосуванням, та з методикою та прикладами задач в середовищі Scratch, розглянуто розв'язання типових задач в середовищі Scratch та було зроблено та описано поетапно створення елементарної ігрової задачі в середовищі Scratch на офіційному сайті розробника.

За даними курсової роботи можна зробити такі висновки:

1. Алгоритми - система правил, сформульована на зрозумілій виконавцеві мові, що визначає процес переходу від припустимих вихідних даних до деякого результату й має властивості масовості, кінцівки, визначеності, детермінованості, які повинні обов'язково виконуватися для того щоб отримати точний і чіткий кінцевий результат.

2. Будь-який алгоритм обов'язково має відповідати певному методу побудови і має бути на основі нього побудований, в курсовій роботі було розглянуто декілька основних методів, таких як: метод частинних цілей; динамічне програмування; метод сходження; дерева розв'язків; програмування з поверненнями назад; евристичні алгоритми.

3. Створюючи елементарні задачі в середовищі Scratch можна зробити висновок, що кожна наступна дія має бути чітка і правильна для того щоб в результаті роботи отримати правильний і точний кінцевий результат.

Список використаних джерел

1. Базові алгоритмічні структури [Електронний ресурс] – Режим доступу до ресурсу: <https://dl.sumdu.edu.ua/textbooks/108989/459153/index.html>.
2. Руденко В. Алгоритмізація і програмування / Віктор Руденко.. – 128 с.
3. Бородкіна І. Теорія алгоритмів. Посібник для студентів вищих навчальних закладів / Ірина Бородкіна., 2018. – 184 с.
4. АЛГОРИТМИ ШТУЧНОГО ІНТЕЛЕКТУ. [Електронний ресурс] – Режим доступу до ресурсу: http://mmsa.kpi.ua/sancho/ASD_HTM/Arti01.html.
5. Динамічне програмування [Електронний ресурс] – Режим доступу до ресурсу: https://pidruchniki.com/14250725/menedzhment/dinamichne_programuvannya.
6. Методи розробки алгоритмів [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wikim_Методи_розробки_алгоритмів.
7. Методи побудови алгоритмів [Електронний ресурс] – Режим доступу до ресурсу: http://choippo.cn.sch.in.ua/navchaljni_temi/metodi_pobudovi_algoritniv/.
8. Обговорення користувача : Марців Віктор [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Обговорення_користувача:Марців_Віктор.
9. Застосування евристичних алгоритмів [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/3751878/>.
10. Евристичні алгоритми [Електронний ресурс] – Режим доступу до ресурсу: http://mmsa.kpi.ua/sancho/ASD_HTM/Arti04.html.
11. Середовище виконання алгоритму [Електронний ресурс] – Режим доступу до ресурсу: <https://www.slideshare.net/olenapyshchita/ss-39904981>.
12. Як працювати в середовищі програмування скретч [Електронний ресурс] – Режим доступу до ресурсу: <https://www.slideshare.net/volodiamigasyuk/3-27-73323679>.
13. Морзе Н. В. Інформатика / Н. В. Морзе, О. В. Барна, О. Г. Кузьмінська. – Київ, 2017.

Додатки

План-конспект уроку

Тема: Алгоритми з розгалуженням. Складання та виконання алгоритмів з розгалуженням в середовищі Scratch.

Мета:

Навчальна: навчити а розрізнати структури розгалуження в алгоритмах із життя та навчальної діяльності, вивчати запис структури розгалуження у вигляді блок-схеми; навчити учнів записувати алгоритм з розгалуження у вигляді послідовності команд виконавця у середовищі Scratch;

Виховна: виховувати пізнавальну зацікавленість до предмету та привчати працювати творчо, виховувати старанність, активність при вивченні нового матеріалу;

Розвивальна: розвивати пізнавальний інтерес, навички роботи з середовищем виконання алгоритму; алгоритмічне мислення, пам'ять, уважність.

Тип уроку: комбінований.

Обладнання та наочність: ПК із встановленими ОС і середовищем Scratch, презентація до уроку, інструктивна карта для виконання практичного завдання

Хід уроку

I. Організаційний етап.

Вітання з класом. Перевірка присутності і готовності учнів до уроку.

II. Актуалізація знань

1. Що таке алгоритм (Алгоритм — це запис скінченої послідовності вказівок, що визначає, які дії і в якому порядку потрібно виконати для досягнення певної мети).
2. Хто є виконавцем алгоритму? (Виконавець алгоритму — жива істота (людина чи тварина) або автоматичний пристрій (робот, верстат з програмним

керуванням, електронна обчислювальна машина тощо), що спроможна діяти і діє згідно з наданим алгоритмом).

3. Що таке система команд виконавця? (Система команд виконавця — сукупність усіх команд, які може виконувати даний виконавець).
4. Що таке комп'ютерне середовище виконання алгоритму? (Комп'ютерне середовище виконання алгоритму – спеціальна програма, яка дає змогу створювати і виконувати алгоритми для обраних виконавців із визначеною системою команд на комп'ютері).
5. Гра «Світлофор». Визначити істинне чи хибне висловлювання
Якщо висловлювання істинне - зелена картка, якщо хибне – червона
 - 1) У лісах збирають солодкі кавуни. (хибне)
 - 2) Наступний день після неділі – понеділок (істинне)
 - 3) Місяць грудень – рік починає (хибне)
 - 4) Якщо до двох додати три, то отримаємо шість. (хибне)
 - 5) Рік має 12 місяців (істинне)
 - 6) Сонце обертається навколо Землі (хибне)
 - 7) Тарас Шевченко – великий український поет

III. Мотивація. Оголошення теми, мети уроку.

Розгадайте ребус



Сьогодні ми познайомимось із однією із базових алгоритмічних структур – структурою розгалуження.

IV. Вивчення нового матеріалу

Пояснення вчителя з елементами демонстрування презентації (використовуються можливості локальної мережі кабінету або проектор)

В такому простому алгоритмі в зразу ж знайдете недоліки. А що робити, якщо виконавець себе погано почуває (захворів), а якщо вже зібрав речі ввечері, а

якщо не встиг напередодні вивчити всі уроки і необхідно щось повторити, а що значить одягнути верхній одяг (він залежить від пори року, погоди тощо).

Якщо ж спробувати прослідкувати за вашою поведінкою впродовж дня, то з'ясується, що майже ніколи ви не дієте за лінійним алгоритмом. Весь час ви аналізуєте ситуацію, змінюєте свою поведінку та свої плани, пристосовуєтесь до обставин.

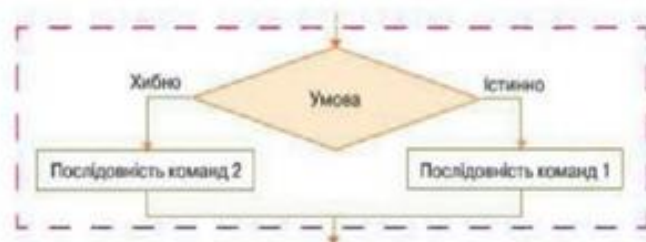
Тому набагато частіше зустрічається другий тип алгоритму - розгалужений. Цей алгоритм обов'язково містить в собі хоча б одну умову і виконується він в залежності від цієї умови. Наприклад, алгоритм переходу дороги по пішохідному переходу, який регулюється світлофором: якщо горить зелене світло, слід переходити дорогу, в іншому разі — слід зупинитися перед пішохідним переходом.

Алгоритмом із розгалуженням можна вважати алгоритм здійснення дзвінка з мобільного телефону: якщо є кошти на рахунку і мережа доступна, то ти набираєш номер адресата, інакше виклик не буде здійснено.

При побудові таких алгоритмів використовують алгоритмічну структуру розгалуження.

Алгоритмічна структура, що дає змогу виконавцеві алгоритму вибрати сценарій подальших дій залежно від істинності певного висловлювання, називається розгалуженням.

Структура розгалуження повної форми схожа на умовне висловлювання «Якщо — то — інакше», у якому після «то» та «інакше» записують не висловлювання, а команди, які необхідно виконати залежно від істинності висловлювання, записаного в умові. Її можна подати графічно:



Наприклад, алгоритм написання слів із великої літери:

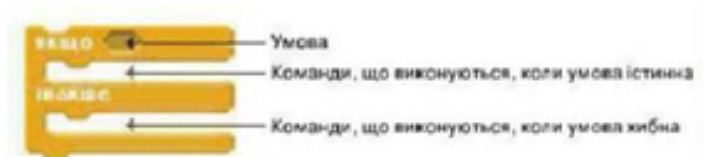


Для опису алгоритмів із розгалуженням у середовищі Scratch можна використовувати блоки з групи **Датчики**. Ці блоки можуть бути складовими відповідних команд, що передбачають виконання певних дій після перевірки висловлювання на істинність чи хибність.

Складені умови, які використовують сполучники І, АБО, НЕ можна описати за допомогою блоків із групи **Оператори**.

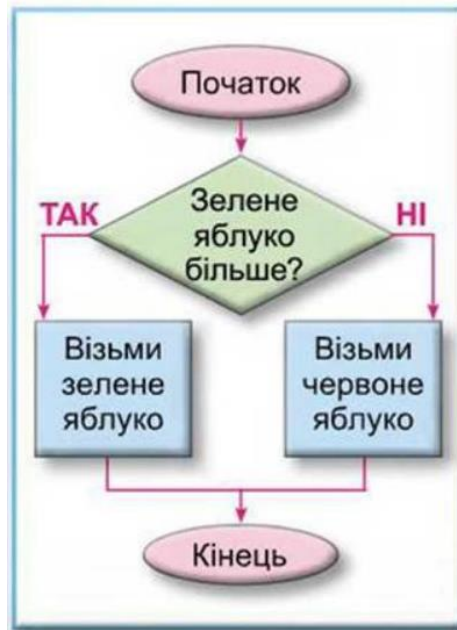


Базову структуру розгалуження в середовищі Scratch можна подати командою



V. Засвоєння нових знань

Завдання. Маємо два яблука – червоне і зелене. Скласти алгоритм, за допомогою якого слід обрати більше серед цих яблук.



VI. Фізкультхвилинка

Сильний вітер до землі
Гне дерева молоді.
А вони ростуть, міцніють,
Вгору тягнуться, радіють.
Вітер віє нам в лице;
Захиталось деревце.
Вітерець все тихше, тихше,
Деревце все вище, вище!

VII. Формування вмінь та навичок

Робота за комп'ютером-

- 1) Інструктаж учителя.
- 2) Практична робота за комп'ютерами.
- 3) Вправи для очей.

Практичне завдання

Завдання. Створити скрипт, в якому Рудий Кіт запитує учня приклад на множення. Якщо учень відповідає вірно, Кіт говорить: «Молодець!», а якщо відповідь невірна, Кіт говорить: «Вивчи таблицю множення!»

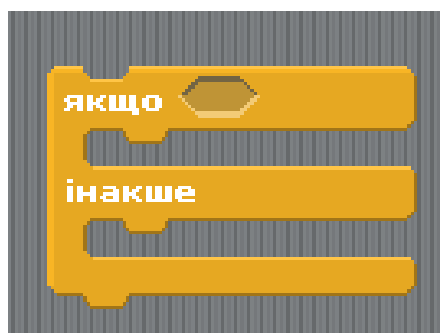
Технологія виконання завдання.

Згадайте правила безпечної роботи з комп'ютером.

1. Завантажте середовище Scratch.
2. У вкладці Змінні створіть дві змінні a і b. Надайте їм значення випадкових чисел від 1 до 10, скориставшись вкладкою Оператори.
3. Для запитання Кота скористайтесь вкладкою Датчики.



4. Створіть повне розгалуження із вкладки Керувати



5. Умову складіть із блоків вкладок Оператори, Змінні і Датчики.
6. Створіть скрипт за заданим сценарієм:



7. Перевірте правильність виконання програми.
8. Завершіть роботу в середовищі Scratch.

9. Зробіть висновок чи навчилися ви створювати повне розгалуження у середовищі Scratch.

План-конспект уроку

Тема: Алгоритми з розгалуженнями

Мета роботи:

формувати предметні компетентності: розуміння сутності поняття розгалуження; Розуміння відмінностей між повним та неповним розгалуженнями; уміння складати, виконувати, змінювати алгоритми з розгалуженнями;

розвивати ключові компетентності:

- Компетентності в галузі природничих наук, техніки й технологій (застосування логічного, алгоритмічного, структурного та системного мислення для розв'язування життєвих проблемних ситуацій);
- Інформаційно-комунікаційна компетентність (уміння подавати алгоритми в певному формальному вигляді та виконувати їх; використовувати в алгоритмах алгоритмічну структуру розгалуження);
- Навчання впродовж життя (розвиток алгоритмічного мислення: вміння визначати послідовність дій, які необхідно виконати для розв'язання певної задачі; обирати оптимальний алгоритм розв'язання задачі);

Очікувані результати: учні повинні уміти складати, налагоджувати, тестувати програми з розгалуженням.

Обладнання: комп'ютер зі встановленим середовищем програмування Scratch 2, презентація до уроку, проектор та екран, картки із самостійною роботою, чисті аркуші паперу.

Тип уроку: засвоєння нових знань та вмінь.

Хід уроку

1. Організаційний етап.

- a. перевірка готовності учнів до уроку
- b. привітання.

2. Перевірка домашнього завдання

Слайд 1.

Інтерактивна вправа «Команди і виконавці»

- Середовище
- Алгоритм
- Скрипт
- Виконавець
- Команда
- Спрайт
- Проект
- Скретч

Слайд 2.

Вправи для очей. Рухи очима вгору, вниз, ліворуч, праворуч та по колу. Закрити очі та порахувати до 10.

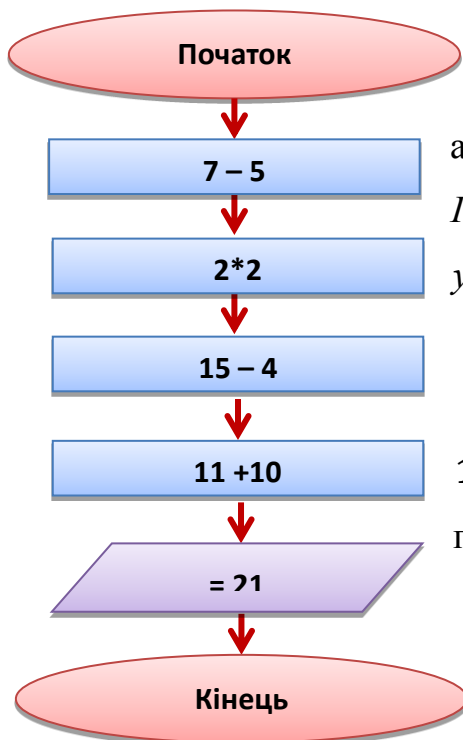
3. Актуалізація опорних знань

Аналіз самостійної роботи

Слайд 3.

1. Дайте визначення таким поняттям:

1. Команда	Г) це повідомлення, яке спонукає до виконання певної дії
2. Виконавець	А) об'єкт, здатний виконати задані йому команди.
3. Система команд виконавця	Б) набір команд, які може виконати даний виконавець.
4. Алгоритм.	В) це скінченна послідовність команд, виконання яких приводить до розв'язання поставленої задачі.



2. У якій формі (яким способом) можна подати

алгоритм?

Графічно (блок-схеми, рисунок), словесно (письмово, усно)

3. Подай алгоритм обчислення значення виразу

$15 - (7 - 5)^2 + 10$ у вигляді блок-схеми (графічним способом) та обчисліть його.

4. Мотивація навчальної діяльності.

Вправа « Логічні висловлювання»

Слайд 4 – 8.

оголошення теми уроку. Слайд 9

5. Вивчення нового матеріалу.

Слайд 10

Алгоритмічна структура, що дає змогу виконавцеві алгоритму вибрати сценарій подальших дій залежно від істинності певного висловлювання (умови), називається **розгалуженням**.

Слайд 11

Умовою називають логічне висловлювання, з якого починають будувати алгоритмічну структуру розгалуження. На блок-схемах умову записують у блоці, що має форму ромба.

Розглянемо приклад запропонований на слайді 12

В алгоритмах використовують розгалуження двох видів: **повне розгалуження і неповне розгалуження**.

Слайд 13.

Виконання повного розгалуження відбувається так: виконавець виконує **команду перевірки умови**: якщо результат виконання цієї команди **Так**, то виконавець виконує **послідовність команд 1**, після чого переходить до виконання першої команди наступного фрагмента алгоритму; якщо результат виконання цієї команди **Ні**, то виконавець виконує **послідовність команд 2**, після чого також переходить до виконання першої команди наступного фрагмента алгоритму.

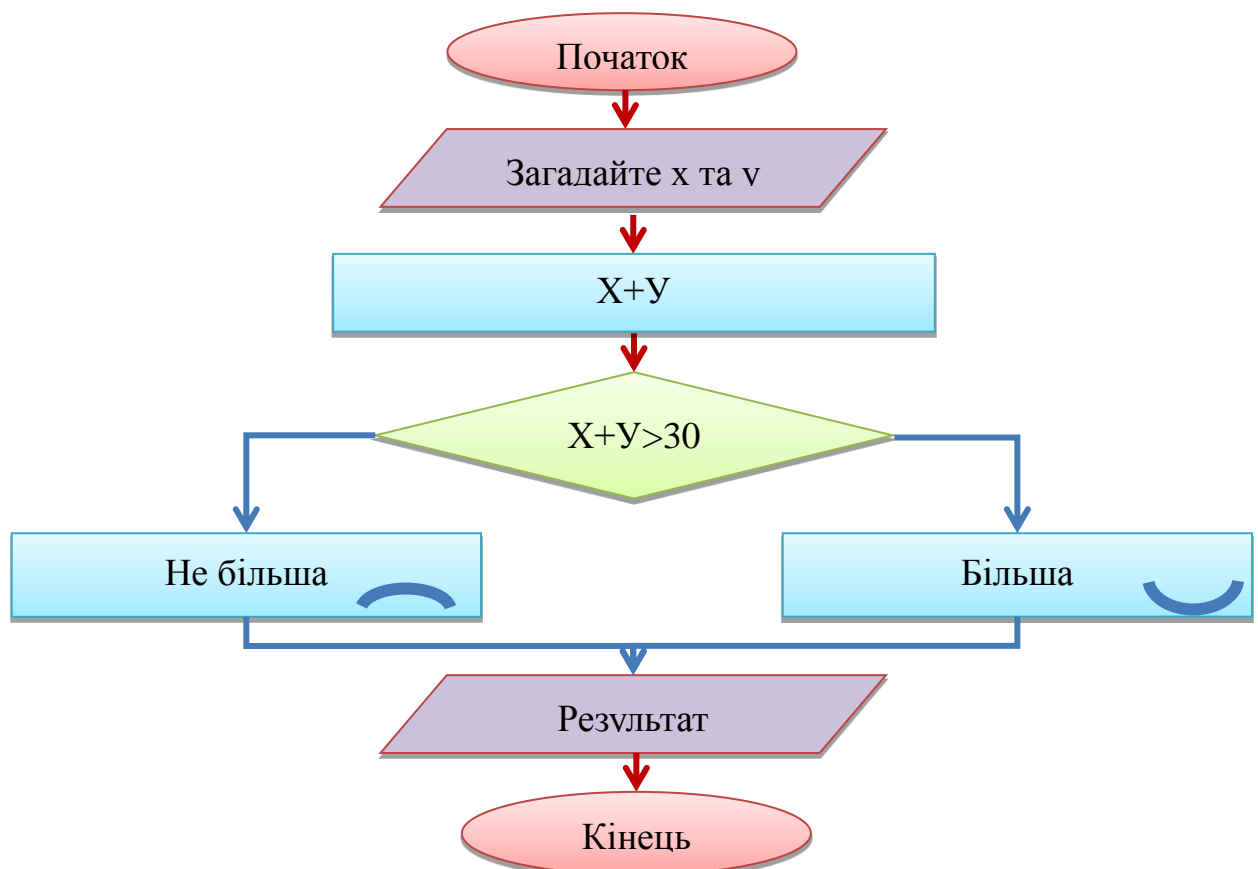
Слайд 14. Завдання на повне розгалуження

Слайд 15

Виконання неповного розгалуження відбувається так: виконавець виконує **команду перевірки умови**: якщо результат виконання цієї команди **Так**, то виконавець виконує **послідовність команд 1**, після чого переходить до виконання першої команди наступного фрагмента алгоритму; якщо результат виконання цієї команди **Ні**, то виконавець переходить до виконання першої команди наступного фрагмента алгоритму.

Слайд 16 Завдання на неповне розгалуження

6. Закріплення знань. Практична частина.

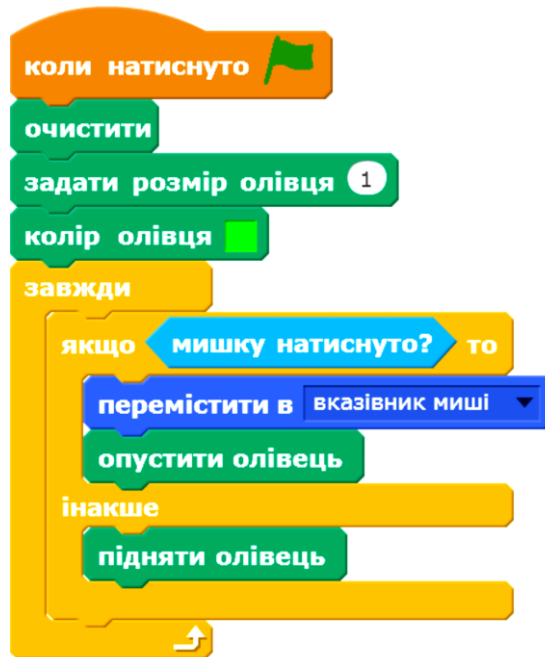


Виконання завдань з підручника с.169 № 3;5.

Слайд 17. Робота за комп'ютером

- 1) Повторення правил безпечної поведінки за комп'ютером.
- 2) Інструктаж учителя.
- 3) Практична робота за комп'ютерами.

Вправи для очей.



7. Підсумок уроку.

Слайд 18 Виконайте «Загадкову інструкцію»

Обговорення результатів.

Яке розгалуження було у вправі? Якою командою в середовищі Скретч воно задається?

Який ще типи розгалуження вам відомий? Якою командою в середовищі Скретч воно задається?