

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**О. С. Пшенична,  
Н. М. Д'яченко,  
І. В. Красікова,  
О.О. Тітова**

**ВІЗУАЛЬНЕ ПРОГРАМУВАННЯ**

**Методичні рекомендації до лабораторних занять  
для здобувачів ступеня вищої освіти бакалавра  
спеціальності «Середня освіта» освітньо-професійної програми  
«Середня освіта (Інформатика)»**

Затверджено  
вченою радою ЗНУ  
Протокол №\_\_ від \_\_\_\_\_

Запоріжжя  
2025

УДК 004.4'236(076.5)  
П 932

Пшенична О. С., Д'яченко Н. М., Красікова І. В., Тітова О. О. Візуальне програмування : методичні рекомендації до лабораторних занять для здобувачів вищої освіти бакалавра спеціальності «Середня освіта» освітньо-професійної програми «Середня освіта (Інформатика)». Запоріжжя : Запорізький національний університет, 2025. 83 с.

У методичних рекомендаціях подано в систематизованому вигляді короткий теоретичний матеріал та зміст лабораторних робіт з дисципліни «Візуальне програмування». Основна увага приділяється засвоєнню знань про візуальні середовища розробки Scratch та Google Blockly, принципи написання програм, використання відповідних блоків. До кожної теми підібрано теоретичний матеріал, надано рекомендації до виконання лабораторних робіт, наведено приклади написання програм, запропоновано запитання для самоперевірки. Надано список рекомендованої літератури. Довідковий матеріал міститься в додатку.

Методичні рекомендації призначені для здобувачів ступеня вищої освіти бакалавра спеціальності «Середня освіта» освітньо-професійної програми «Середня освіта (Інформатика)», однак можуть використовуватися й студентами інших спеціальностей та працівниками освіти.

Рецензент

*К. С. Решевська*, кандидат технічних наук, доцент кафедри комп'ютерних наук

Відповідальний за випуск

*Г. М. Шило*, доктор технічних наук, професор, завідувач кафедри комп'ютерних наук

---

## ЗМІСТ

|                                                                                                |    |
|------------------------------------------------------------------------------------------------|----|
| Вступ                                                                                          | 5  |
| Загальні рекомендації до виконання лабораторних робіт                                          | 7  |
| Лабораторна робота №1 Тема: Основи роботи в середовищі Scratch                                 | 8  |
| Теоретичні відомості                                                                           | 8  |
| Завдання до лабораторної роботи №1                                                             | 13 |
| Методичні рекомендації до лабораторної роботи №1                                               | 13 |
| Контрольні питання до лабораторної роботи №1                                                   | 17 |
| Лабораторна робота №2 Тема: Створення анімацій. Анімації зі звуком.                            |    |
| Малювання                                                                                      | 18 |
| Теоретичні відомості                                                                           | 18 |
| Завдання до лабораторної роботи №2                                                             | 21 |
| Методичні рекомендації до лабораторної роботи №2                                               | 23 |
| Контрольні питання до лабораторної роботи №2                                                   | 25 |
| Завдання для самостійного виконання                                                            | 25 |
| Лабораторна робота №3 Тема: Створення анімацій з використанням циклів, розгалужень та датчиків | 26 |
| Теоретичні відомості                                                                           | 26 |
| Завдання до лабораторної роботи №3                                                             | 28 |
| Методичні рекомендації до лабораторної роботи №3                                               | 30 |
| Контрольні питання до лабораторної роботи №3                                                   | 33 |
| Лабораторна робота №4 Тема: Створення гри в середовищі Scratch. Робота зі списками             | 34 |
| Теоретичні відомості                                                                           | 34 |
| Завдання до лабораторної роботи №4                                                             | 36 |
| Методичні рекомендації до лабораторної роботи №4                                               | 38 |
| Контрольні питання до лабораторної роботи №4                                                   | 41 |
| Лабораторна робота №5 Тема: Математичні обчислення. Оператори порівняння                       | 41 |
| Теоретичні відомості                                                                           | 41 |
| Завдання до лабораторної роботи №5                                                             | 45 |
| Методичні рекомендації до лабораторної роботи №5                                               | 46 |
| Контрольні питання до лабораторної роботи №5                                                   | 48 |
| Лабораторна робота №6 Тема: Цикли. Масиви                                                      | 50 |
| Теоретичні відомості                                                                           | 50 |
| Завдання до лабораторної роботи №6                                                             | 52 |
| Методичні рекомендації до лабораторної роботи №6                                               | 54 |
| Контрольні питання до лабораторної роботи №6                                                   | 58 |
| Лабораторна робота №7 Тема: Робота з текстом. Створення підпрограм                             | 59 |
| Теоретичні відомості                                                                           | 59 |
| Завдання до лабораторної роботи №7                                                             | 62 |
| Методичні рекомендації до лабораторної роботи №7                                               | 63 |
| Контрольні питання до лабораторної роботи №7                                                   | 68 |

|                                                   |    |
|---------------------------------------------------|----|
| Лабораторна робота №8 Тема: Робота з «Черепашкою» | 69 |
| Теоретичні відомості                              | 69 |
| Методичні рекомендації до лабораторної роботи №8  | 71 |
| Завдання до лабораторної роботи №8                | 73 |
| Контрольні питання до лабораторної роботи №3      | 77 |
| Список використаних джерел                        | 78 |
| Рекомендована література                          | 78 |
| Список ілюстрацій                                 | 80 |

---

## ВСТУП

В межах даного курсу вивчається програмування в середовищах Scratch і Google Blockly. Це особливий вид програмування, яке називається *візуальним програмуванням*. Воно не вимагає ручного написання кодів, а передбачає зручне, зрозуміле учням складання кодів програм за допомогою перетягування блоків – окремих структурних одиниць програми. Таке програмування схоже на гру, тому підвищує зацікавленість учнів та мотивує їх до навчання.

Обидва середовища відносять до освітніх. Саме задля навчання програмування вони були створені. В середовищі Scratch школярі молодшого віку складають із блоків програми, які дозволяють керувати яскравими персонажами, їхніми рухами, діями, звуками, малюнками, створювати ігри, цікаві проєкти. Школярі більш старшого віку застосовують середовище Google Blockly, де можуть побачити «переклад» власної програми на такі мови програмування, як JavaScript, Python, Dart або XML, щоб у старшому віці програмувати на них.

Майбутні випускники спеціальності Середня освіта (Інформатика) мають досконало володіти візуальним програмуванням не лише для того, щоб вчити програмувати учнів закладів середньої освіти, а й щоб самим створювати власні освітні проєкти навчально-методичного характеру для використання в освітній діяльності.

**Метою** вивчення навчальної дисципліни «Візуальне програмування» є засвоєння студентами основ створення програм для ЕОМ шляхом маніпулювання графічними об'єктами замість написання тексту програми та ознайомлення з середовищами візуального програмування Scratch та Google Blockly.

Основними **завданнями** вивчення дисципліни «Візуальне програмування» є:

- засвоєння системних знань з візуального програмування;
- засвоєння основних методів створення програм за допомогою візуальних мов програмування;
- набуття навичок роботи в середовищі Scratch;
- оволодіння основами програмування у Google Blockly.

У результаті вивчення навчальної дисципліни студент повинен

**знати:**

- особливості візуальних мов програмування;
- візуальну мову програмування Scratch;
- візуальну мову програмування Google Blockly;
- основні області застосування візуальних мов програмування;

**уміти:**

- застосовувати візуальну мову програмування Scratch для створення анімаційних проєктів;
- застосовувати візуальну мову Google Blockly для написання стандартних програм;

- використовувати можливості та особливості візуальних мов програмування при створенні програмних продуктів.

**У результаті вивчення навчальної дисципліни студент повинен набути таких результатів навчання (знання, уміння тощо) та компетентностей:**

**результати навчання:**

- розуміти місце інформатики в системі наук, перспективи розвитку інформатики та інформаційних технологій, їхнє суспільне значення та можливості практичного застосування в різних сферах;
- застосовувати знання вищої математики та інформатики для розв'язання складних спеціалізованих задач професійної діяльності;
- розробляти, досліджувати і реалізовувати алгоритми розв'язування навчальних задач з інформатики, оцінювати ефективність розроблених алгоритмів, визначає межі їх застосовності;
- створювати інформаційні моделі, реалізовувати та досліджувати їх засобами інформаційно-комунікаційних технологій, інтерпретувати, аналізувати та узагальнювати отримані результати;

**компетентності:**

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність застосовувати інформаційні та комунікаційні технології;
- здатність вчитися і оволодівати сучасними знаннями;
- здатність до адаптації та дії в новій ситуації;
- здатність використовувати знання наукових фактів, концепцій, теорій, принципів і методів інформатики у практиці навчання інформатики в базовій середній школі та в роботі гуртка з інформатики;
- здатність реалізовувати інформаційну модель засобами інформаційно-комунікаційних технологій; здійснювати комп'ютерний експеримент;
- здатність до математичного, логічного і алгоритмічного мислення, обґрунтування вибору методів розв'язання задач, інтерпретації отриманих результатів.

**Міждисциплінарні зв'язки.** При вивченні курсу «Візуальне програмування» від здобувачів очікуються базові знання з дисципліни «Основи програмування».

У свою чергу, уміння й навички, засвоєні під час вивчення дисципліни «Візуальне програмування», знадобляться при вивченні дисципліни «Практикум з програмування», а також при проходженні виробничої практики (педагогічної) та виконанні курсової роботи, передбаченою освітньо-професійною програмою «Середня освіта (Інформатика)».

---

## ЗАГАЛЬНІ РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

Лабораторні заняття з дисципліни «Візуальне програмування» призначені для того, щоб студент під керівництвом викладача опрацьовував теоретичний матеріал, вчився створювати алгоритми та застосовував візуальні середовища Scratch та Google Blockly для написання стандартних програм, для розв'язання прикладних завдань, використовував можливості та особливості візуальних мов програмування при створенні програмних продуктів, набував практичних навичок роботи з комп'ютерною технікою.

Захист лабораторної роботи відбувається таким чином: студент демонструє етапи реалізації завдання та отримані результати, пояснює розв'язання обраних викладачем завдань, відповідає на запитання. Захист роботи – це презентація своїх результатів за допомогою комп'ютера, а також практичне виконання деяких елементів роботи у режимі реального часу. Після цього студент надсилає звіт з виконання лабораторної роботи до системи Moodle.

При виконанні кожної роботи необхідно ознайомитись із загальними теоретичними відомостями та матеріалами відповідної лекції (деякі моменти необхідно опрацьовувати самостійно); виконати практичні завдання; оформити звіт з виконання лабораторної роботи, який повинен містити:

- титульний аркуш;
- тему роботи;
- хід виконання відповідного завдання;
- файл з виконаним завданням.

Готовий звіт завантажується до СЕЗН Moodle у відведені викладачем терміни. Якщо розмір файлу перевищує 5Мб, то відповіді розміщуються на зовнішніх ресурсах, а до СЕЗН додається посилання на ці ресурси.

## Лабораторна робота №1

## Тема: Основи роботи в середовищі Scratch

**Мета:** Ознайомитись з основними можливостями середовища Scratch, блоками, їх призначенням. Навчитися розробляти найпростішу анімацію.

## Теоретичні відомості

Зазвичай програми пишуться, тобто друкуються з клавіатури і складаються з набору команд, наприклад: `int i = 0; b = sin(a) + ln(x); do ... while(), if(...) ... else ...` і т.д. Тобто код програми подається у текстовому вигляді. Такий підхід не завжди зручний. Наприклад, маленьким дітям складно пояснити значення доволі абстрактних команд, а з телефонів чи планшетів написання навіть найкоротшого коду займатиме багато часу і буде незручним.

Візуальну мову програмування Scratch учні вивчають на різних рівнях освіти – від початкової школи до вищої. Програмування блоками в Scratch інтуїтивно зрозуміле навіть дитині. Програмісту не треба писати код, йому треба лише розташувати на полі блоки у правильній взаємодії один з іншим. Простий візуальний інтерфейс дозволяє створювати анімацію, ігри, тести, навчальні проєкти тощо. Scratch розвиває алгоритмічне та абстрактне мислення учнів, визначає творчий підхід до навчання.

Розробку та модерування Scratch здійснює некомерційна організація Foundation Scratch<sup>1</sup>. Середовище розробки повністю перекладене майже 70 різними мовами, а його використання є безкоштовним.

Щоб *розпочати роботу* у візуальному середовищі розробки Scratch, можна його установити на персональному комп'ютері, скориставшись файлом завантаження з сайту <https://scratch.mit.edu/download>, або використати *онлайн-середовище*, перейшовши за посиланням: <https://scratch.mit.edu/>. В рамках даного видання зосередимо увагу на огляді функціональних можливостей саме онлайн-середовища Scratch.

Для *відкриття вікна написання програми* потрібно клацнути на кнопку «Розпочати творити» (рис. 1.1).

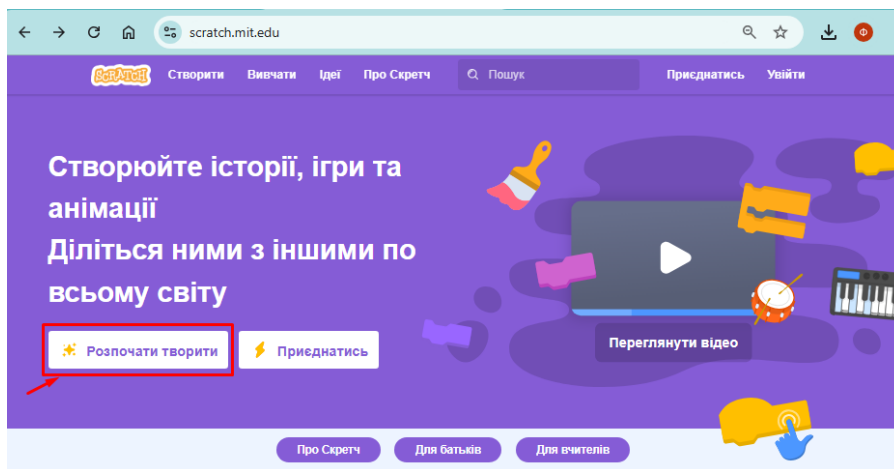


Рис. 1.1 – Стартова сторінка візуального середовища Scratch

<sup>1</sup> Фундація Скретчу. URL: <https://www.scratchfoundation.org/>

Інтерфейс середовища програмування Scratch (рис. 1.2) має три основні секції: палітра блоків, область сцени, область кодування.

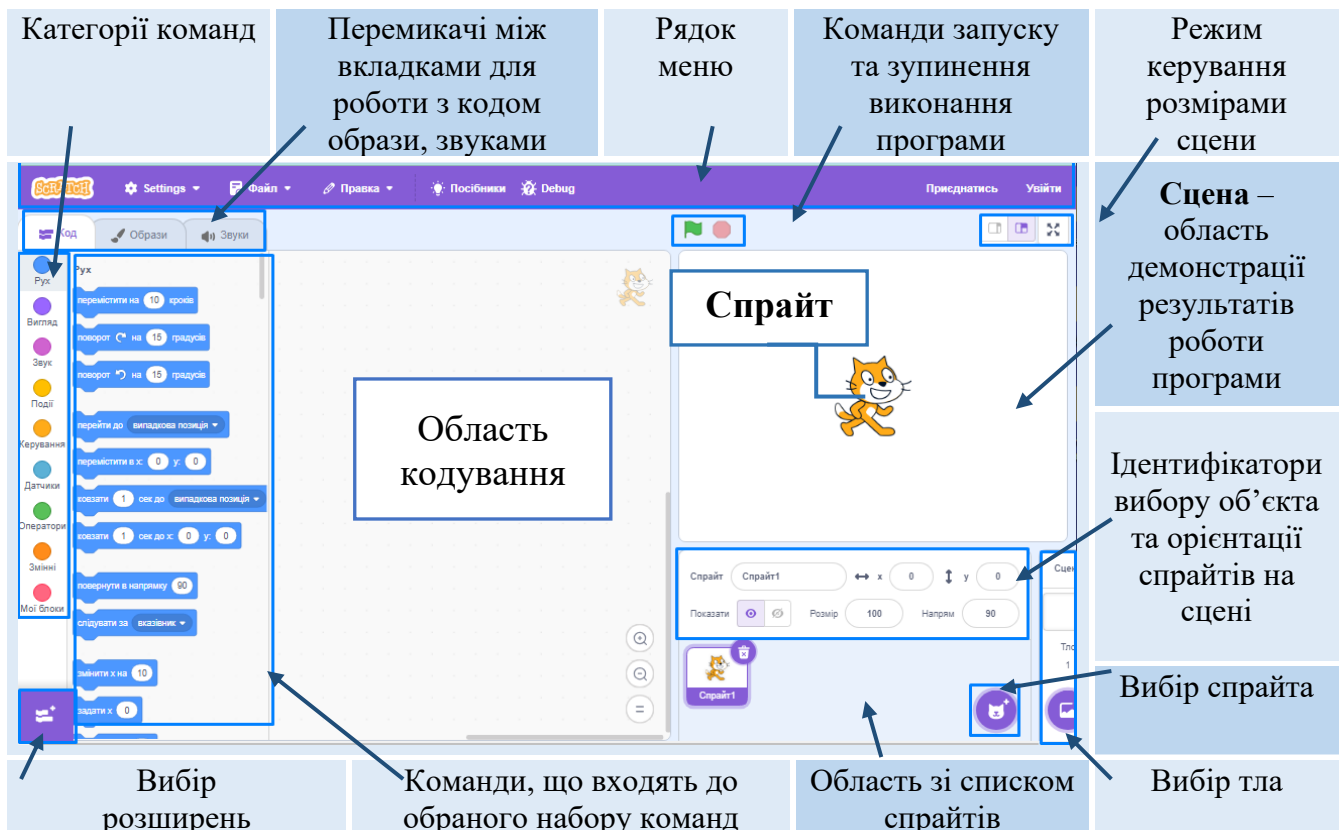


Рис. 1.2 – Інтерфейс вікна середовища візуального програмування Scratch

Всі *команди* за призначенням розбиті на 9 основних категорій: «Рух», «Вигляд», «Звук», «Події», «Керування», «Датчики», «Оператори», «Змінні», «Мої блоки». За цим же принципом блоки розфарбовані у різні кольори, щоб полегшити пошук потрібного елементу. Також існують додаткові категорії (див. рис. 1.3), які можна додати за допомогою кнопки вибору *розширень* (рис. 1.2).

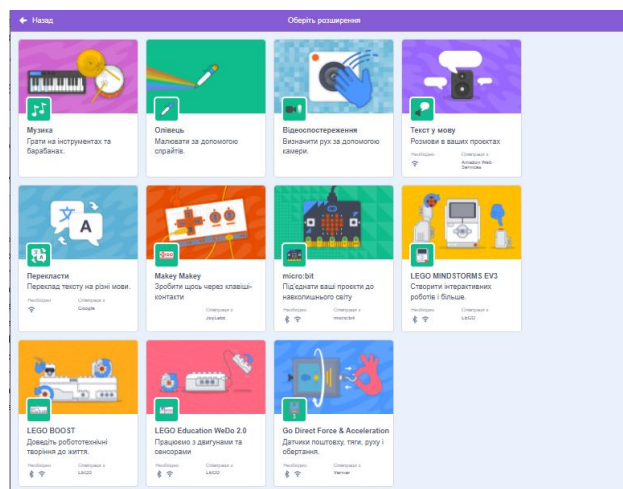


Рис. 1.3 – Вікно обрання розширень

Переглянемо *основні категорії*. Перша основна категорія – «Рух» (рис. 1.4). Команди цієї категорії відповідають за переміщення, зміну напрямків та позицій об'єктів на сцені (повороти, переміщення вздовж осей, крокування, слідування за вказівником та ін.).

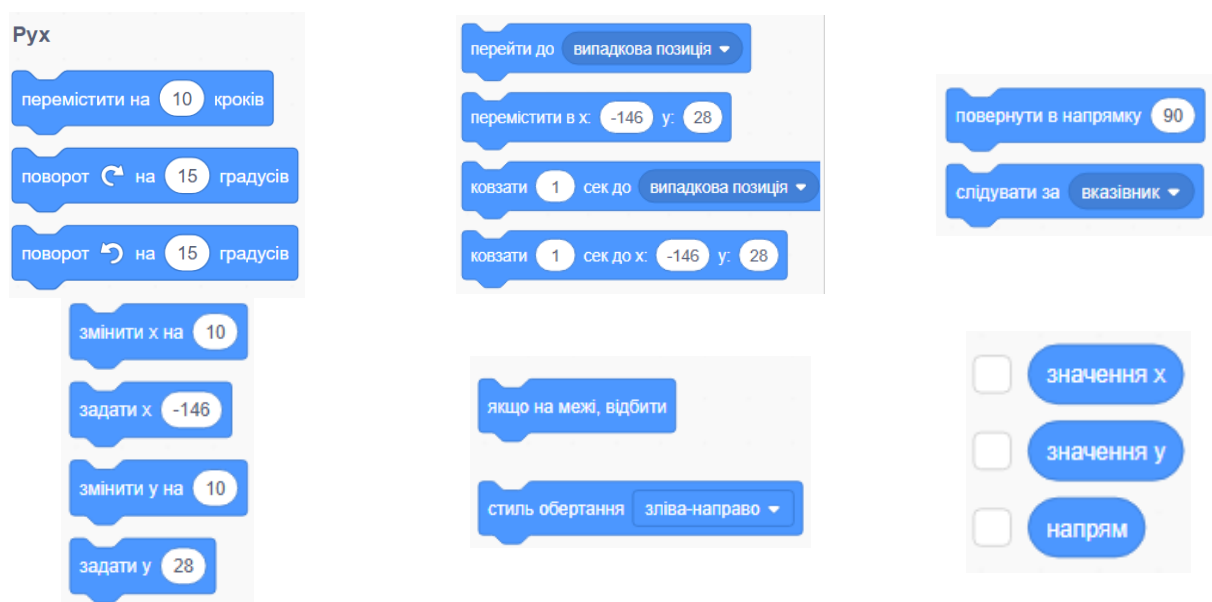


Рис. 1.4 – Команди категорії «Рух»

Друга категорія «Вигляд» (рис. 1.5) відповідає за управління текстовими повідомленнями, зміною образів, розмірів, кольору об'єктів, появою та приховуванням об'єкта, взаємним положенням об'єктів.

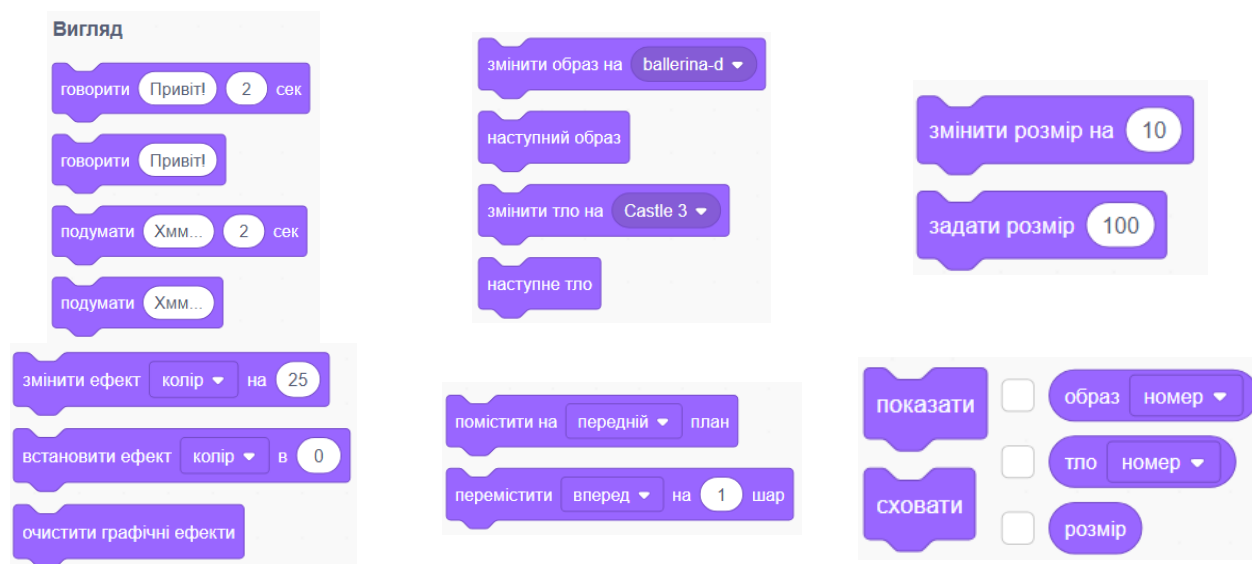


Рис. 1.5 – Команди категорії «Вигляд»

Команди третьої категорії «Звук» дозволяють відтворювати звук та звукові файли, керувати ефектами звуку (висотою, гучністю звуку) та ін. (рис. 1.6).

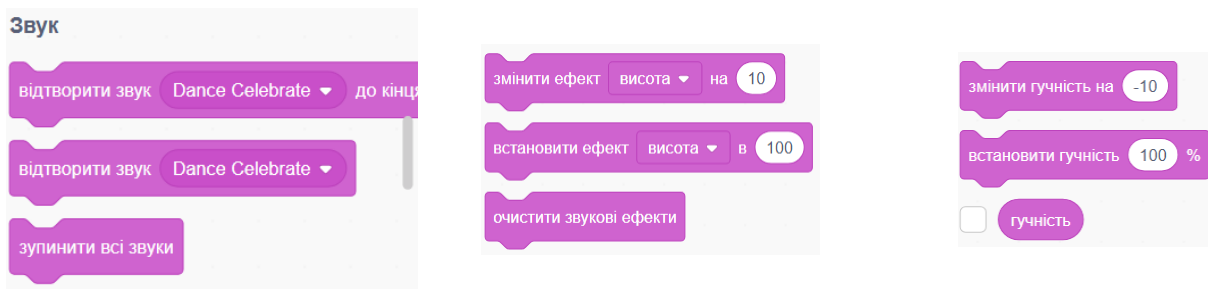


Рис. 1.6 – Команди категорії «Звук»

Команди четвертої категорії «Події» керують подіями: управляють запуском програми, включають клавіші управління спрайтами, оповіщення та ін. (рис. 1.7),

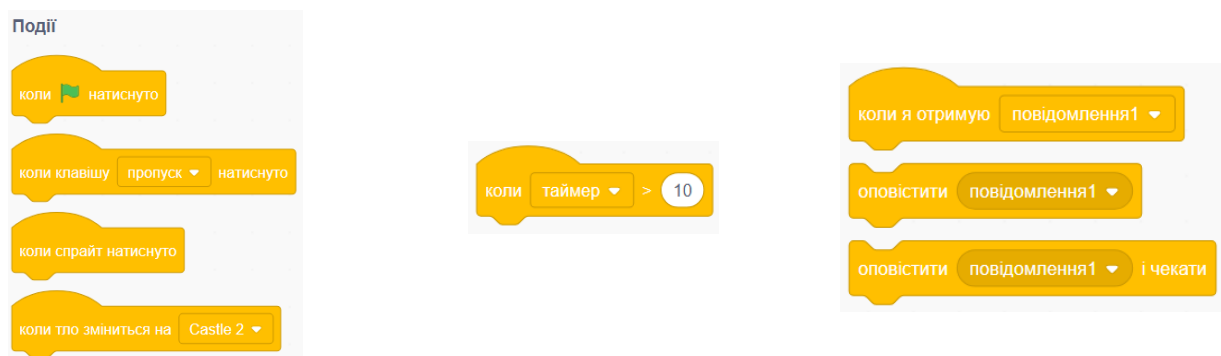


Рис. 1.7 – Команди категорії «Події»

Команди п'ятої категорії «Керування» включають блоки циклів з відомою та невідомою кількістю повторень, блоки розгалуженої структури, також запуск, створення та вилучення клонів та ін. (рис. 1.8).

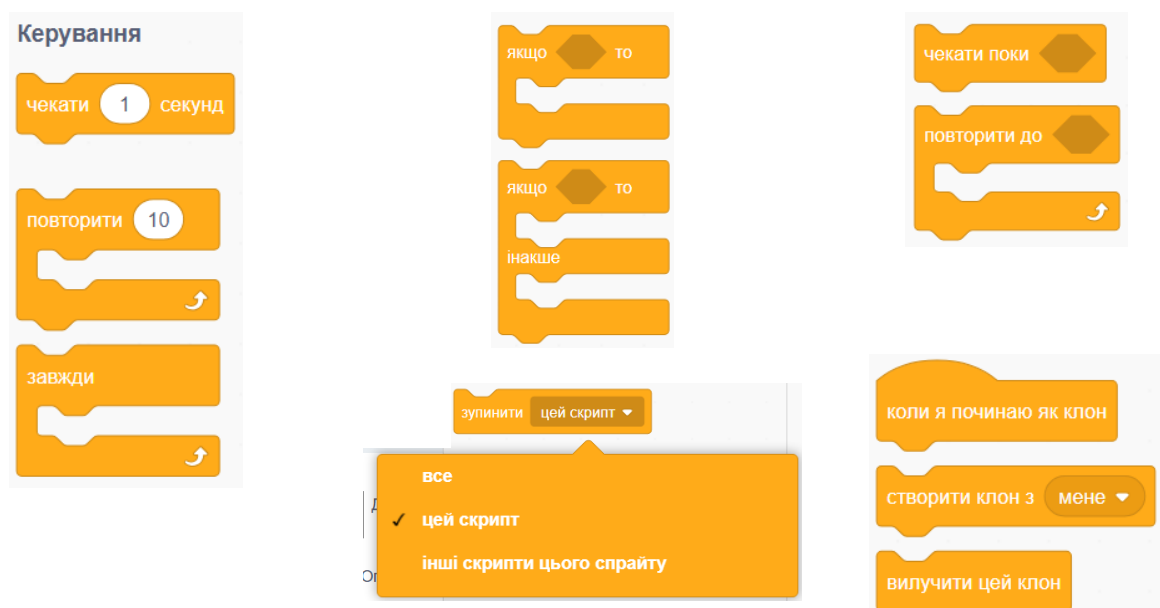


Рис. 1.8 – Команди категорії «Керування»

Команди категорії «Датчики» визначають взаємодію спрайтів, дії при торканні до кольору, клацанні по клавішам, керувати «діалогами» між спрайтами і користувачем програми та ін. (рис. 1.9).

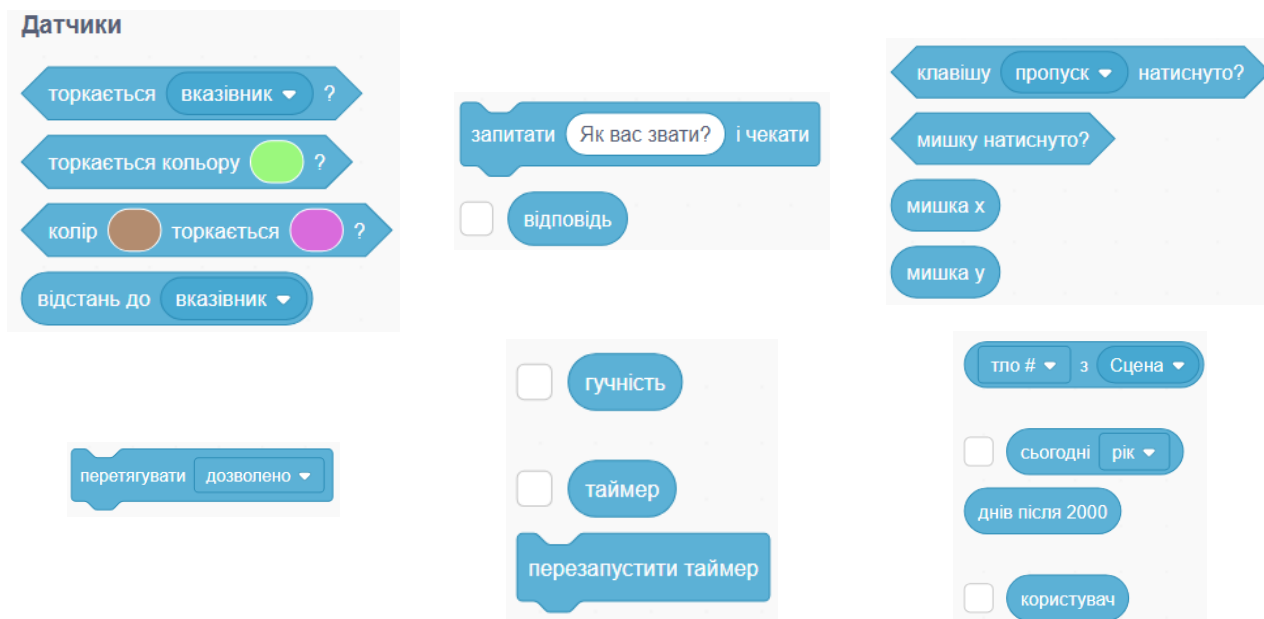


Рис. 1.9 – Команди категорії «Датчики»

Команди категорії «Оператори» дозволяють виконувати математичні дії з числовими константами та змінними, операції над текстом, задавати логічні відношення між числами (рис. 1.10).



Рис. 1.10 – Команди категорії «Оператори»

Команди категорії «Змінні» дозволяють використовувати змінні та списки (рис. 1.11).

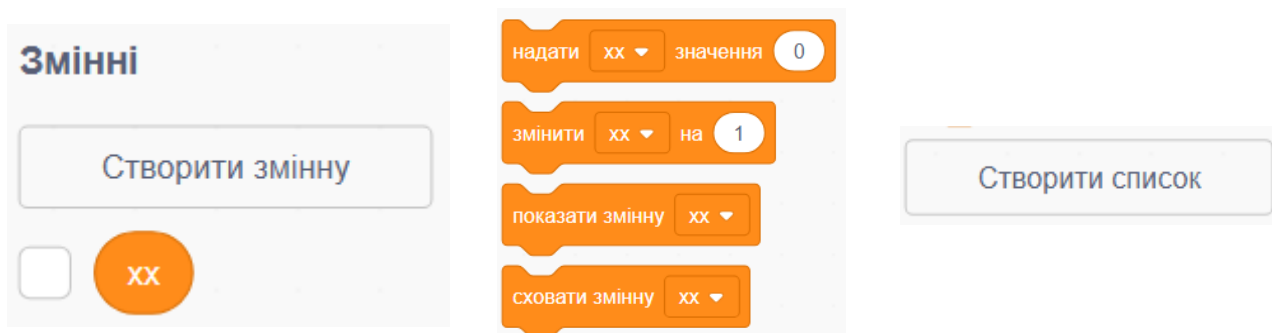


Рис. 1.11 – Команди категорії «Змінні»

Scratch також дозволяє створювати блоки для подальшої роботи з процедурами (рис. 1.12).

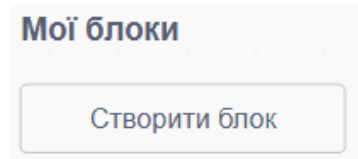


Рис. 1.12 – Команди категорії «Змінні»

Для запуску програми на виконання та зупинки її роботи використовують функціональні кнопки  і  відповідно (див рис. 1.2 – Інтерфейс вікна середовища візуального програмування Scratch).

Для створення нової програми, завантаження існуючої, файл з якою розміщено на комп'ютері, або ж збереження старого потрібно у стрічці меню обрати «Файл», а у меню, що випадає, обрати відповідну позицію, як показано на рис. 1.13.

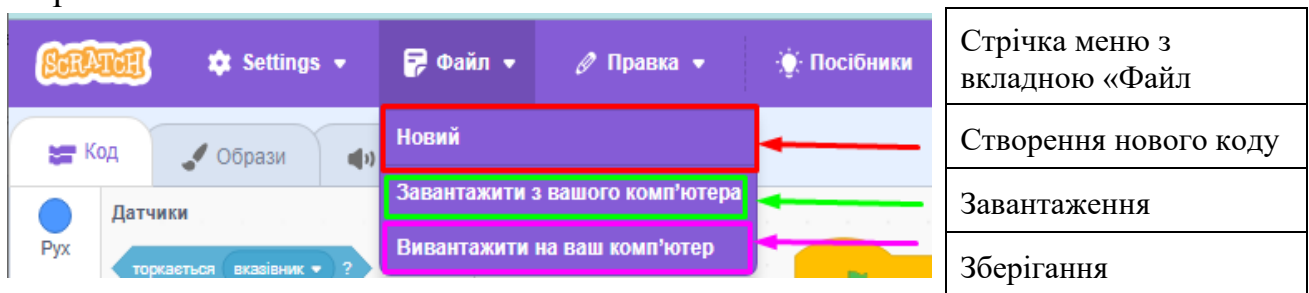


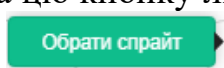
Рис. 1.13 – Робота з файлами програм

### Завдання до лабораторної роботи №1

1. Ознайомитися з інтерфейсом середовища Scratch, його основними об'єктами та командами.
2. Створити власний проект (сюжет обрати за бажанням), але в проєкті мають бути обов'язкові елементи:
  - один спрайт (без анімації);
  - тло;
  - друковане привітання або інший текст;
  - переміщення спрайта (спрайт може переміщуватися по сцені довільним чином, може відбиватися від межі, повертатися, стрибати, літати, плавати, змінювати рух);
  - рух спрайта по сцені керується різними способами (натисканням на стрілки, пробіл і ін).

### Методичні рекомендації до лабораторної роботи №1

Додавання об'єкта на сцену можна реалізувати через додавання з бібліотеки, намалювати вручну або завантажити власний об'єкт.

✓ Додавання спрайта на сцену через вибір з бібліотеки реалізується через функціональну кнопку . При натисканні на цю кнопку лівою кнопкою миші або при виборі в меню, що випадає, пошуку   відкривається вікно вибору спрайтів (рис. 1.14). Переходьте за категоріями і обирайте спрайт. Після клацання по обраному спрайту він з'являється на сцені.

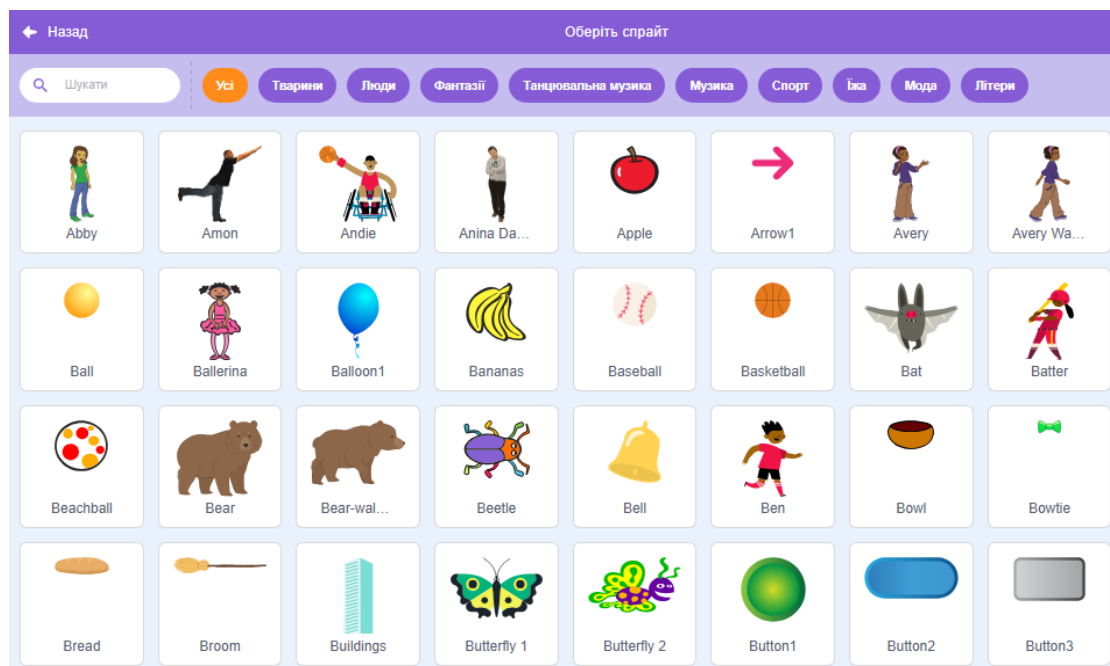
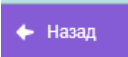


Рис. 1.14 – Бібліотека спрайтів

Для виходу з вікна вибору спрайтів без створення нового спрайту клацніть на стрілку  у верхньому лівому куті вікна, та перейдьте до вікна створення програми.

Спрайти також можна *намалювати* власноруч або завантажити власне зображення ( ознайомитися з такими функціями самостійно!).

Для вилучення непотрібного слайду зі сцени потрібно клацнути на кошику, розміщений поруч з зображенням спрайту в області зі списком спрайтів (рис. 1.15).

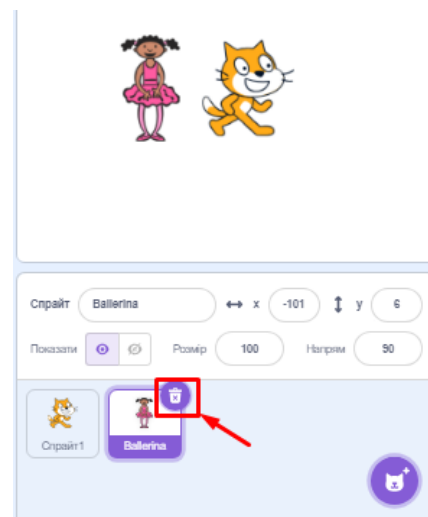


Рис. 1.15 – Видалення спрайта зі сцени

✓ Додавання тла з бібліотеки на сцену реалізується за допомогою кнопки  за тією ж схемою, що й додавання спрайта. Вікно вибору тла наведено на рис. 1.16.

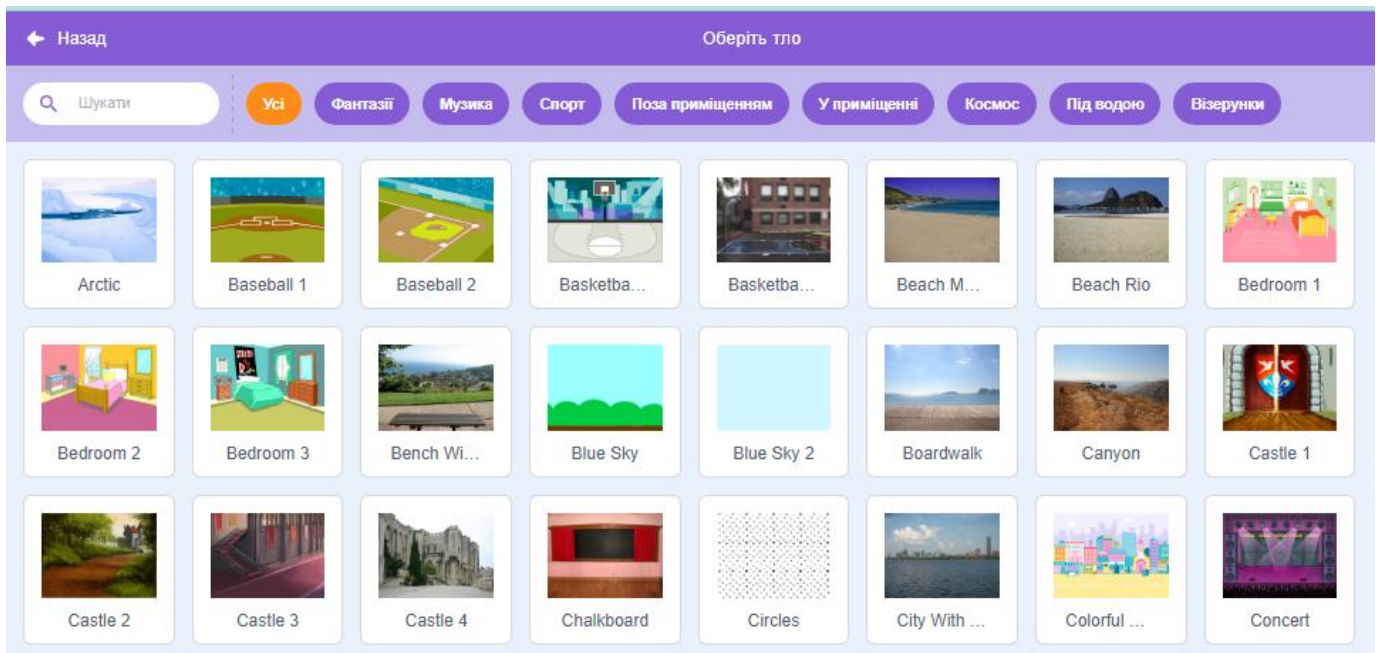


Рис. 1.16 – Бібліотека «Оберіть тло»

✓ Для додавання привітання або іншого тексту потрібно використати команди з категорій «Події» та «Вигляд». Код може мати, наприклад, такий вигляд, як на рис. 1.17.

**Завдання 1.1.** Складіть код з рис. 1.17.

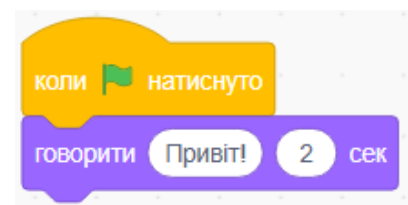


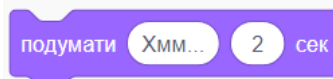
Рис. 1.17 – Код програми з додаванням привітання

✓ Розглянемо *переміщення об'єкта*. За це відповідає категорія «Рух» (рис. 1.4).

Ознайомлення з командами категорії «Рух» демонструє необхідність розуміння координат та куту повороту об'єкта на сцені. На рис. 1.18 зображено відповідну координатну сітку на сцені. Відповідні числові значення орієнтації спрайта на сцені розміщуються в області ідентифікаторів (див. рис. 1.2, рис. 1.18).

**Приклад 1.1.** Написати програму переміщення спрайта по прямокутнику на сцені.

► Зразок найпростішої програми реалізації руху спрайта наведено на рис. 1.19. Зверніть увагу, що програма використовує команду



категорії «Вигляд» для кращої наочності роботи програми. ◀

**Завдання 1.2.** Внесіть код прикладу до середовища Scratch для ознайомлення з принципами роботи програми та її результатами.



Рис. 1.18 – Орієнтація спрайта на сцені на координатній сітці



Рис. 1.19 – Код програми переміщення спрайта

Параметри орієнтації положення спрайта також можна «підсвітити», поставивши прапорець на командах «значення x», «значення y» або «напрямок» категорії «Рух» (рис. 1.20, а), параметри відобразяться у верхньому лівому куті сцени (рис. 1.20, б).

При написанні ігрових програм спрайт може опинитися за межами сцени, тоді використовують команду

якщо на межі, відбити

Але можуть виникнути несподівані перевертання спрайтів. Щоб такого не відбувалося, застосовують команду

стиль обертання

зліва-направо

(1.1)

**Завдання 1.3.** Складіть програму відповідно до рис. 1.21, як з використанням команди (1.1), так і без неї.

Зауважимо, що в програми на рис. 1.21 застосовано циклічну структуру із категорії «Керувати».



а

б

Рис. 1.20 – Параметри орієнтації спрайта на сцені

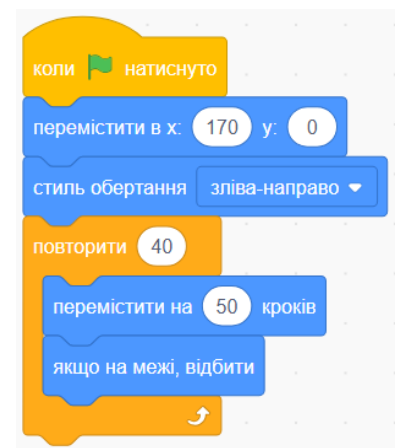


Рис. 1.21 – Код програми, що передбачає відбиття спрайта від межі сцени

✓ Розглянемо керування рухом спрайта. Для цього можна використати цикл з повтором «завжди»; в блок розгалуження «якщо» вставити як логічний вираз датчик, що визначає дії при натисканні клавіш, щоб забезпечити напрям руху при такому натисканні.

**Приклад 1.2.** Написати програму керування рухом спрайта за допомогою клавіатури.

► Зразок реалізації програми руху спрайта при натисканні стрілок на клавіатурі наведено на рис. 1.22. ◀

**Завдання 1.4.** Проаналізуйте програму або введіть її код в середовищі Scratch та зробіть висновок, що додатково виконує наведена програма.

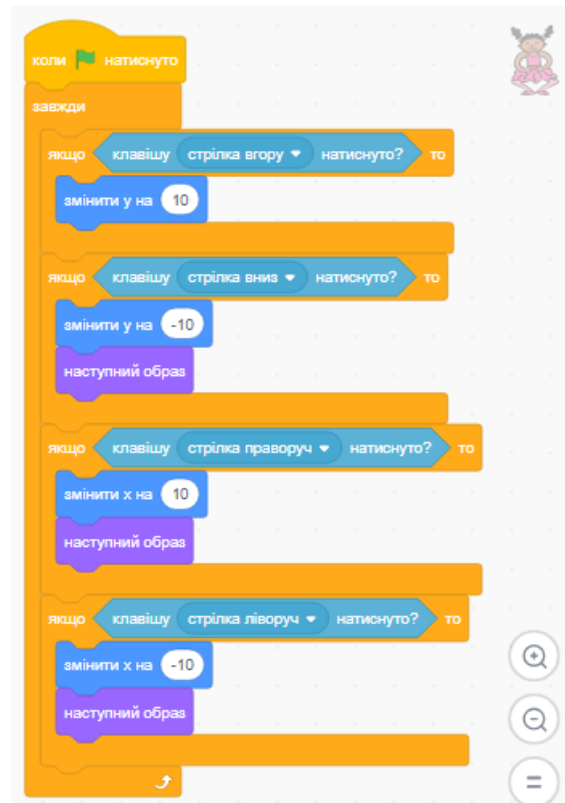


Рис. 1.22 – Програма керування рухами спрайта

### Контрольні питання до лабораторної роботи №1

1. Які особливості має мова програмування Scratch?
2. Які способи додавання об'єктів на сцену?
3. Які категорії команд має середовище Scratch?
4. Які функції виконують команди кожного блоку?
5. В який спосіб можна запобігти перевертання спрайта при відбитті від межі сцени?
6. В якій категорії міститься команда, яка дозволяє реалізувати функцію управління спрайтом за допомогою клавіш клавіатури?

## Лабораторна робота №2

### Тема: Створення анімацій. Анімації зі звуком. Малювання

**Мета:** Ознайомитися з засобами створення анімованих проєктів зі звуковим супроводженням. Навчитися створювати анімації зі звуком.

### Теоретичні відомості

Анімація створюється комбінацією команд всіх категорій мови програмування Scratch. Один з проявів анімації було розглянуто в лабораторній роботі №1.

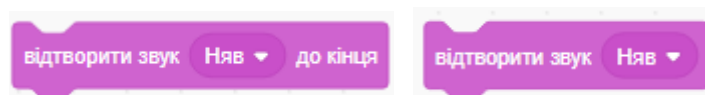
Проте, попередня лабораторна робота не передбачала застосування звуків. Створення звукових ефектів в середовищі Scratch можна реалізовувати за допомогою команд категорії «Звук», вкладки «Звуки» панелі інструментів, розширення «Музика» та «Текст у мову».

У спрайтів є своє звукове супроводження, яке можна доповнювати. Проте, є особливі спрайти, яких розміщено в групі «Танцювальна музика» та «Музика», наділені більшим спектром вибору звукових сценаріїв.

Отже, розпочнемо по порядку.

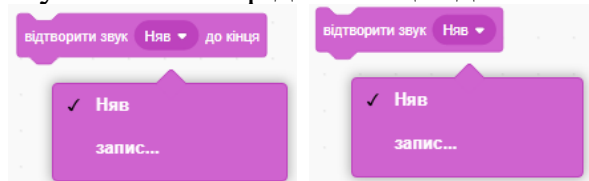
✓ Розглянемо *категорію «Звук»* (див. рис. 1.6) (рис. 1.6 – Команди категорії «Звук») на прикладі спрайта котика.

Команди



дозволяють відтворювати нявкання котика з відмінним функціоналом. Якщо після першої команди поставити іншу, то доки не завершиться нявкання, додана команда не буде виконуватися. Якщо ж аналогічну операцію зробити з другою командою, то додана команда буде виконуватися одночасно з нявканням.

Ці ж команди допускають вибір двох опцій для котика



А саме: або відтворення нявкання, або ж запис аудіо за допомогою мікрофона, щоб використовувати його у подальшому. При виборі «запис...» відкривається вікно запису (рис. 2.1, а), а після клацання на кнопку «Записати» здійснюється аудіозапис до моменту, коли його зупиняють. Потім з'являється вікно (рис. 2.1, б), яке дозволяє обрізати зайві частини аудіо, відтворити, перезаписати або зберегти запис. Після збереження реалізується перехід до інструменту «Звуки», функціонал якого можна побачити на рис. 2.2.

Якщо Ви хочете додати інше аудіо для спрайта, то скористайтеся кнопкою «Обрати звук» в нижньому лівому куті інструментального вікна «Звуки». Функціонал кнопки див. на рис. 2.3. Якщо реалізувати вибір функціоналу за вкладкою  «Обрати звук», то у вікні, що відкриється, можна обрати звук з бібліотеки звуків (рис. 2.4).

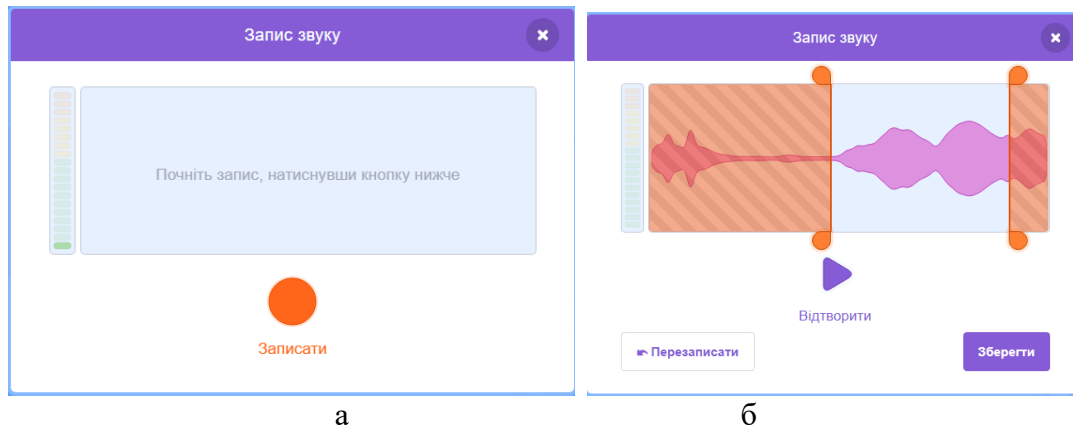


Рис. 2.1 – Запис аудіо для подальшого відтворення спрайтом

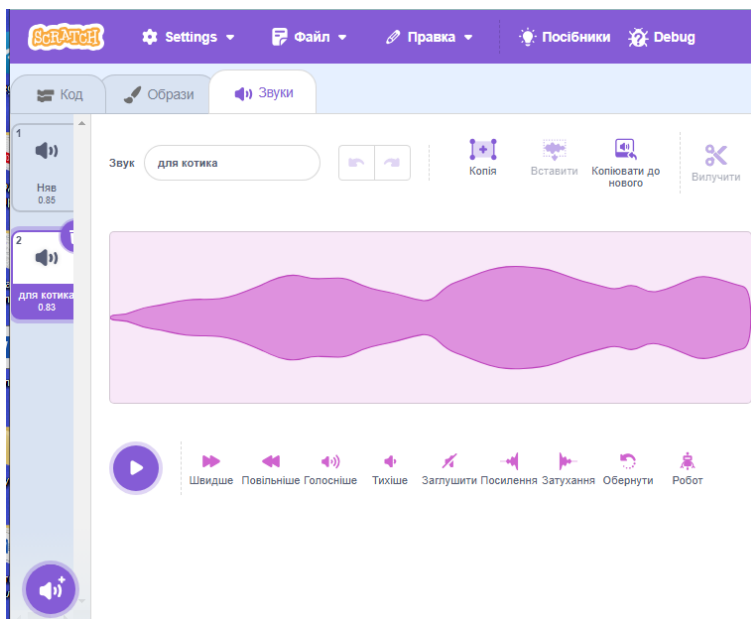


Рис. 2.2 – Вікно інструменту «Звуки»

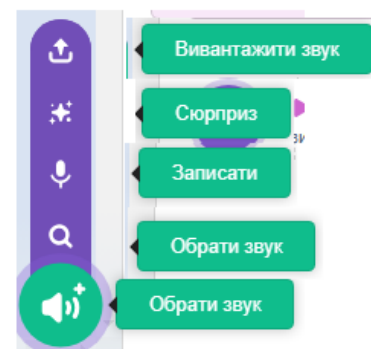


Рис. 2.3 – Функціонал кнопки «Обрати звук»

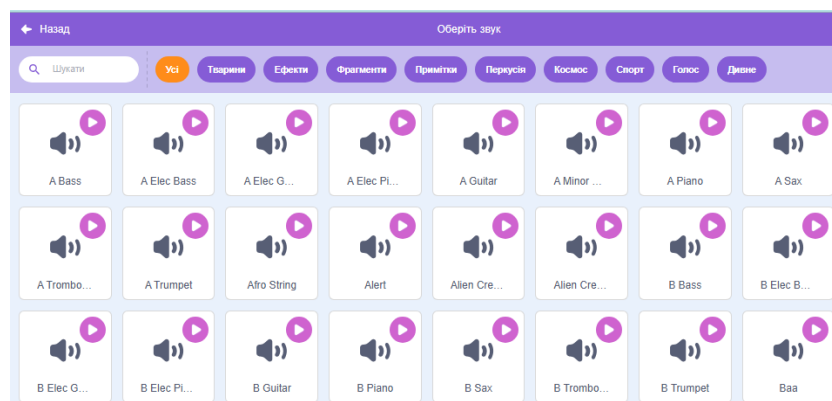


Рис. 2.4 – Бібліотека звуків

Всі звуки, які Ви преднали до спрайта, додадуться до вибору команд «відтворити звук...».

Інші ефекти керування звуком (див. рис. 1.6Рис. 1.6 – Команди категорії «Звук») в однойменній категорії зрозумілі за їхньою назвою. Уважно перегляньте команди категорії та поекспериментуйте з ними в середовищі Scratch.

✓ Розглянемо розширення «Музика» та «Текст у мову», які додаються до категорій за допомогою кнопки «Додати розширення», розміщеної у нижньому лівому куті вікна Scratch (див. рис. 1.2 – Інтерфейс вікна середовища візуального програмування Scratch). Команди категорій див. на (рис. 2.5).

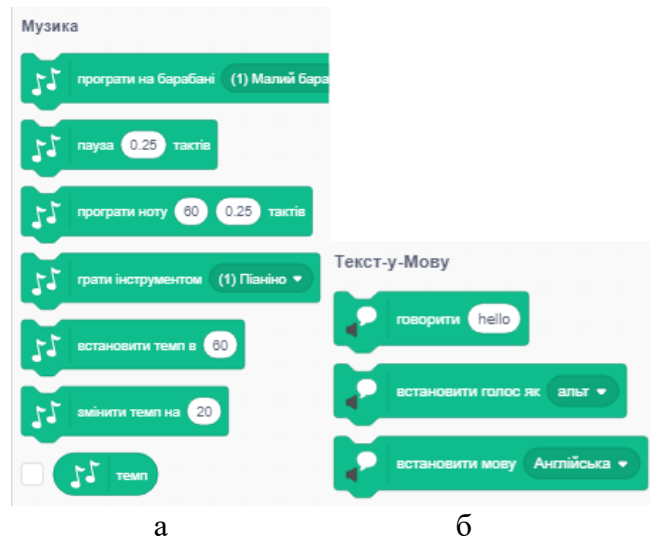


Рис. 2.5 – Команди категорій «Музика» (а) і «Текст у мову» (б)

Команда «програти ноту...» дозволяє обрати конкретну ноту, якій присвоєно певний номер, установити тривалість її звучання, а в поєднанні з командою «грати інструментом...» реалізувати звучання саме на ньому (рис. 2.6). У поєднанні з іншими командами категорії можна задати темп, паузи, залучити інші інструменти і т.ін., що дозволить створити власну музичну композицію

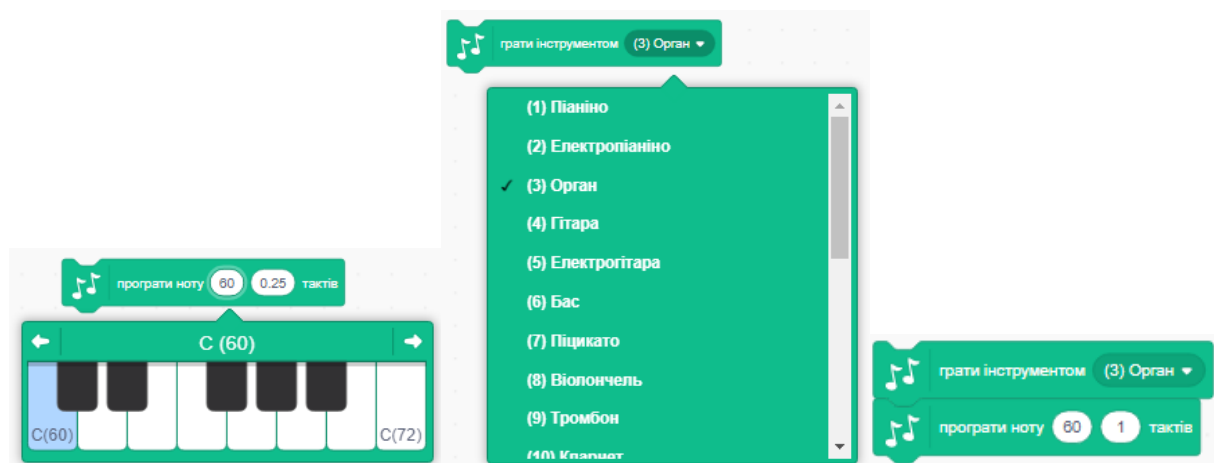


Рис. 2.6 – Використання команд категорії «Музика»

Команда «програти на барабані...» дозволяє залучати ударні інструменти і керувати тривалістю проміжку часу перед залученням наступного інструменту. Можна також встановлювати паузи командою «пауза...».

Якщо Ви хочете, щоб спрайт заговорив, використайте команди категорії «Текст у мову». Проекспериментуйте і дізнайтеся про переваги та недоліки цієї категорії!

## **Завдання до лабораторної роботи №2**

Ознайомитися з командами категорій «Звук», «Музика», «Текст у мову». Навчитися створювати анімації зі звуковим супроводженням. Виконати завдання свого варіанту (номер варіанту збігається з номером в списку групи).

### **Варіант 1**

1. Створити танцювальну анімацію зі спрайтом Anina Dance. Обрати тло, підібрати музику та танцювальні рухи таким чином, щоб спрайт насправді танцював, а не просто хаотично змінював пози. Анімація має продовжуватися не менше 30 секунд.
2. Створити музикальну композицію (звучання не менше 20с), в якій візьмуть участь спрайти Drum Kit та Guitar. За бажанням можна додати інші музикальні інструменти.
3. Створити анімацію за умовою: спрайт Hedgehog бігає по лісовій галявині в різні боки. При натисканні на нього, згортається у клубочок. При натисканні на «пробіл» – засинає.

### **Варіант 2**

1. Створити танцювальну анімацію зі спрайтом Cassy Dance. Обрати тло, підібрати музику та танцювальні рухи таким чином, щоб спрайт насправді танцював, а не просто хаотично змінював пози. Анімація має продовжуватися не менше 30 секунд.
2. Створити музикальну композицію (звучання не менше 20с), в якій візьмуть участь спрайти Drum та Guitar Electric 1. За бажанням можна додати інші музикальні інструменти.
3. Створити анімацію за умовою: спрайт Lama піднімається вгору по схилу і на вершині опускається на 4 лапи.

### **Варіант 3**

1. Створити танцювальну анімацію зі спрайтом LB Dance. Обрати тло, підібрати музику та танцювальні рухи таким чином, щоб спрайт насправді танцював, а не просто хаотично змінював пози. Анімація має продовжуватися не менше 30 секунд.
2. Створити музикальну композицію (звучання не менше 20 с), в якій візьмуть участь спрайти Drum-Cymbal та Keyboard. За бажанням можна додати інші музикальні інструменти.
3. Створити анімацію за умовою: спрайт Hare бігає по лісу в різні боки. При натисканні на «пробіл» – сідає.

#### **Варіант 4**

1. Створити танцювальну анімацію зі спрайтом Jovi Dance. Обрати тло, підібрати музику та танцювальні рухи таким чином, щоб спрайт насправді танцював, а не просто хаотично змінював пози. Анімація має продовжуватися не менше 30 секунд.
2. Створити музикальну композицію (звучання не менше 20 с), в якій візьмуть участь спрайти Drum-Highhat та Saxophone. За бажанням можна додати інші музикальні інструменти.
3. Створити анімацію за умовою: спрайт Dog 2 бігає по пляжу в різні боки. При натисканні на нього – замислюється.

#### **Варіант 5**

1. Створити танцювальну анімацію зі спрайтом Ten80 Dance. Обрати тло, підібрати музику та танцювальні рухи таким чином, щоб спрайт насправді танцював, а не просто хаотично змінював пози. Анімація має продовжуватися не менше 30 секунд.
2. Створити музикальну композицію (звучання не менше 20 с), в якій візьмуть участь спрайти Drum-snare та Trumpet. За бажанням можна додати інші музикальні інструменти.
3. Створити анімацію за умовою: спрайт Butterfly 2 при натисканні на нього перелітає з квітки на квітку (тло Flowers).

#### **Варіант 6**

1. Створити танцювальну анімацію зі спрайтом D-Money Dance. Обрати тло, підібрати музику та танцювальні рухи таким чином, щоб спрайт насправді танцював, а не просто хаотично змінював пози. Анімація має продовжуватися не менше 30 секунд.
2. Створити музикальну композицію (звучання не менше 20 с), в якій візьмуть участь спрайти Drums Conga та Guitar Electric 2. За бажанням можна додати інші музикальні інструменти.
3. Створити анімацію за умовою: спрайт Hen гуляє по фермі та дзьобає зернятка, коли Ви натискаєте на «пробіл».

#### **Варіант 7**

1. Створити танцювальну анімацію зі спрайтом Champ99. Обрати тло, підібрати музику та танцювальні рухи таким чином, щоб спрайт насправді танцював, а не просто хаотично змінював пози. Анімація має продовжуватися не менше 30 секунд.
2. Створити музикальну композицію (звучання не менше 20 с), в якій візьмуть участь спрайти Drums Tabla та Bell. За бажанням можна додати інші музикальні інструменти.
3. Створити анімацію за умовою: спрайт Owl літає довільним чином та при натисканні на «пробіл» сідає на камінь (тло Garden-rock).

## Методичні рекомендації до лабораторної роботи №2

✓ Розглянемо деякі можливі варіанти *синхронізації рухів спрайта та музичного супроводу*.

Узгодженість рухів котика і ударних інструментів демонструє код на рис. 2.9.

**Завдання 2.1.** Проаналізуйте, що є результатом роботи наданої на рис. 2.9 програми та складіть її.

Спрайти з каталогу бібліотеки «Танцювальна музика» мають більше образів

**Приклад 2.1.** Для спрайта Jovi Dance написати програму її танцю під музику.

► Зверніть увагу, що розмір персонажа має бути узгодженим з обраним фоном (рис. 2.8).

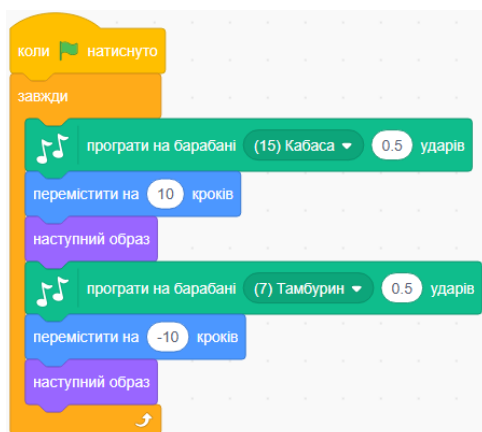


Рис. 2.7 – «Танок» спрайта під ударні інструменти

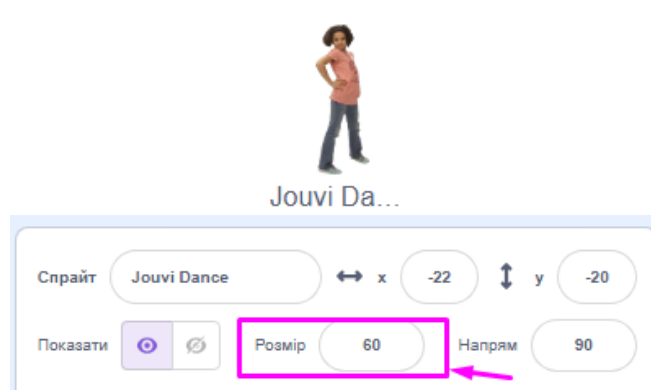


Рис. 2.8 – Опрацювання розмірів спрайта

Обраний спрайт має 13 образів і мелодію, яку Ви можете за бажанням змінити. Відтворення звукової мелодії краще зробити самостійним блоком з командою «відтворити звук... до кінця», щоб звучання мелодії не залежало від послідовності команд, які виконує спрайт.

Прослухайте мелодію, встановіть її ритмічність, щоб узгодити з тривалістю пауз між змінами образів.

Зміну образів спрайта потрібно робити циклічною:

- якщо для спрайта Ви застосовуєте цикл «завжди», то той же цикл установіть і для мелодії;
- якщо ж для персонажа обрано цикл з відомою кількістю повторів, то узгодьте їх кількість з часом звучання мелодії, як на (рис. 2.9). ◀

**Завдання 2.2.** Складіть програму

з рис. 2.9. Додайте переміщення спрайта по сцені з відбиттям від межі.

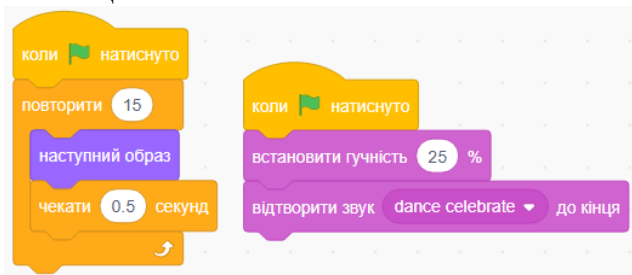


Рис. 2.9 – Код програми танцю спрайта Jovi Dance

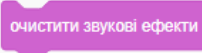
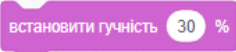
✓ Тепер розглянемо спрайти з каталогу *бібліотеки образів «Музика»*, до розміщено спрайти музичних інструментів, співачки та ін. Кожен зі спрайтів має свій набір звукових ефектів.

**Приклад 2.2.** Написати дві програми звучання інструментів. У першій інструментами керує користувач, а у другій – звучання написаної Вами мелодії здійснюється без зовнішнього втручання.

► Якщо Ви маєте бажання керувати інструментами в процесі роботи програми, то визначте кожному інструменту свою клавішу, наприклад, так, як показано на рис. 2.10.

Якщо Ви хочете отримати звучання запрограмованої Вами мелодії, то кожному інструменту можете призначити команду . У разі повторень однакової мелодії застосуйте циклічні структури з можливою зміною висоти, гучності звучання на кожному повторі (рис. 2.11).

Зверніть увагу, що середовище запам'ятовує стан звукових ефектів, який був використаним по завершенню роботи програми. Тому при повторному

запису на виконання потрібно скинути ефекти  і встановити початкову бажану гучність . ◀

**Завдання 2.3.** Для двох довільних інструментів напишіть програми з визначенням керування кожним інструментом та без нього.

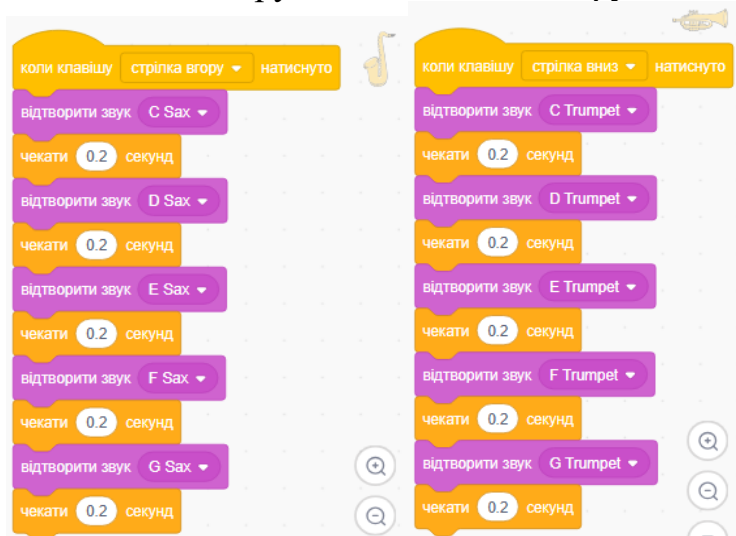


Рис. 2.10 – Керування музичними інструментами

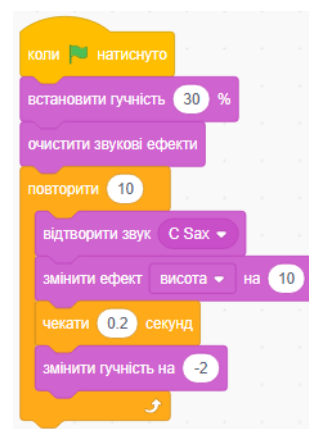


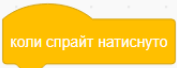
Рис. 2.11 – Циклічне повторення звучання інструмента

Тепер розглянемо випадок необхідності *«реакції» персонажа на зовнішній вплив*. Наприклад, спрайт рухався, а після натискання певної клавіші клавіатури чи на сам спрайт повинна відбутися певна дія.

**Приклад 2.3.** Створити анімацію за умовою: спрайт рухається по сцені, при натисканні на клавішу «пропуск» замислюється, а при натисканні на нього підстрибує.

► Для візуалізації руху спрайта потрібно змінювати його образи. Інколи, щоб краще побачити зміни, можна застосовувати команду «чекати... секунд» категорії «Керувати». Для постійного руху персонажа застосовується цикл «завжди».

Використання датчика «клавішу ПРОПУСК натиснуто?» (див. рис. 2.12) передбачає використання розгалуження Тут візьмемо розгалуження «якщо».

Команда  ініціалізує початок окремого блока. Саме в ньому запрограмуємо підстрибування. Для кращої візуалізації стрибка потрібно додати паузи очікування («чекати...», «подумати...» і т.п.) між окремими рухами.

Приклад коду програми наведено на рис. 2.13. ◀

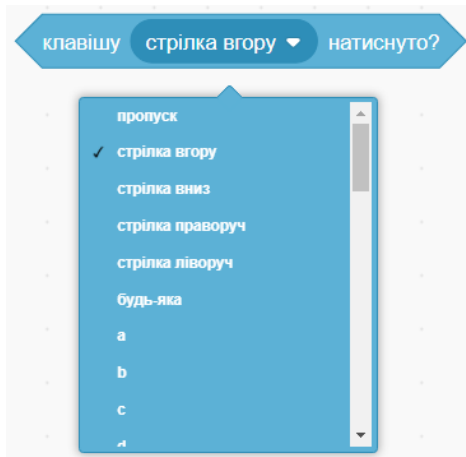


Рис. 2.12 Датчик «клавішу... натиснуто?»

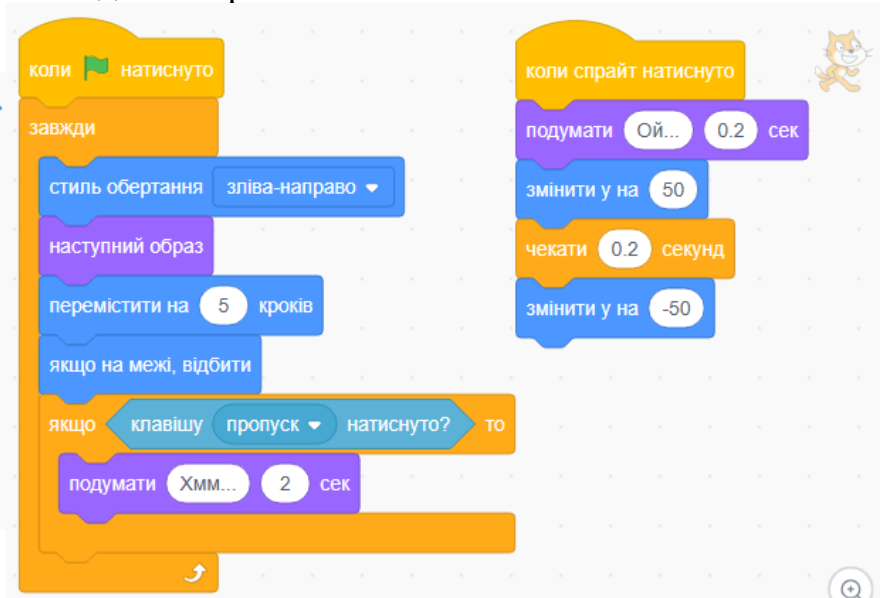


Рис. 2.13 – Приклад коду з реакцією рухомого персонажа на зовнішній вплив

**Завдання 2.4.** Складіть код програми з рис. 2.13. Проаналізуйте дії персонажа та його реакції на зовнішній вплив.

### Контрольні питання до лабораторної роботи №2

1. Які категорії команд дозволяють створювати звукові ефекти в середовищі Scratch? Які з них є основними, а які потребують приєднання через додаткові розширення? Розкажіть про функціонал команд кожної з цих категорій.
2. Спрайти з якого каталогу бібліотеки наділені спеціальні звуковими ефектами для створення танцювальної анімації? Наведіть приклад коду танцюючого персонажа.
3. Спрайти якого каталогу бібліотеки дозволяють створювати музичну композиції? Наведіть приклад коду програми з музичною композицією.
4. Наведіть приклад коду програми реалізації анімації з реакцією рухомого персонажа на зовнішній вплив.

### Завдання для самостійного виконання

Завантажити додаткове розширення «Олівець». Ознайомитися з засобами створення проєктів малювання. Виконати наступне завдання.

Створити програму, по якій спрайт намалює на сцені правильний  $(N + 2)$  – кутник довільного кольору, де  $N$  – номер Вашого варіанту.

## Лабораторна робота №3

**Тема:** Створення анімацій з використанням циклів, розгалужень та датчиків

**Мета:** Ознайомитися зі створенням програм з циклами та розгалуженням. Навчитися створювати анімації з використанням датчиків.

## Теоретичні відомості

- ✓ Розглянемо команди категорії «Керувати» (рис. 1.8).  
Функціонал команд циклічної структури розкрито в рис. 3.1.

|  |                                    |  |                                                                                 |
|--|------------------------------------|--|---------------------------------------------------------------------------------|
|  | Цикл з відомою кількістю повторень |  | Цикл «завжди», який виконується, поки його не буде зупинено за допомогою кнопки |
|  | Цикл з передумовою                 |  | за допомогою кнопки панелі інструментів або командою                            |

Рис. 3.1– Команди циклічної структури

Приклад використання циклічної структури наведено на рис. 3.2.

**Завдання 3.1.** Складіть програму з рис. 3.2. Поясніть необхідність використання кожної з команд та принцип роботи циклу.

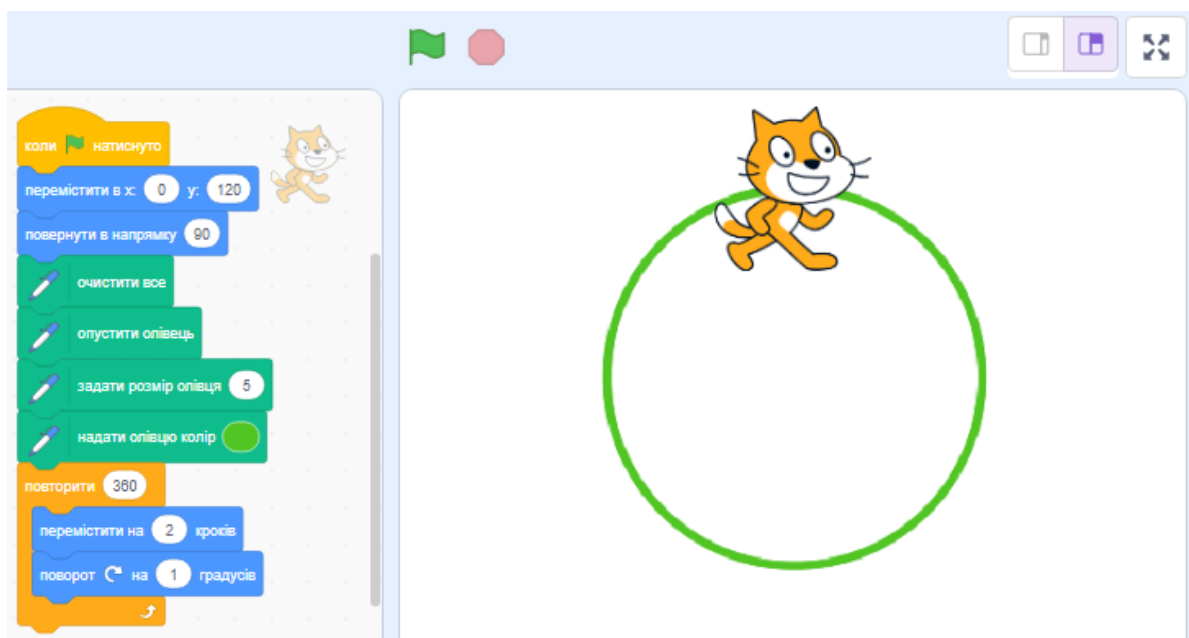


Рис. 3.2 – Код програми з використанням команди циклічної структури і розширення «Олівець»

Функціонал команд розгалуженої структури розкрито в рис. 3.3. **Ошибка! Источник ссылки не найден.,**

Функціонали команди очікування, зупинки та керування клонами відображено в рис. 3.4.

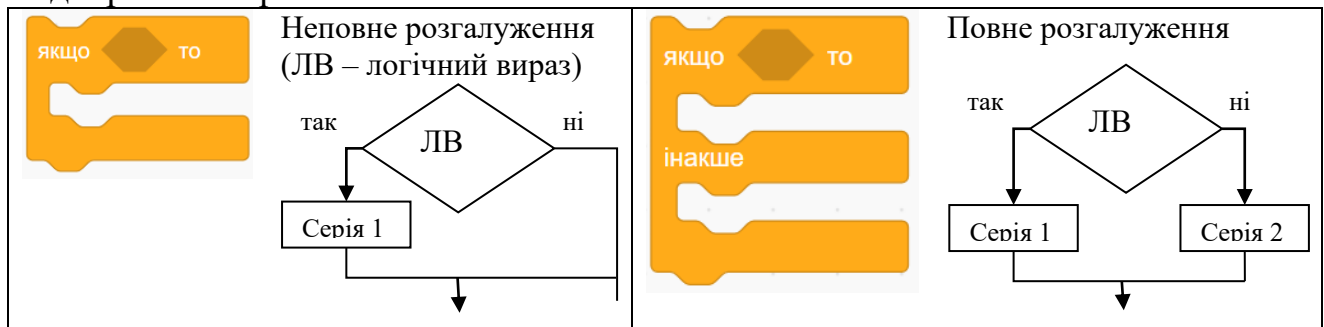


Рис. 3.3 – Команди розгалуженої структури

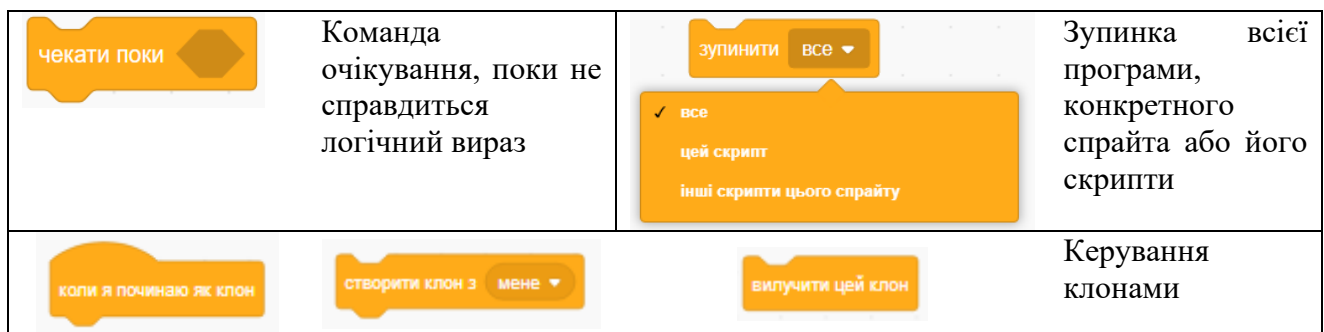


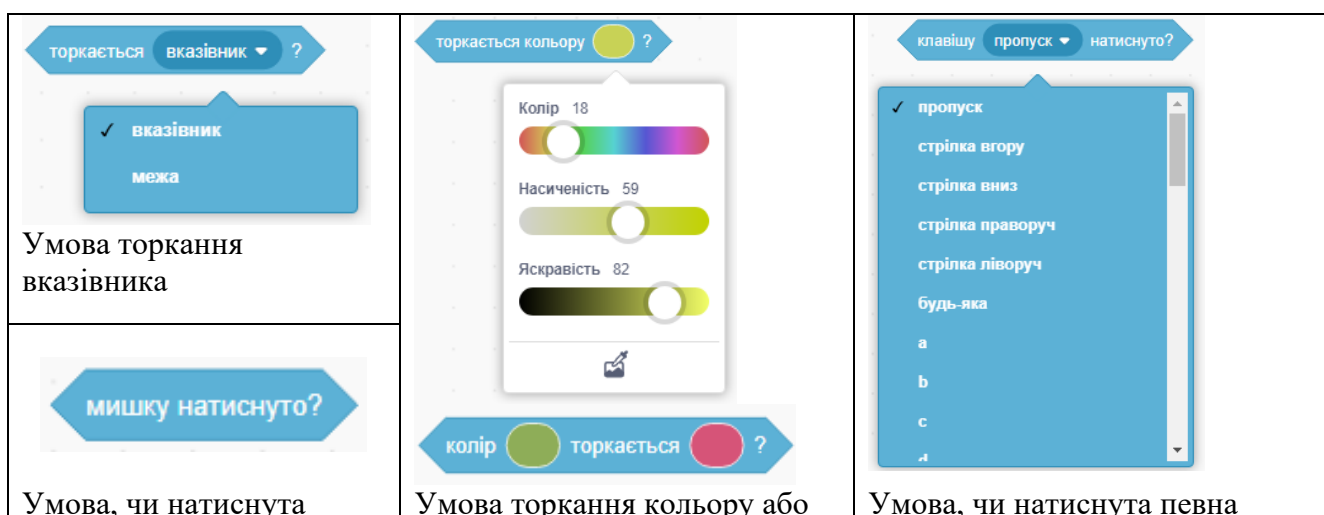
Рис. 3.4 – Команди очікування, зупинки, керування клонами

Як цикли, так і розгалуження можуть бути вкладеними один в одного, наприклад, як показано на рис. 3.5.

✓ Розглянемо команди категорії «Датчики» (рис. 1.9). Деталі щодо функціоналу цих команд розкрито в рис. 3.6 і рис. 3.7.



Рис. 3.5 – Приклади вкладених структур



|      |                           |          |
|------|---------------------------|----------|
| миша | взаємне торкання кольорів | клавіша: |
|------|---------------------------|----------|

Рис. 3.6 – Команди категорії «Датчики»: логічні вирази, що перевіряють певну умову

|                                                         |                                                                                                                                             |
|---------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Ідентифікатори значень відповідних характеристик</p> | <p>Перезапуск таймеру, дозвіл на перетягування</p> <p>Постановка питання, на яке користувач дає відповідь, значення якої отримує змінна</p> |
|---------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|

Рис. 3.7– Команди категорії «Датчики»: ідентифікатори значень, перезавантаження та дозволу, постановки питань для отримання відповідей

- ✓ Розглянемо команди категорії «Оператори» (рис. 1.9). Деталі щодо функціоналу цих команд розкрито в рис. 3.8.

|                                                  |                                               |                                                             |
|--------------------------------------------------|-----------------------------------------------|-------------------------------------------------------------|
| <p>Дії над числовими константами та змінними</p> | <p>Логічні вирази</p> <p>Логічні операції</p> | <p>Дії з числовими та текстовими константами й змінними</p> |
|--------------------------------------------------|-----------------------------------------------|-------------------------------------------------------------|

Рис. 3.8 – Класифікація команд категорії «Оператори»

### Завдання до лабораторної роботи №3

Ознайомитися з командами категорій «Логіка», «Датчики», «Оператори». Навчитися створювати анімації з використанням циклів, розгалужень та датчиків. Виконати завдання свого варіанту (номер варіанту збігається з номером в списку групи).

### Варіант 1

1. Створити простенький (на 5 питань) чат-бот. Тематика питань довільна. Обов'язкові умови: використання операторів розгалуження; використання в питаннях отриманих відповідей.
2. Створити гру, в якій їжачок рухається по сцені (керування з клавіатури) та збирає розкидані по сцені яблука. При торканні кульки «з'їдає» яблучко (говорить «Хрум!», після чого яблучко зникає). Гра зупиняється за лічильником часу (час оберіть самостійно).

### **Варіант 2**

1. Створити простенький (на 5 питань) чат-бот. Тематика питань довільна. Обов'язкові умови: використання операторів розгалуження; використання в питаннях отриманих відповідей.
2. Створити гру, в якій довільним чином по сцені переміщуються 3 миші (невеликого розміру). На сцені є кіт, рухом якого керують з клавіатури. Кіт має зловити мишей – доторкнутися миші, нявкнути, після чого миша має зникнути. Гра зупиняється з клавіатури.

### **Варіант 3**

1. Створити простенький (на 5 питань) чат-бот. Тематика питань довільна. Обов'язкові умови: використання операторів розгалуження; використання в питаннях отриманих відповідей.
2. Створити гру, в якій спрайт рухається по сцені довільним чином. На сцені розташовані повітряні кульки різних кольорів (обрати кульки невеликого розміру). При торканні кульки спрайт змінює свій колір на колір кульки, при цьому видає якийсь звук. Гра зупиняється з клавіатури.

### **Варіант 4**

1. Створити простенький (на 5 питань) чат-бот. Тематика питань довільна. Обов'язкові умови: використання операторів розгалуження; використання в питаннях отриманих відповідей.
2. Створити гру, в якій по сцені справа наліво на різній висоті летять повітряні кульки на деякій відстані одна від іншої. Спрайт має підстрибнути (керування рухом здійснюється з клавіатури), доторкнутися кульки, після чого вона зникає, а спрайт говорить: "Yes!!!". Створити лічильник кульок, що зникли. Гра зупиняється з клавіатури.

### **Варіант 5**

1. Створити простенький (на 5 питань) чат-бот. Тематика питань довільна. Обов'язкові умови: використання операторів розгалуження; використання в питаннях отриманих відповідей.
2. Створити гру, в якій спрайт рухається по сцені довільним чином. На сцені розташовані повітряні кульки синього кольору (обрати кульки невеликого розміру). Коли спрайт знаходиться від кульки на відстані менше 20, кулька змінює колір на червоний, а спрайт дивується та розвертається в протилежний бік. Гра зупиняється з клавіатури.

### **Варіант 6**

1. Створити простенький (на 5 питань) чат-бот. Тематика питань довільна. Обов'язкові умови: використання операторів розгалуження; використання в питаннях отриманих відповідей.
2. Створити лабіринт, по якому спрайт має дійти до скарба. Керування спрайтом здійснюється з клавіатури. Реалізуйте випадок торкання стінок лабіринта. Гра зупиняється за лічильником часу (час оберіть самостійно).

### **Варіант 7**

1. Створити простенький (на 5 питань) чат-бот. Тематика питань довільна. Обов'язкові умови: використання операторів розгалуження; використання в питаннях отриманих відповідей.
2. Створити гру, в якій спрайт має лопнути 7 повітряних кульок (обрати кульки невеликого розміру), які розташовані на сцені. Керування спрайтом здійснюється з клавіатури. Гра зупиняється за лічильником часу (час оберіть самостійно).

### **Методичні рекомендації до лабораторної роботи №3**

**Приклад 3.1.** Створити програму, яка реалізує діалог між користувачем і спрайтом Giga. Персонаж ставитиме різні питання, реакція на які буде залежати від відповіді.

► Реалізація програми та її блок-схеми показана на рис. 3.9. ◀

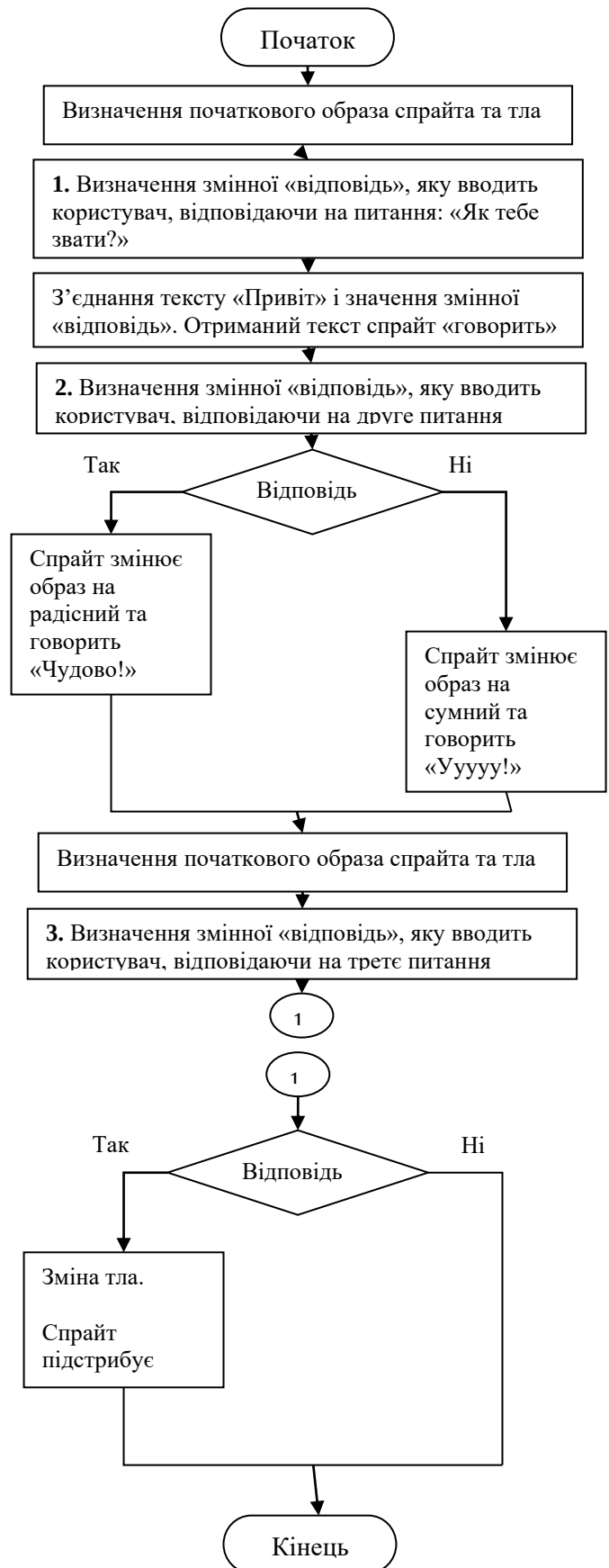
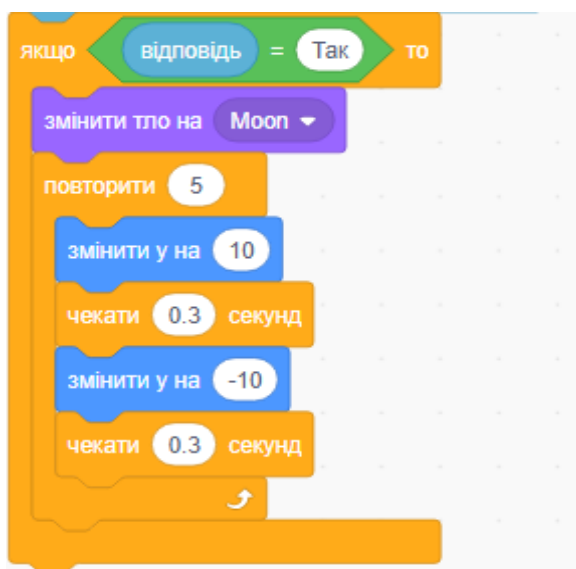


Рис. 3.9 – Код програми діалогу між спрайтом та користувачем з діями, що залежать від відповідей

**Завдання 3.2.** Скласти запроповану вище програму (рис. 3.9), провести її аналіз. Доповнити програму супроводжуючими ефектами перетворення тексту в мову. Відповідні доповнення на початку програми проілюстровано на рис. 3.10.

**Приклад 3.2.** Скласти таку гру:

- на сцені з'являються м'ячі (напрямок руху задамо горизонтальний),
- спрайт має торкнутися м'яча, після чого гавкнути, а м'яч має зникнути, а інший – з'явитися,
- керування рухом м'яча реалізується з клавіатури,
- здійснюється підрахунок спійманих м'ячів (кількість очок),
- гра закінчується за таймером через 30 секунд.

► Орієнтовні скрипти для спрайта Dog2 та м'яча наведено в рис. 3.12. ◀

**Завдання 3.4.** Складіть код програми з рис. 3.12. Проаналізуйте його. Додайте інші ефекти реакції на торкання спрайта до м'яча, змініть напрям руху м'яча. Змініть персонажа та об'єкт. По завершенню зробіть так, щоб спрайт сказав, скільки він набрав очок, наприклад, при натисканні клавіші «пропуск» (див. рис. 3.11).

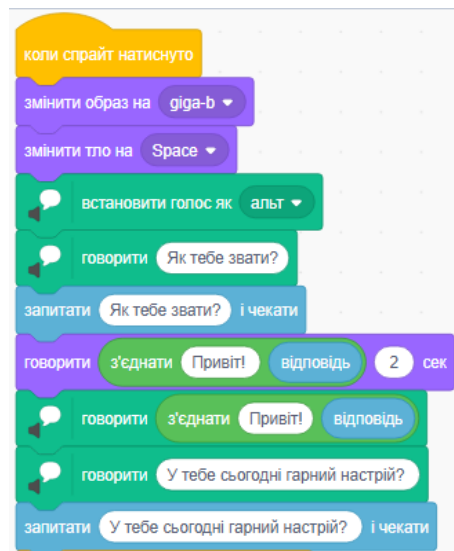


Рис. 3.10 – Додавання до програми кодів ефектів перетворення тексту в мову



Рис. 3.11 – Повідомлення про кількість набраних очок

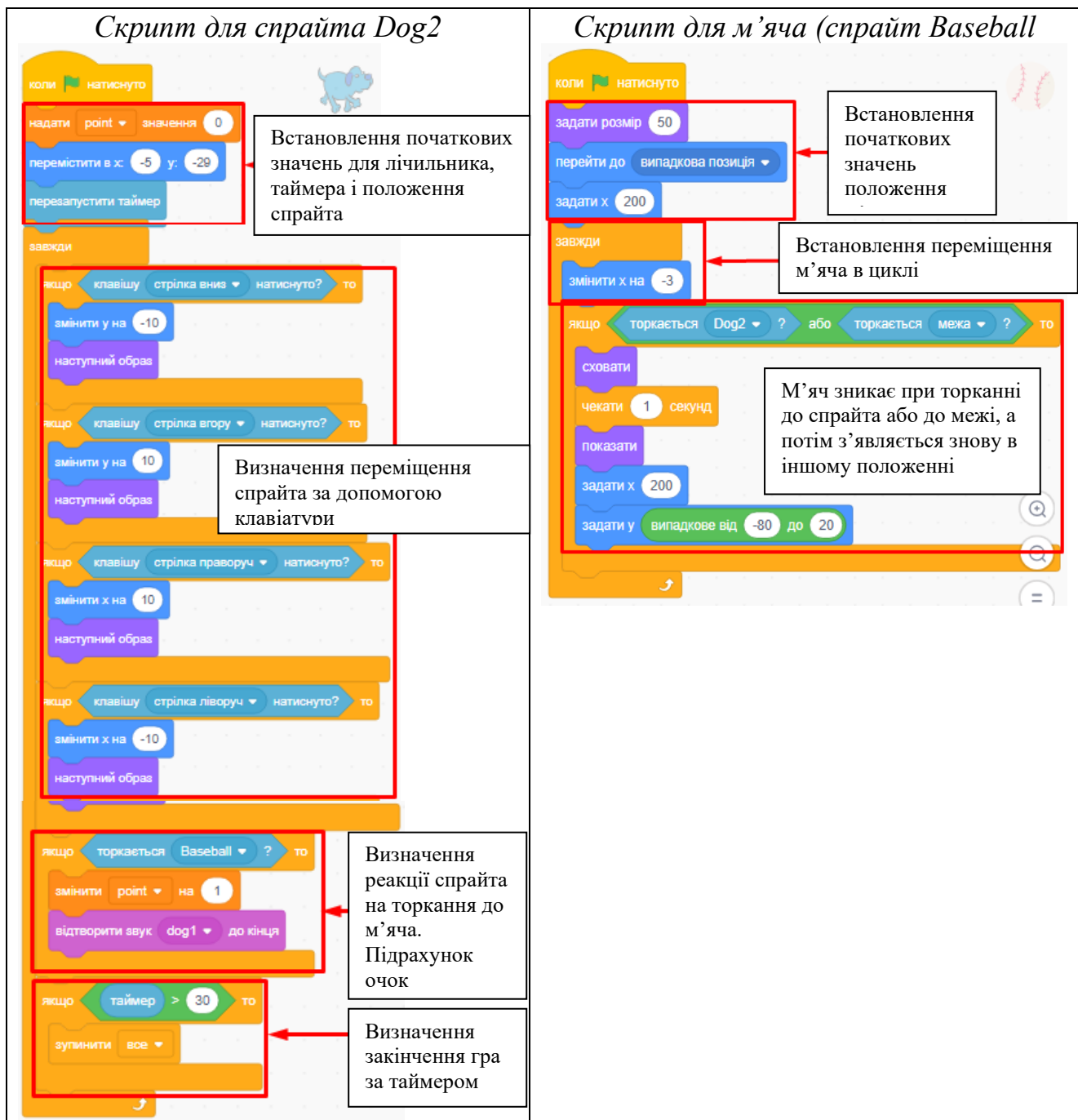


Рис. 3.12 – Код програми гри спрайта з м'ячем

**Зауваження 3.1.** Для підрахунку очок в програмі рис. 3.12 встановимо змінну point для лічильника. Робота з лічильником вивчалася на дисципліні «Основи програмування». Щодо роботи зі змінними в середовищі Scratch, то це питання детальніше буде розглянуто в наступній лабораторній роботі.

### Контрольні питання до лабораторної роботи №3

1. Які типи команд розміщено в категорії «Керування»?
2. Які типи циклічних структур має Scratch? Опишіть принципи їх роботи.
3. Які типи структур розгалужень має Scratch? Опишіть принципи їх роботи.
4. Які команди Scratch дозволяють створити логічні вирази?

5. Які команди дозволяють визначити математичні дії з числовими константами та змінними? З текстовими константами та змінними?
6. Опишіть принцип роботи команди «з'єднати...».
7. Опишіть алгоритм створення лічильника.
8. Яка команда дозволяє поставити питання, відповідь на яке зберігається як значення змінної «відповідь»?
9. Які команди відповідають за роботу з таймером?  
Яка команда дозволяє встановити взаємодію між спрайтами?

## Лабораторна робота №4

**Тема: Створення гри в середовищі Scratch. Робота зі списками**

**Мета:** Ознайомитися зі створенням програм з використанням змінних та списків. Навчитися створювати проекти з їх використанням.

### Теоретичні відомості

- ✓ Розглянемо команди категорії «Змінні» (рис. 1.11). Деталі щодо функціоналу цих команд розкрито в рис. 4.1.

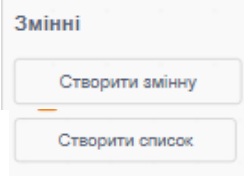
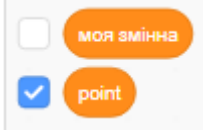
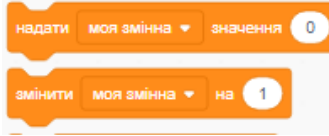
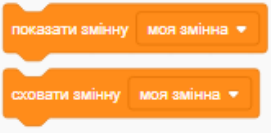
|                                                                                                                        |                                                                                                                   |                                                                                                                                 |                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p>Створення змінних та списків</p> |  <p>Ідентифікатори змінних</p> |  <p>Присвоєння та зміна значень змінних</p> |  <p>Показування/приховування значень змінних в лівому верхньому куті сцени</p> |
|------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Рис. 4.1 – Функціонал команд категорії «Змінні»

На рис. 4.2 продемонстровано послідовність дій від створення змінної до результату, який з'являється в категорії «Змінні».

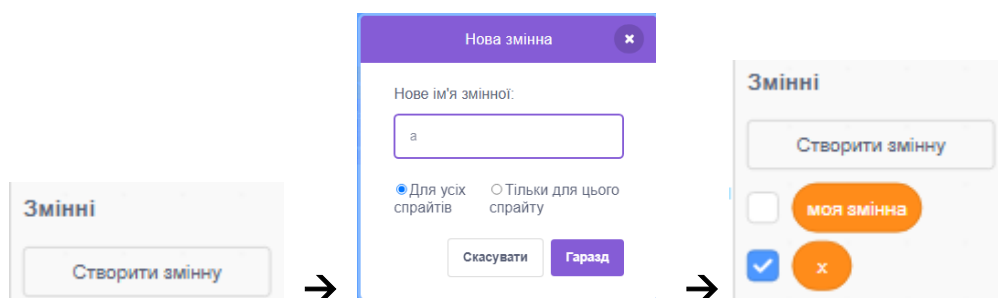


Рис. 4.2 – Створення змінної

На рис. 4.3 продемонстровано послідовність дій щодо створення списку.

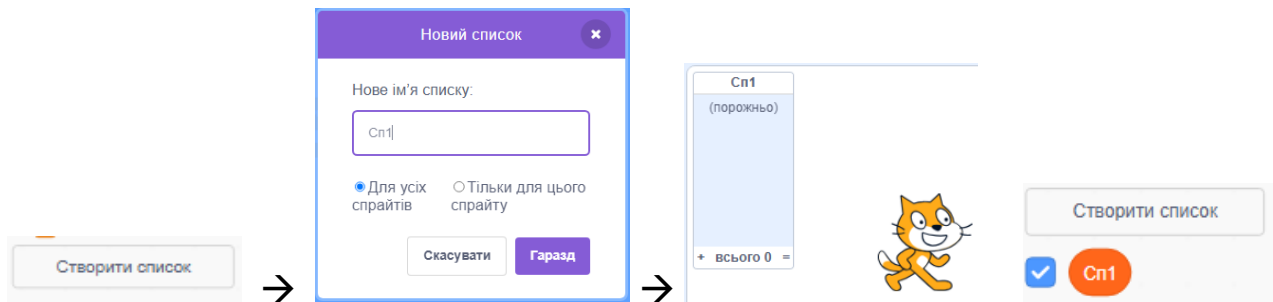


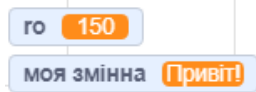
Рис. 4.3 – Створення порожнього списку

Як тільки створюється список, відразу в категорії «Змінні з'являються команди для роботи зі списками, наведені в рис. 4.4. Тип змінних визначається

в процесі присвоєння їй значення. Наприклад, операції



призведуть до результату:

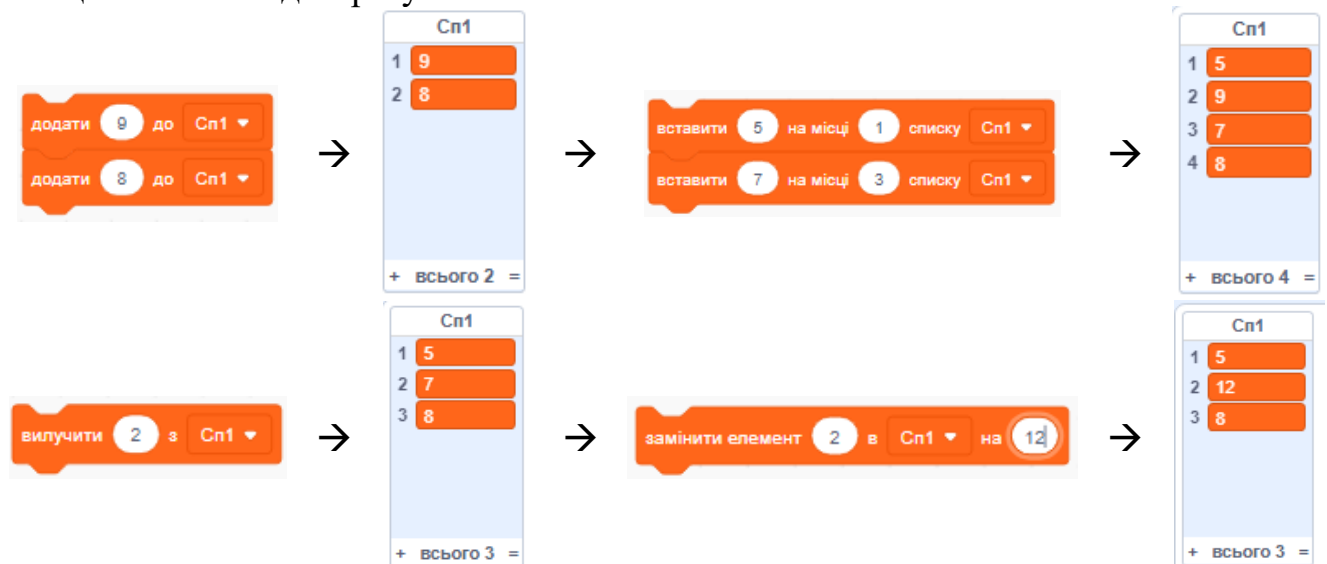


Тобто змінна «моя змінна» має текстовий тип, а «го» - ціле число.

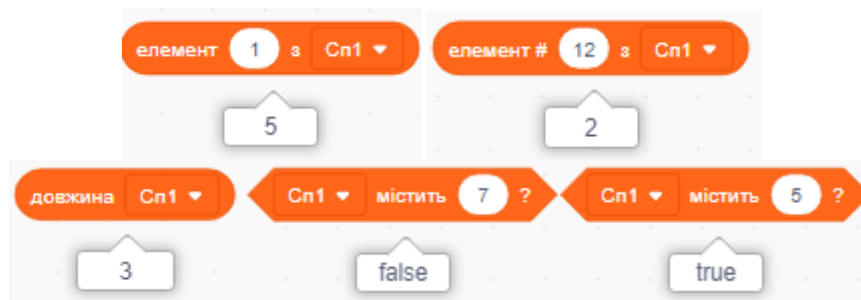
|                                                                                                      |                                                                                                          |                                                                                                |                                                                           |
|------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| <div> <div>додати щось до Сп1</div> <div>вилучити 1 з Сп1</div> <div>вилучити все з Сп1</div> </div> | <div> <div>вставити щось на місці 1 списку Сп1</div> <div>замінити елемент 1 в Сп1 на щось</div> </div>  | <div> <div>елемент 1 з Сп1</div> <div>елемент # щось з Сп1</div> <div>довжина Сп1</div> </div> | <div> <div>показати список Сп1</div> <div>сховати список Сп1</div> </div> |
| <p>Додавання та вилучання елементів списку</p>                                                       | <p>Робота з окремими елементами списку</p> <p>Логічний вираз перевірки належності елементу до списку</p> | <p>Ідентифікатор елементів, номерів елементів та довжини списку</p>                            | <p>Показування/приховування списку в лівому верхньому куті сцени</p>      |

Рис. 4.4 – Функціонал команд підкатегорії «Списки» в категорії «Змінні»

Покажемо на прикладі принципи роботи деяких команд. Запишемо ланцюжок команд та результатів їх виконання:



Далі наведемо результати роботи деяких команд для отриманого вище списку (результати підсвічуються знизу):



Команда  робить список порожнім.

**Завдання 4.1.** Реалізуйте ланцюжок команд, аналогічний до наведеного вище, і усвідомте принцип роботи таких команд.

Поставимо задачу введення списку оцінок учнів групи відповідно до їх порядкового номеру. На рис. 4.5 наведено приклад реалізації задачі.

**Завдання 4.2.** Складіть запропоновану на рис. 4.5 програму. Проаналізуйте принципи її роботи.

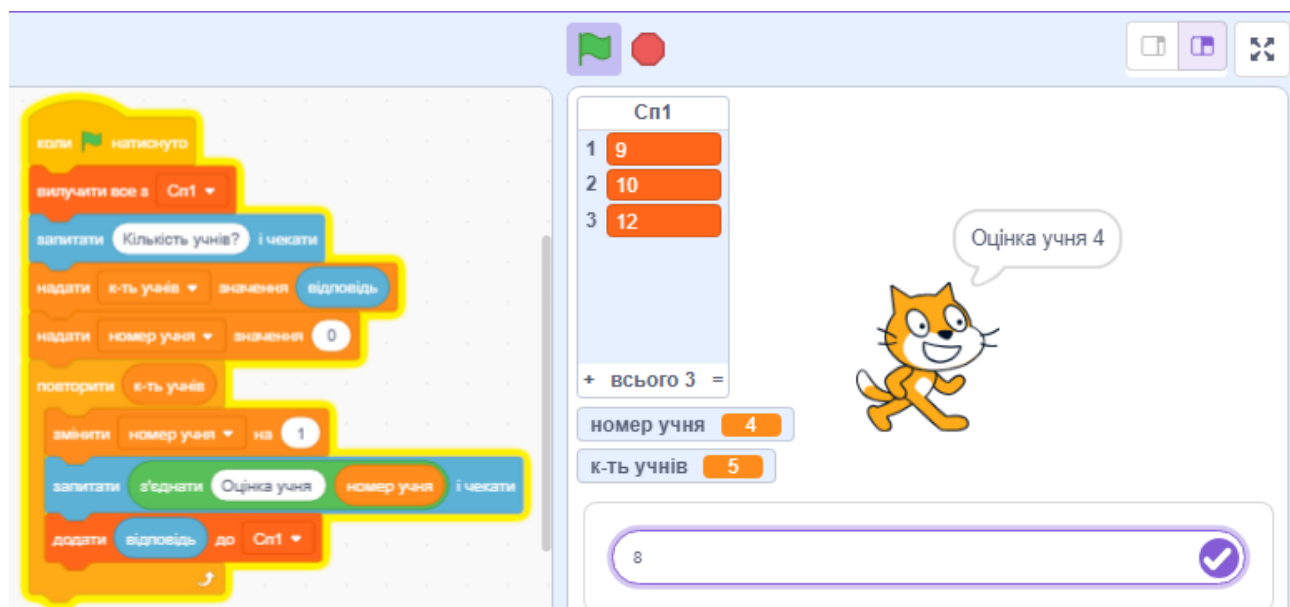


Рис. 4.5 – Складання списку оцінок учнів групи

## Завдання до лабораторної роботи №4

Ознайомитися з командами категорій «Змінні». Навчитися створювати проекти з використанням змінних та списків. Виконати завдання свого варіанту (номер варіанту збігається з номером в списку групи). Перша задача в кожному варіанті НЕ ВИМАГАЄ створення списків. Друга задача – на використання списків.

### **Варіант 1**

1. Створити навчальний проєкт: спрайт просить вводити натуральні числа по черзі з клавіатури (всього 10 чисел). Слід порахувати і повідомити загальну кількість введених парних та непарних чисел.

2. В 8 коробках лежать іграшки (випадкова кількість від 10 до 20). Знайдіть коробку з найменшою парною кількістю іграшок.

### **Варіант 2**

1. Створити навчальний проєкт: спрайт просить вводити натуральні числа по черзі з клавіатури (всього 10 чисел). Слід порахувати і повідомити загальну кількість введених чисел, кратних 5.

2. В 8 коробках лежать іграшки (випадкова кількість від 10 до 20). Знайдіть коробку з найбільшою кількістю іграшок.

### **Варіант 3**

1. Створити навчальний проєкт: спрайт просить додати 2 натуральних числа (всього 10 прикладів). За правильну відповідь учень отримує 1 бал. Наприкінці спрайт повідомляє набрану кількість балів.

2. В 10 коробках лежать іграшки (випадкова кількість від 10 до 20). Знайдіть кількість іграшок в коробках з парними номерами.

### **Варіант 4**

1. Створити навчальний проєкт: спрайт просить вводити натуральні числа по черзі з клавіатури (всього 10 чисел). Слід порахувати і повідомити загальну введених чисел, більших за 50.

2. В 10 коробках лежать 3. іграшки (випадкова кількість від 10 до 20). Перевірити, чи є коробки з однаковою кількістю іграшок.

### **Варіант 5**

1. Створити навчальний проєкт: спрайт задає прості приклади на арифметичні дії (всього 10 прикладів). Слід порахувати і повідомити кількість набраних балів, якщо за кожну правильну відповідь нараховується 1 бал.

2. В 10 коробках лежать іграшки (випадкова кількість від 10 до 20). Порахувати кількість іграшок в тих коробках, де іграшок непарна кількість.

### **Варіант 6**

1. Створити навчальний проєкт: спрайт просить вводити натуральні числа по черзі з клавіатури (всього 10 чисел). Слід порахувати і повідомити загальну кількість двозначних чисел серед введених.

2. Створити список з 20 випадкових двозначних чисел. Переписати цей список за правилом – замість кожного парного числа записати 0, замість кожного непарного – 1.

### **Варіант 7**

1. Створити навчальний проєкт: спрайт просить вводити натуральні числа по черзі з клавіатури (всього 10 чисел). Слід порахувати і повідомити загальну кількість чисел серед введених, які діляться на 3, але не діляться на 7.

2. Створити список з 20 випадкових двозначних чисел. Знайдіть суму чисел списку, які перевищують суму його крайніх елементів.

### Методичні рекомендації до лабораторної роботи №4

**Приклад 4.1.** Скласти програму для перевірки знань учнів з таблиці множення.

► Для виконання цього завдання нам потрібно працювати зі змінними (рис. 1.11), які ми маємо створити відповідно до схеми на рис. 4.2.

Щоб персонаж повідомив, яку дію має зробити учень, потрібно з'єднати одну змінну (a), знак «\*», іншу змінну (b) та текст «= ?», які заключити в блок «запитати.. і чекати», який дозволяє визначити значення змінної «відповідь»:



Структура повного розгалуження дозволяє обрати коментар для реакції на правильність відповіді.

Орієнтовний код програми наведено на (рис. 4.6). ◀

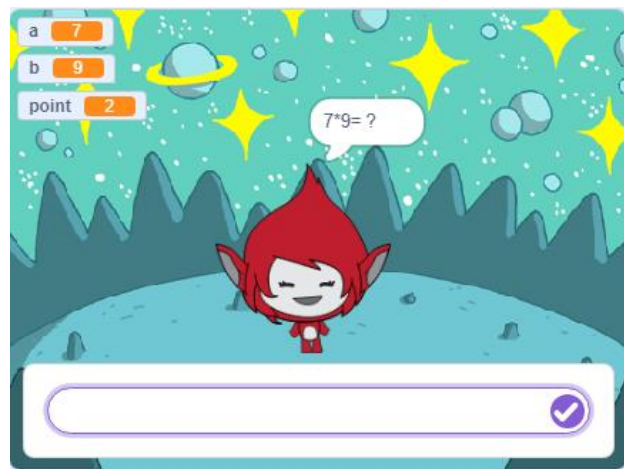
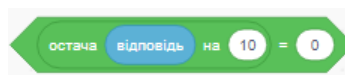


Рис. 4.6 – Гра для перевірки знань таблиці множення

**Завдання 4.3.** Складіть код, запропонований на рис. 4.6 та проаналізуйте його.

**Приклад 4.2.** Створити навчальний проєкт: спрайт просить вводити натуральні числа по черзі з клавіатури (всього 7 чисел). Слід порахувати і повідомити загальну кількість чисел, кратних 10.

► Використаємо персонаж з попереднього прикладу. Наш спрайт буде по черзі 7 разів запитувати числа. Ознакою подільності числа на 10 є нульове значення залишку від ділення цього числа на 10. Тобто, логічний вираз



має набувати істинне значення. Будуть підраховані числа, для яких це виконується, за допомогою лічильника point. Див. рис. 4.7 – код програми підрахунку чисел, кратних 10 ◀

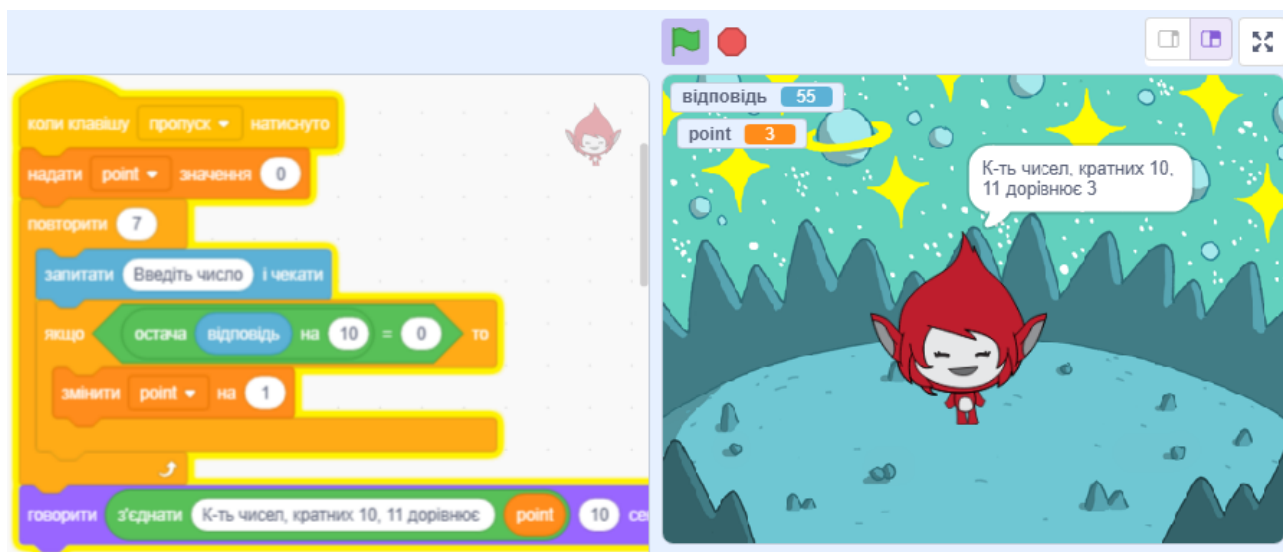
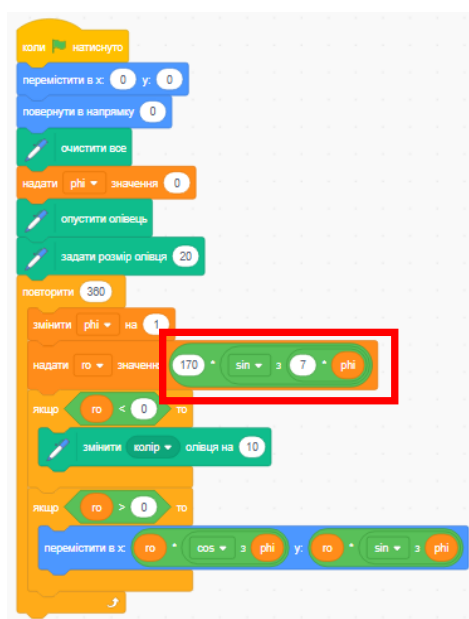


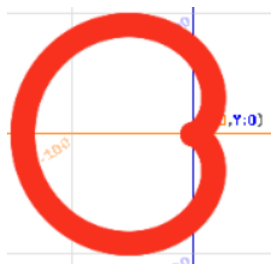
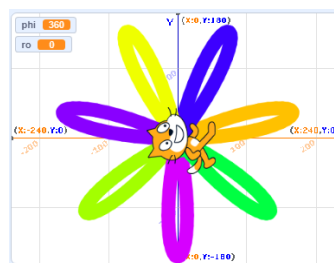
Рис. 4.7 – Код програми підрахунку чисел, кратних 10

**Завдання 4.4.** Складіть наведену вище програму, проаналізуйте принцип її роботи. Додайте перетворення тексту в мову. Додайте ефекти реакції спрайта на введення чисел, що задовольняють умову.

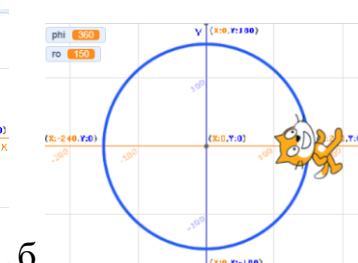
**Завдання 4.5.** Повторіть тему «Полярна система координат» з дисципліни «Математичний аналіз». Нагадаємо, що перехід до декартової системи координат реалізується за формулами:  $x = \rho \cos \varphi$ ;  $y = \rho \sin \varphi$ ;  $\rho \geq 0$ . Складіть програму з рис. 4.8, а. Проаналізуйте алгоритм, принципи роботи програми та отримайте результат у вигляді 7-листка (рис. 4.8, б).



б

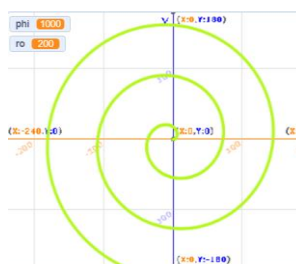


б



в

г



д

Рис. 4.8 – Побудова ліній, заданих в полярній системі координат

Замість блока, виділеного червоним прямокутником, внесіть рівняння інших ліній в полярній системі координат, наприклад,  $\rho = 150$  (коло, рис. 4.8, в)  $\rho = 70(1 - \cos \varphi)$  (кардіоида, рис. 4.8, г),  $\rho = 0.1\varphi$  (спіраль Архімеда, рис. 4.8, д).

**Приклад 4.3.** Утворити список (від 5 до 10 елементів) з випадкових чисел, знайти максимальний елемент і його номер у списку. Спрайт має повідомити результат.

► В коді програми на рис. 4.9 змінна «номер» визначає номер елемента у списку, змінна «макс» відповідає найбільшому елементу списку, а «НомМакс» – номеру цього елемента. ◀

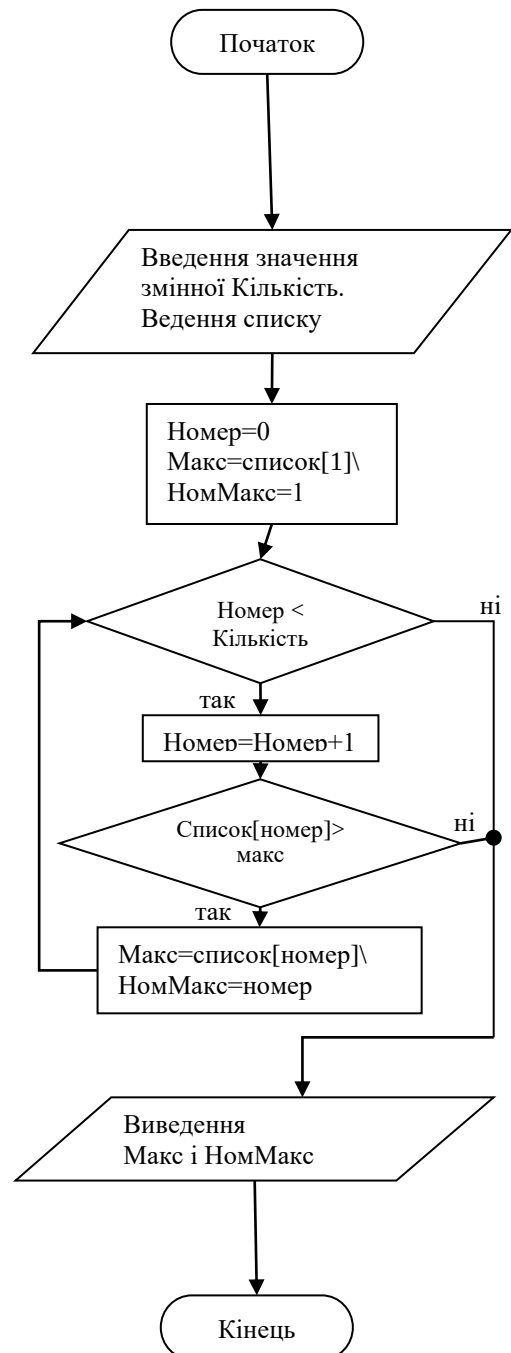


Рис. 4.9 – Код та блок-схема програми пошуку найбільшого елемента списку та його номера

**Завдання 4.6.** Реалізувати програму на рис. 4.9 та провести аналіз її роботи.

### Контрольні питання до лабораторної роботи №4

1. Опишіть, які команди містить категорія «Змінні».
2. Як створюється змінна?
3. Як створюється список?
4. Опишіть принципи роботи команд роботи зі списком.
5. Як функціонують команди додавання, вставки, заміни та видалення елементів зі списку? Визначення елементів списку за номером і за значенням. Логічні вирази з елементами списку.
6. Яким чином можна перевірити умову подільності одного натурального числа на інше?

### Лабораторна робота №5

#### Тема: Математичні обчислення. Оператори порівняння

**Мета:** Ознайомитись з можливостями використання математичних операторів, констант та функцій категорії «Математика» візуального середовища розробки Google Blockly при створенні програм. Ознайомитись з особливостями логічних блоків. Навчитись використовувати оператори порівняння для створення програм.

#### Теоретичні відомості

Корпорація Google створила свій варіант середовища візуального програмування – Google Blockly. Щоб розпочати роботу у візуальному середовищі розробки, перейдіть за посиланням:

<https://blockly-demo.appspot.com/static/demos/code/index.html#>

Середовище розробки повністю перекладено майже 40 різними мовами.

✓ *Інтерфейс середовища візуального програмування Google Blockly аналогічний до середовища Scrsch (рис. 5.1).*

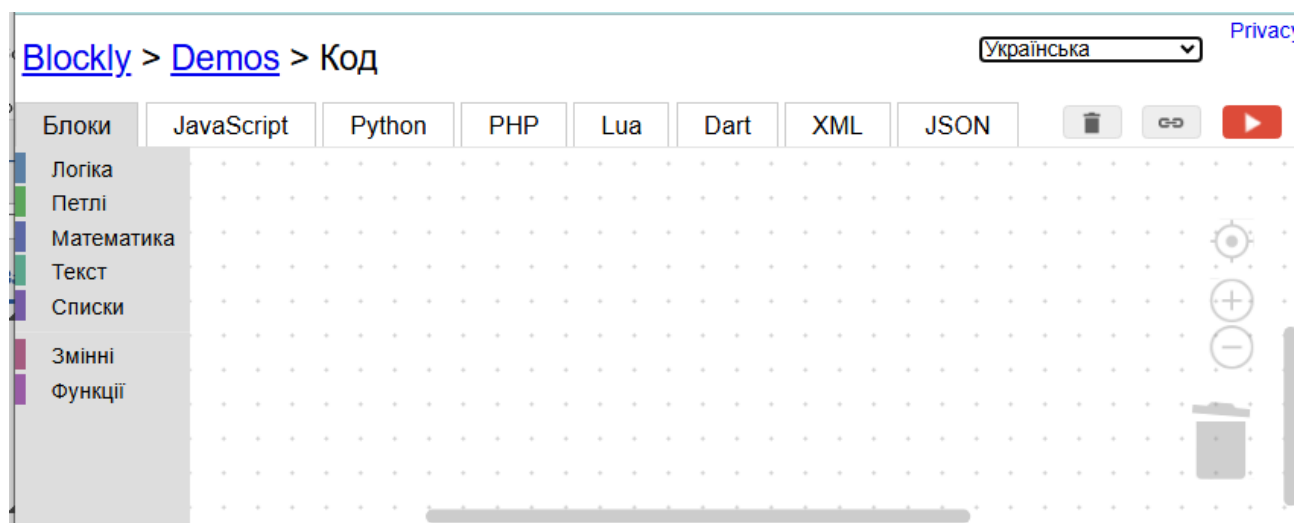


Рис. 5.1 – Інтерфейс середовища візуального програмування: Google Blockly

Всі команди за призначенням розбиті на 8 категорій: від логічних операцій, до роботи з текстом. За цим же принципом блоки розфарбовані у різні кольори, щоб зробити пошук потрібного елементу легшим.

Важливі поради:

- Для запуску програми натисніть ,
- Для збереження посилання на ваш код використайте піктограму .
- Кнопка  дозволяє видалити всі блоки з поля.
- Щоб видалити один блок, виділіть його та натисніть Del, або перетягніть до сміттевого кошика в нижньому правому куті.
- Середовище розуміє такі команди як Ctrl+C та Ctrl+V.
- Цікавою та зручною особливістю є можливість давати змінним імена кирилицею.
- Після створення коду ви можете переглянути, яким він є в таких не візуальних мовах програмування, як JavaScript, Python, Dart або XML.

✓ *Робота зі змінними.* Для створення змінної переходимо до категорії «Змінні», натискаємо на кнопку «Створити змінну», надаємо їй ім'я (назву), після чого з'являється функціонал роботи зі змінною у категорії «Змінні» (рис. 5.2).

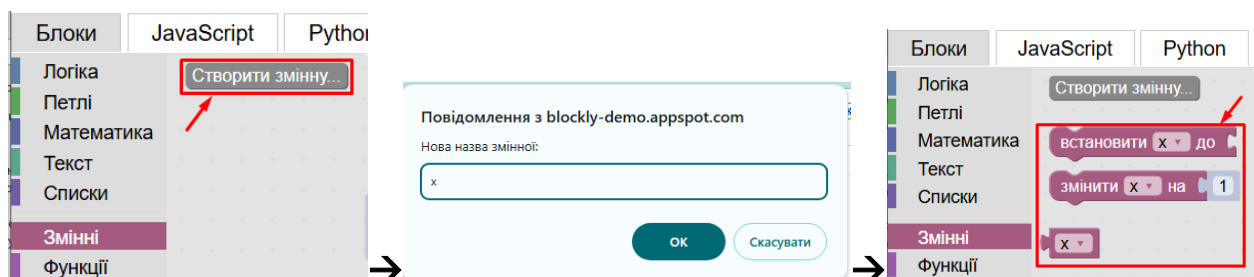


Рис. 5.2 – Створення змінної в середовищі Google Blockly

*Присвоєння значення змінній* можна реалізувати різними способами (рис. 5.3). Тип змінної визначається значеннями, яке їй присвоюється.

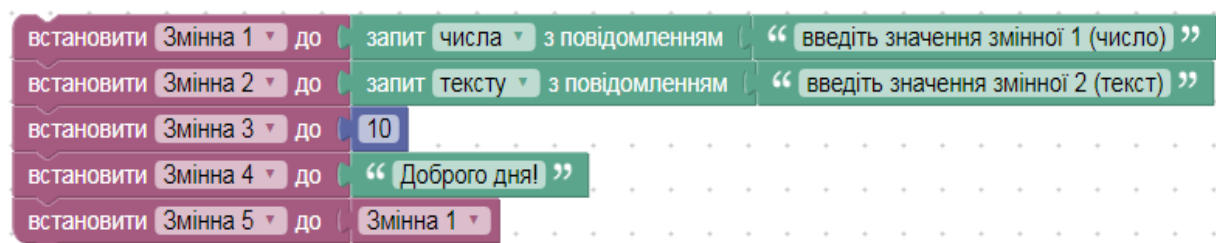


Рис. 5.3 – Присвоєння значення змінній різними способами

**Завдання 5.1** Складіть програму з рис. 5.4, запустіть її на виконання, проаналізуйте.

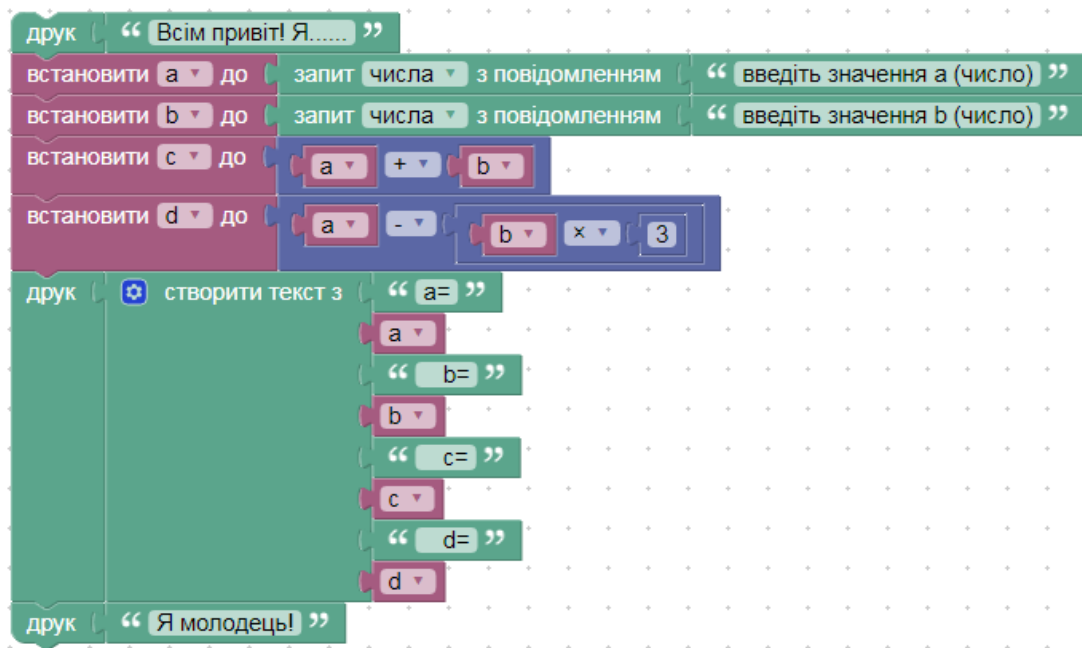


Рис. 5.4 – Програма, яка демонструє виконання арифметичних дій з числами в Google Blockly

✓ *Робота з командами категорії «Математика» (рис. 5.5).*

Google Blockly надає майже необмежені можливості для створення програм, в яких необхідно використовувати математичні формули. В середовищі реалізовано класичні математичні оператори та функції.

Також реалізовано деякі специфічні перевірки чи дії, які не містяться у класичних математичних бібліотеках мов програмування. Наприклад, програмісту надається можливість в одну дію перевірити число на парність/непарність, додатність/від'ємність, чи воно просте, ціле або ділиться на інше число:

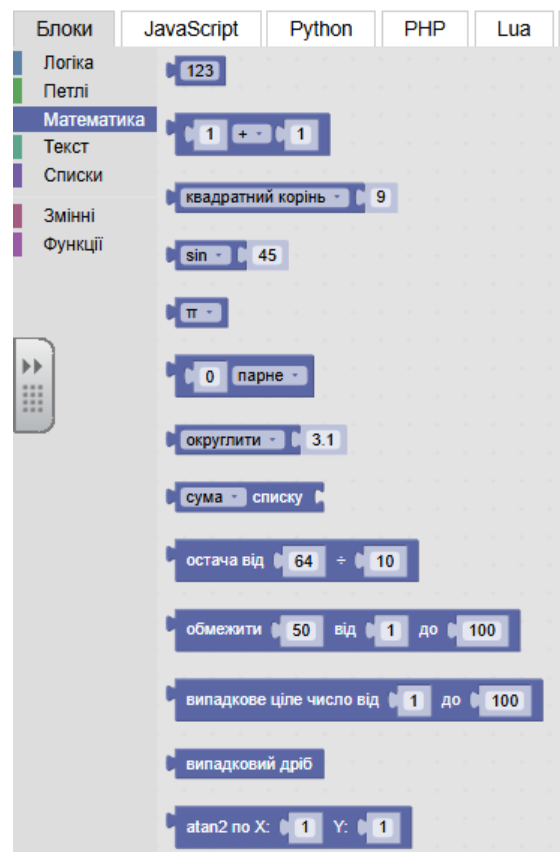
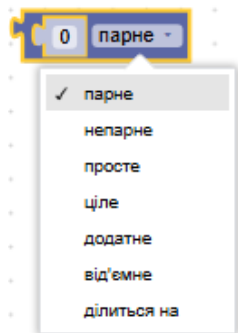


Рис. 5.5 – Команди категорії «Математика»

Візуальне середовище розробки Google Blockly надає такі математичні константи як  $\pi$ ,  $e$ ,  $\infty$ ,  $\phi$  та ін. Реалізовані тригонометричні функції (аргумент яких має градусну міру) та робота з логарифмами і багато інших математичних дій. Є функції, що дозволяють округляти числа до найближчого цілого, створювати псевдовипадкові числа та навіть дробі (див. рис. 5.6).

Особливістю роботи з візуальним середовищем розробки Google Blockly є те, що в кожній дії можуть брати участь не більше ніж дві змінні, тому складні формули доводиться розбивати на кроки. Для того, щоб розбити будь-яку формулу на кроки, слід проаналізувати порядок дій і складати вираз з модулів таким чином, щоб зовні опинились дії, які повинні бути виконані в останню чергу.

✓ *Каталог Logic. Команди категорії «Логіка».*

Google Blockly повноцінно реалізував логічні операції і має в своєму арсеналі стандартний оператор вибору **ЯКЩО ... ТО ... ІНАКШЕ**. Реалізовані всі операції порівняння, логічні **І**, **НЕ** та **АБО**. Тут є такий важливий інструмент, як логічна змінна (**ІСТИНА/ХИБНІСТЬ**), який можна використовувати також і для організації циклів (див. рис. 5.7).

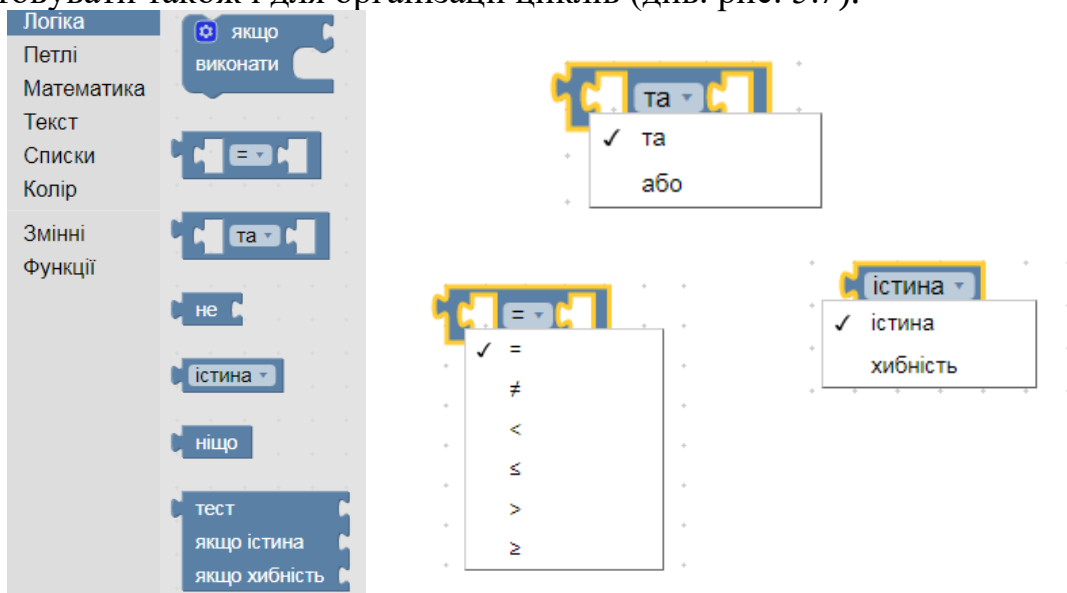


Рис. 5.7 – Команди категорії «Логіка»

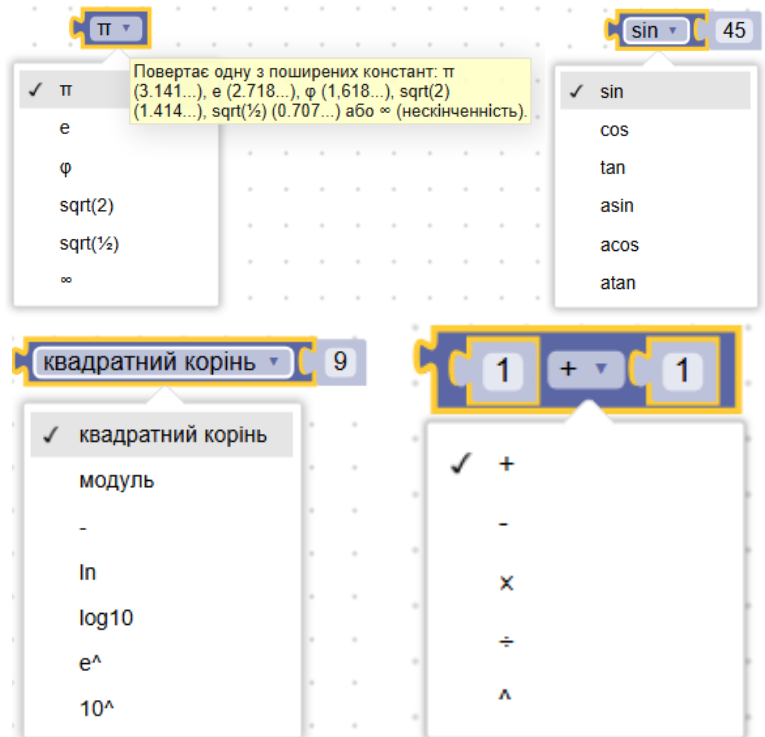


Рис. 5.6 – Математичні операції вз каталогу «Математика»

Слід звернути увагу, що в каталозі лежить спрощена версія оператора **ЯКЩО ... ТО**. Щоб використовувати повну і навіть комплексну версію цього оператора натисніть на коліщатко зліва вгорі і додайте стільки пунктів **ІНАКШЕ ЯКЩО** і **ІНАКШЕ**, скільки вам необхідно (рис. 5.8).

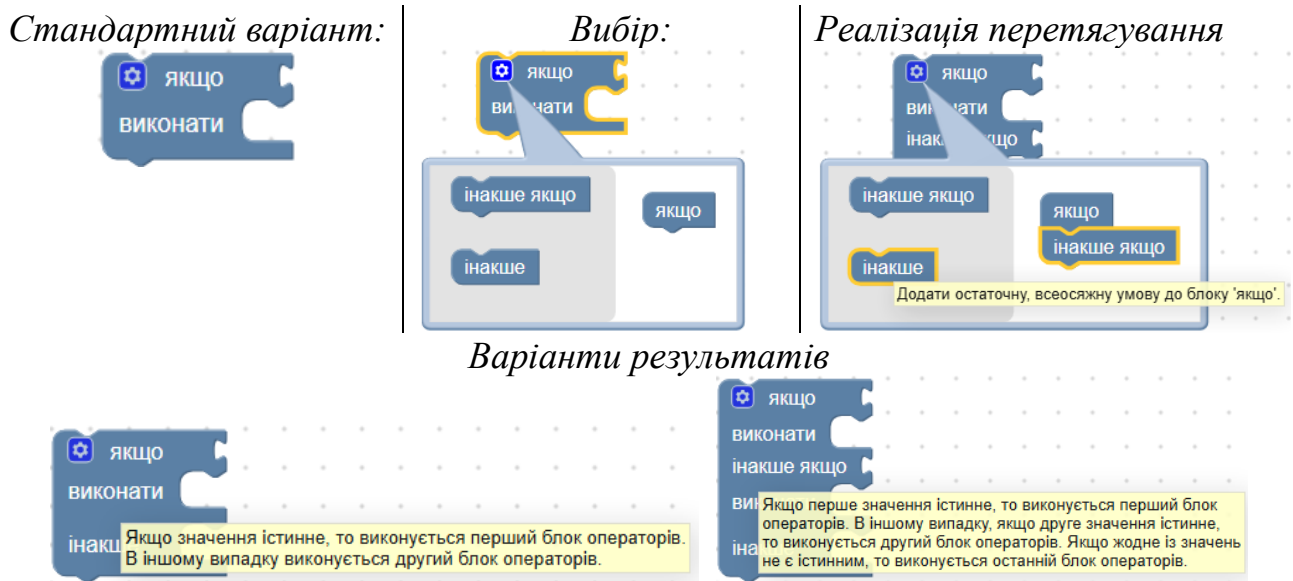


Рис. 5.8 – Утворення різних варіантів структур усови

### Завдання до лабораторної роботи №5

Ознайомитися з командами категорій «Математика» та «Логіка» візуального середовища розробки Google Blockly. Виконати завдання свого варіанту.

#### Варіант 1

1. Задано радіус кола та міра центрального кута. Обчисліть довжину кола й площу круга, площу сектора та сегмента, довжину сторін вписаного та описаного в коло правильного трикутника.
2. Відомі три числа. Перевірте, чи є вони послідовними членами арифметичної чи геометричної прогресії. Виведіть на друк тип прогресії та її крок.

#### Варіант 2

1. Відома довжина сторони правильного трикутника. Обчисліть його периметр, площу, висоту, радіуси вписаного та описаного кіл.
2. Відомі кути трикутника. В наступній послідовності перевірте виконання умов: чи є трикутник правильним, чи є він рівнобедреним або прямокутним. Слід врахувати можливість того, що трикутник одночасно може бути рівнобедреним та прямокутним.

#### Варіант 3

1. Задано радіус кулі та координати її центра у тривимірному просторі. Обчисліть об'єм кулі, площу її поверхні та відстань від центру кулі до початку координат.
2. Відомі довжини трьох сторін. Перевірити, чи можна з них побудувати трикутник. Додатково перевірити, чи не є цей трикутник рівностороннім чи рівнобедреним.

**Варіант 4**

- Відома довжина ребра куба. Обчисліть площу його поверхні, об'єм, а також радіуси, об'єми та площі поверхонь вписаної та описаної куль.
- Написати програму для розв'язання наступної задачі: визначити, чи пройде цеглина розміром  $X \times Y \times Z$  в прямокутне віконце розміром  $A \times B$ .

**Варіант 5**

- Обчисліть об'єм та площу повної поверхні циліндра та конуса за відомими радіусом та висотою.
- Відомі три кути. Перевірити, чи існує трикутник з такими кутами. Додатково перевірити, чи не буде він правильним, рівнобедреним чи прямокутним.

**Варіант 6**

- Задано дві сторони трикутника та кут між ними. Обчисліть міри його невідомих кутів та сторони, периметр та площу фігури.
- Відомі координати точки на осях  $x$  та  $y$ . В наступній послідовності перевірте виконання умов: чи лежить точка на перетині осей  $x$  та  $y$ ; чи лежить точка на одній з координат та на якій; чи лежить точка в одному з чотирьох квадрантів та в якому.

**Варіант 7**

- Задано відстань, швидкість течії та власну швидкість катера. Обчисліть, скільки часу знадобиться катеру, щоб пройти задану користувачем відстань за течією та проти неї.
- Визначити, чи увійде до конверту з внутрішніми розмірами  $a$  і  $b$  мм прямокутна листівка розміром  $c$  і  $d$  мм. Для розміщення листівки в конверті потрібен проміжок в 1 мм з кожного боку.

**Методичні рекомендації до лабораторної роботи №5**

- ✓ Розв'яжемо задачу на використання команд категорії «Математика».

**Приклад 5.1.** Написати програму обчислення значення виразу  $x = \frac{\ln|a+b|}{a-b}$  ( $a+b \neq 0, a-b \neq 0$ ) в припущенні, що параметри  $a$  і  $b$  вводяться користувачем так, що умови, зазначені в дужках, виконуються.

► Реалізацію програми подано на рис. 5.9. ◀

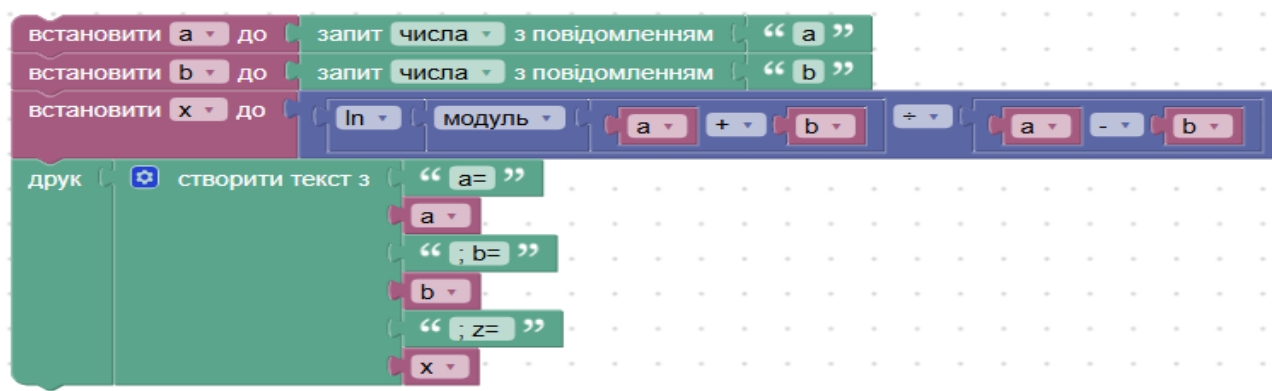
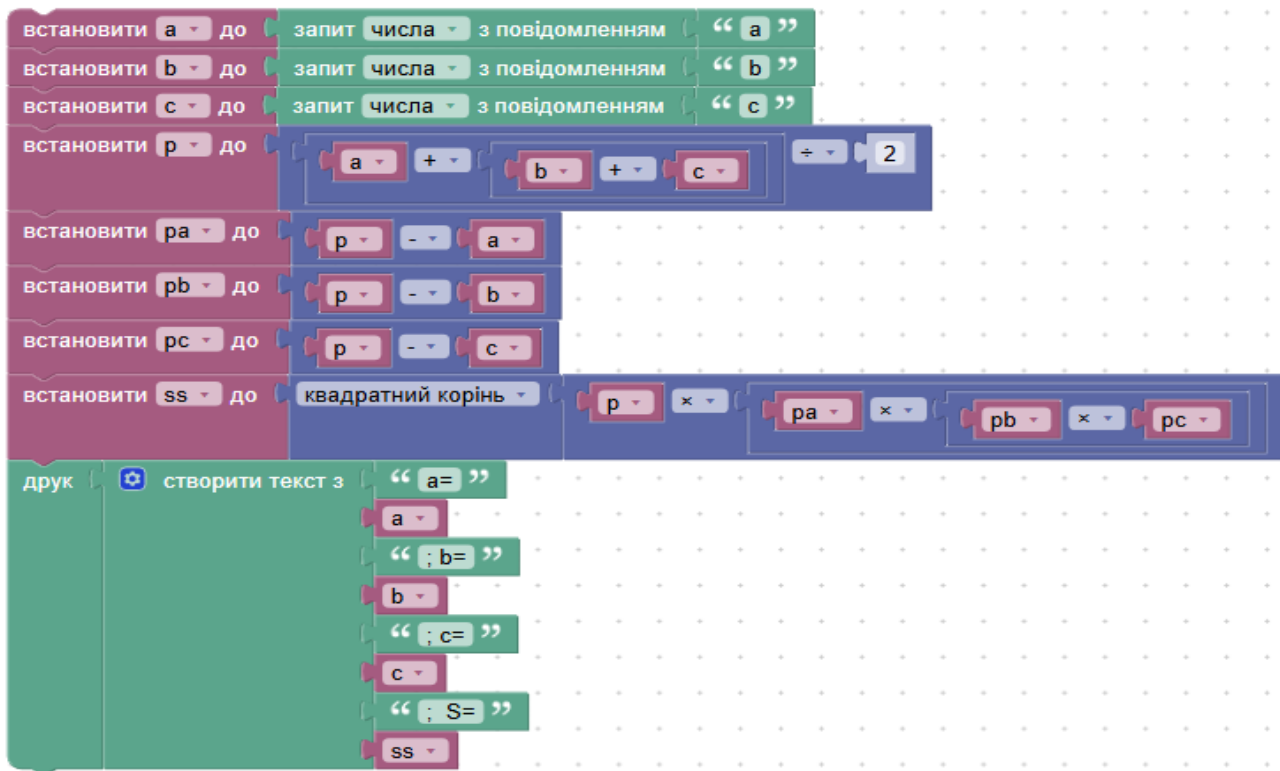


Рис. 5.9 – Програма з використанням команд категорії «Математика»

**Приклад 5.2.** Написати програму обчислення площі трикутника за формулою Герона. Вважаємо, що користувач вводить значення довжин сторін  $a, b, c$ , які дійсно утворюють трикутник.

► Програму наведено на рис. 5.10. ◀



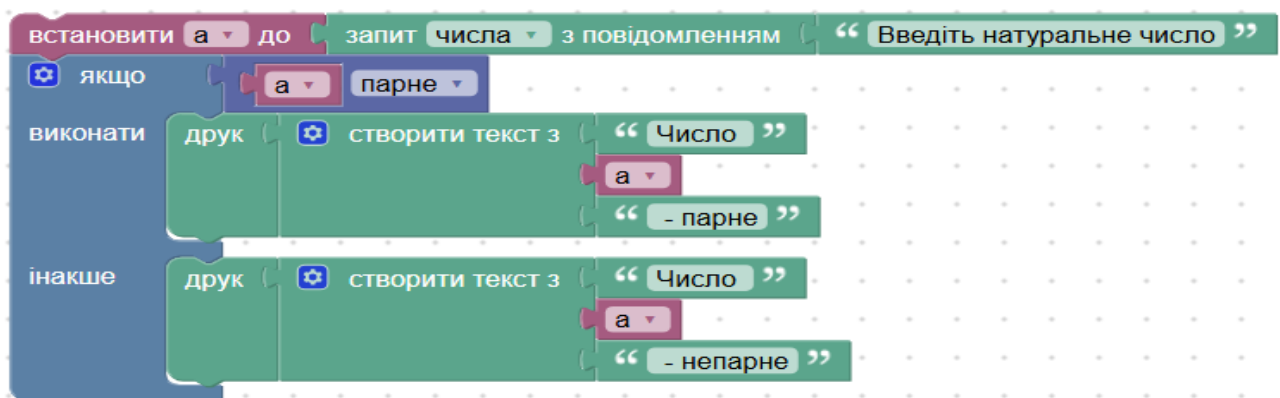
Результат: `a=3; b=4; c=5; S=6`

Рис. 5.10 – Програма обчислення площі трикутника за формулою Герона

✓ Розглянемо використання *розгалужених структур*.

**Приклад 5.3.** Написати програму перевірки на парність введенного користувачем натурального числа.

► Результат виконання поставленого завдання наведено на рис. 5.11. ◀



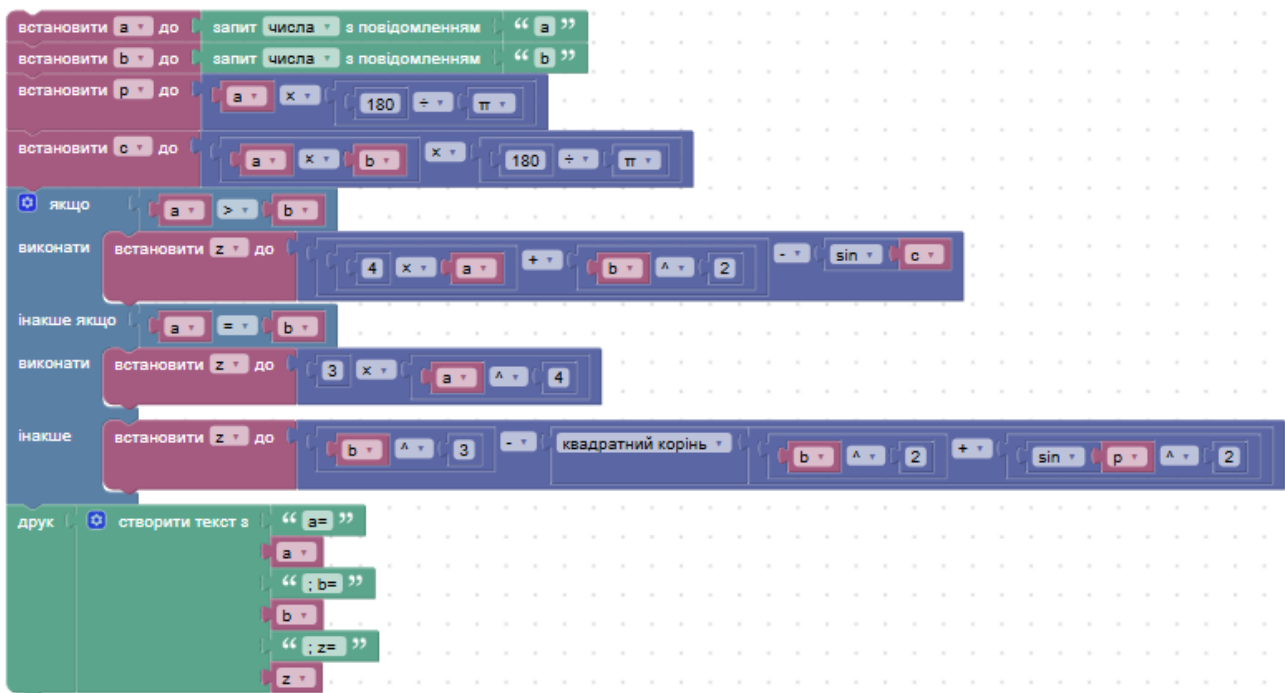
Результат: 1) `Число 20 - парне` 2) `Число 15 - непарне`

Рис. 5.11 – Програма розгалуженої структури для визначення парності числа

**Приклад 5.3.** Написати програму, яка дозволяє обчислити значення виразу:

$$z = \begin{cases} 4a + b^2 - \sin ab, & a > b; \\ 3a^4, & a = b; \\ b^3 - \sqrt{b^2 + \sin^2 a}, & a < b. \end{cases}$$

► Результат виконання завдання наведено на рис. 5.12. Звертаємо увагу, що аргументи тригонометричних функцій мають вимірюватися в градусах, а значення параметрів  $a$  і  $b$  визначаються в радіанах. Зазначене вимагає попереднього переведення  $a$  і  $a \cdot b$  до градусної міри, що реалізовано в програмі через обчислення параметрів  $p = a \cdot \frac{180}{\pi}$  і  $c = a \cdot b \cdot \frac{180}{\pi}$ . ◀



Результат: 1)  $a=1; b=0.5; z= 3.770574461395797$  2)  $a=2; b=2; z= 48$   
 3)  $a=1; b=2; z= 5.830190464977727$

Рис. 5.12 – Програма розгалуженої структури для обчислення значення функції з різним заданням на різних множинах

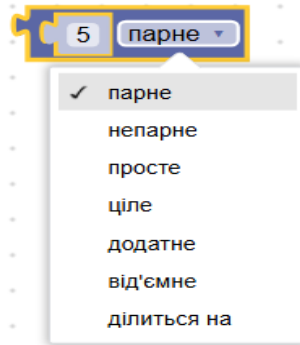
**Завдання 5.1.** Реалізувати приклади 5.1–5.3 в середовищі Google Blockly.

### Контрольні питання до лабораторної роботи №5

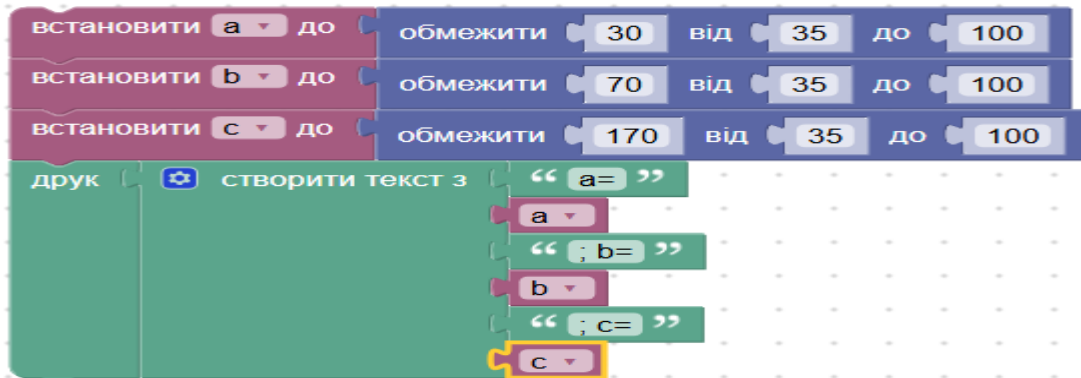
1. На які типи розбито блоки в середовищі Google Blockly?
2. Як створити змінну в середовищі Google Blockly?
3. Як присвоїти значення змінним в середовищі Google Blockly?
4. Які особливості математичних операторів у Google Blockly Ви знаєте?
5. Які математичні функції реалізовано в середовищі Google Blockly?
6. Поясніть, яка дія виконується за допомогою блока:



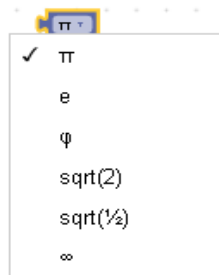
7. Поясніть, які дії виконуються за допомогою блока:



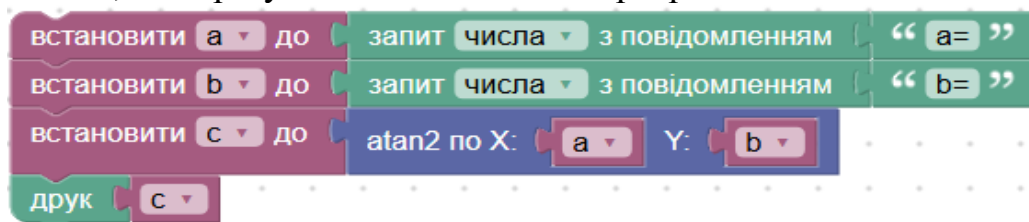
8. Поясніть, що є результатом виконання програми:



9. Поясніть значення кожної з констант:



10. Поясніть, що є результатом виконання програми:



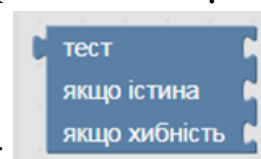
11. Які значення може приймати логічна змінна?

12. Які операції порівняння Ви знаєте?

13. Наведіть приклади операторів (блоків), результатом виконання яких є істина або хибність.

14. Які оператори (блоки) реалізують логічні І, НЕ та АБО?

15. В яких випадках використовують блок ?



16. В яких випадках використовують блок ?

## Лабораторна робота №6

## Тема: Цикли. Масиви

**Мета:** Навчитись створювати циклічні програми у візуальному середовищі розробки Google Blockly. Ознайомитись з можливостями створення масивів різних типів даних та роботою з ними (звернення до елементів, друк елементів та ін.).

## Теоретичні відомості

- ✓ Категорія «Петлі» містить чотири види циклів (рис. 6.1):
- з відомою кількістю повторень;
  - цикл з передумовою;
  - цикл за змінною;
  - спеціальний цикл за змінною для роботи зі списками.

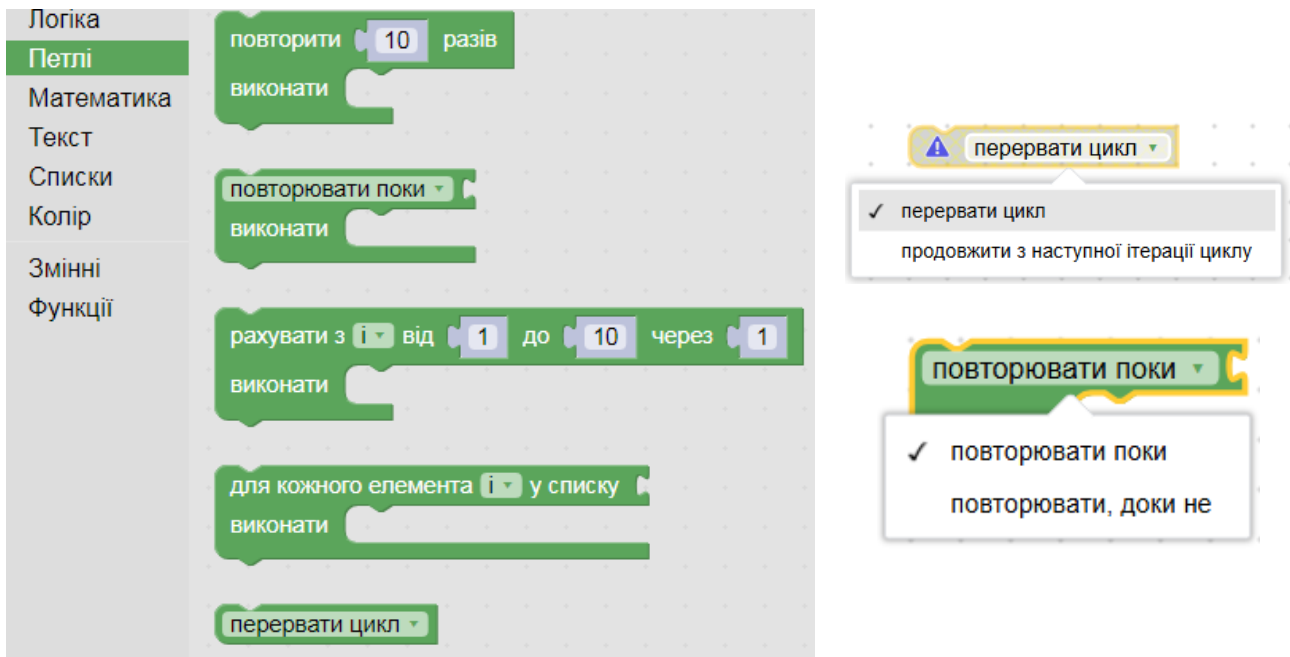


Рис. 6.1 – Команди категорії «Петлі»

Якщо виникає необхідність роботи з циклом з післяумовою, слід зробити нескінченний цикл з умовою виходу з нього, наприклад, так, як показано на рис. 6.2.

Як і в деяких інших мовах програмування, ітератор циклу має бути цілим числом.

Цикли можна включати один в інший так само, як це відбувалося в Scratch (рис. 3.5).

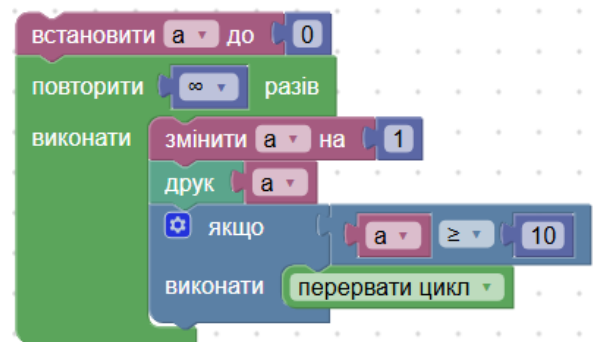


Рис. 6.2 – Цикл з нескінченною кількістю повторень

✓ Категорія «Списки» (рис. 6.3).

Google Blockly дозволяє створювати списки з будь-яких однорідних елементів, в тому числі з чисел, символів та кольорів. Потрібно лише привласнити змінній значення списку (порожнього або попередньо заповненого елементами одного типу).

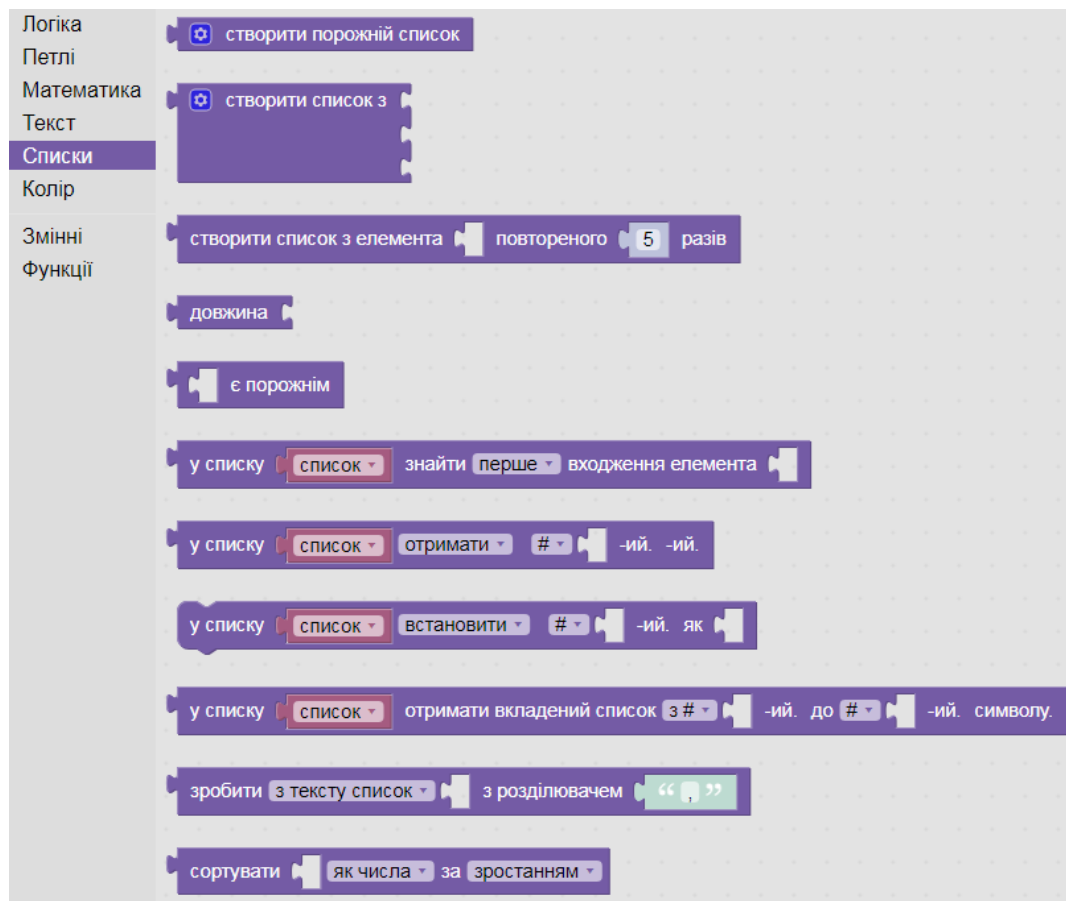
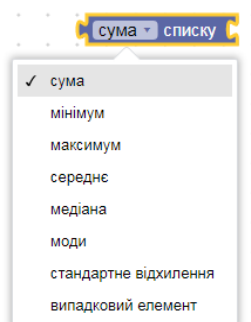


Рис. 6.3 – Команди категорії «Списки»

Надалі робота зі списками мало чим відрізняється від роботи з масивами у будь-якій іншій мові програмування.

Каталог «Списки» надає можливість в одну дію дізнатися чи є список порожнім, яку він має довжину, знайти входження певного елемента чи виділити підсписок. Можна звертатися до елементів за індексом.

Для роботи зі списками (масивами) існують вбудовані алгоритми, наприклад, можна знайти суму елементів числового масиву, найменший елемент тощо. Відповідний блок можна знайти в категорії «Математика».



**Приклад 6.1.** Написати програму знаходження найбільшого елементу в списку випадкових цілих чисел з визначенням місця його розміщення у списку.

► Реалізацію програми наведено на рис. 6.4.

Спочатку здійснюється ініціалізація масиву (списку

встановити **a** до створити порожній список

Потім елементам масиву присвоюють значення за допомогою оператора

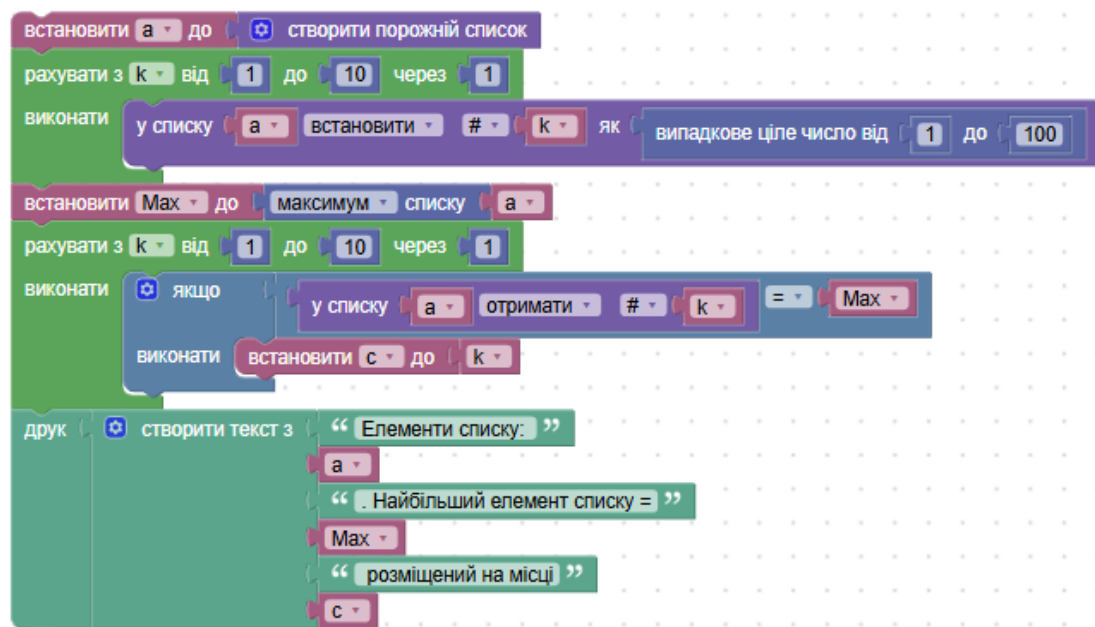
у списку **a** встановити # як

. Список заповнюється випадковими числами в циклі за допомогою блоку випадкове ціле число від **1** до **100** категорії «Математика».

Найбільший елемент списку знаходимо за допомогою команди

максимум списку **a** тієї ж категорії.

Структура розгалуження, вкладена в цикл, дозволяє визначити номер елемента в списку з найбільшим значенням. ◀



**Результат:** Елементи списку: 70,38,4,53,61,52,99,56,3,99. Найбільший елемент списку =99 розміщений на місці10

Рис. 6.4 – Програма пошуку найбільшого елемента списку випадкових чисел з визначенням місця його розміщення

**Завдання 6.1.** Реалізувати програму з прикладу 6.1.

### Завдання до лабораторної роботи №6

Ознайомитися з командами категорій «Петлі» і «Списки». Виконати завдання свого варіанту.

#### Варіант 1

- Відома кількість курсів на спеціальності і кількість студентів на кожному з них. Не використовуючи масив визначити середню кількість студентів на курсі.

2. Вивести на друк перші  $n$  простих чисел ( $n$  задається з клавіатури).
3. Створити масив з 20 елементів, на парних місцях в якому стоять одиниці, а на непарних місцях – числа, що дорівнюють залишку від ділення їхнього номера на 5.

### Варіант 2

1. Дано два натуральних числа  $a$  і  $b$ . Знайти суму всіх кратних 3 чисел, які належать проміжку  $[a; b]$ .
2. Написати програму для розв'язку наступної задачі: сумувати числа, що вводяться з клавіатури, доки не зустрінеться нуль.
3. Створити масив з 20 випадкових двозначних чисел. Всі парні числа замінити на 2.

### Варіант 3

1. Відомі два натуральних числа  $x$  та  $y$  ( $x < y$ ). Вивести на друк суму простих чисел в діапазоні від  $x$  до  $y$ .
2. Знайти найбільше число, факторіал якого є 4-значним числом.
3. Створити масив з 20 випадкових двозначних чисел. Знайти суму елементів, що стоять на парних місцях та перебільшують останній елемент масиву (додатково вивести ці елементи на друк).

### Варіант 4

1. Відома кількість курсів на спеціальності і кількість студентів на кожному з них. Не використовуючи масив визначити курси, на яких найбільша й найменша кількість студентів.
2. Сумувати числа, що вводяться з клавіатури, доки не зустрінуться два однакових числа поспіль.
3. Створити масив з 20 випадкових двозначних чисел. З'ясувати, яких чисел в цьому масиві більше – які діляться на 5, або які діляться на 3.

### Варіант 5

1. З клавіатури вводиться кількість студентів, а потім їх оцінки з математичного аналізу напередодні екзаменів. Не використовуючи масивів розрахуйте кількість студентів, які не отримали допуск (мають менше за 60 балів).
2. В перший день спортсмен пробіг 10 км, кожного наступного дня він збільшував відстань на 10%. На який день спортсмен пробіжить  $n$  км? ( $n$  задається з клавіатури).
3. Створити масив з 20 випадкових двозначних чисел. Знайдіть кількість різних елементів цього масиву.

### Варіант 6

1. Використовуючи дві допоміжні змінні, але не використовуючи масив, знайти  $n$ -е число послідовності Фібоначчі.
2. В перший День Народження дідусь подарив хлопчику 1 долар. Кожного наступного року він подвоював свій подарунок та додавав до нього кількість

років. В якому віці хлопчик отримає  $n$  чи більше доларів ( $n$  задається з клавіатури).

- Створити масив з 20 випадкових двозначних чисел. Знайдіть 2 сусідніх елементи масиву, сума яких мінімальна.

### Варіант 7

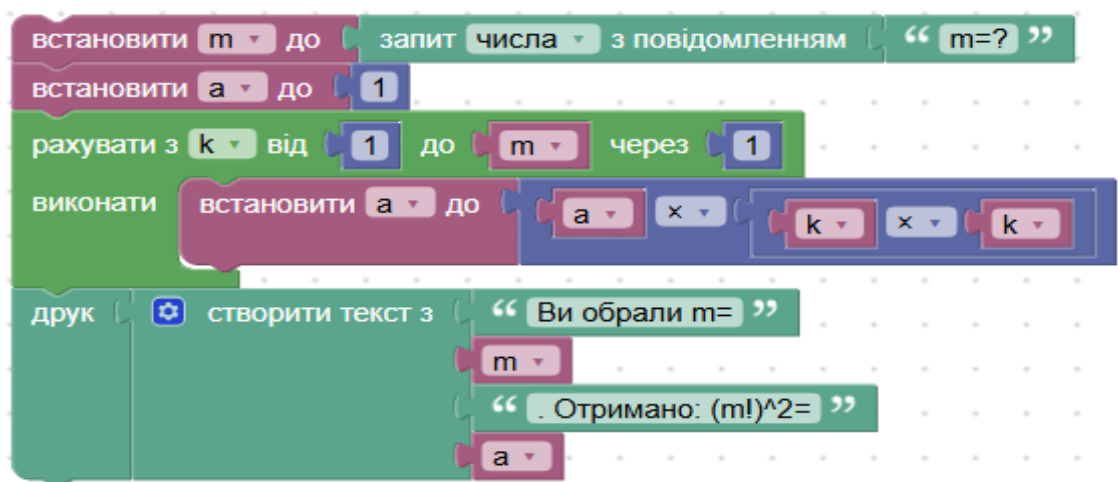
- Знайдіть суму квадратів перших  $n$  чисел.
- Комп'ютер загадує число від 1 до 100. Користувач намагається вгадати це число. Після кожної невдалої спроби комп'ютер повідомляє, менше або більше задумане число.
- Створити масив з 20 випадкових двозначних чисел. Серед елементів з непарними номерами знайдіть найбільший елемент масиву, який ділиться на 3.

### Методичні рекомендації до лабораторної роботи №6

✓ Розглянемо задачі на застосування *циклічних структур*, що не передбачають використання масивів (списків).

**Приклад 6.2.** Написати програму обчислення квадрата факторіала  $(m!)^2$ , в якій  $m$  вводить користувач.

► Програма, реалізована на рис. 6.5, використовує *циклічну структуру* з відомою кількістю повторень. ◀



**Результат:** Ви обрали m=3. Отримано:  $(m!)^2=36$

Рис. 6.5 – Програма обчислення квадрата факторіала натурального числа

**Приклад 6.3.** Написати програму, яка дозволяє без використання масиву знайти, якою була найменша оцінка в класі за контрольну з інформатики. Кількість учнів та їхні оцінки вводяться з клавіатури. Система оцінювання – 12-бальна.

► Для реалізації програми (рис. 6.6) використовуються дві змінні, з іменами  $a$  та  $Min$ . Спочатку змінній  $Min$  присвоюється найбільше можливе

значення (у нашому випадку – 12), яке змінюється, якщо серед чисел  $a$ , що вводяться, з'являється менше за  $Min$ . ◀

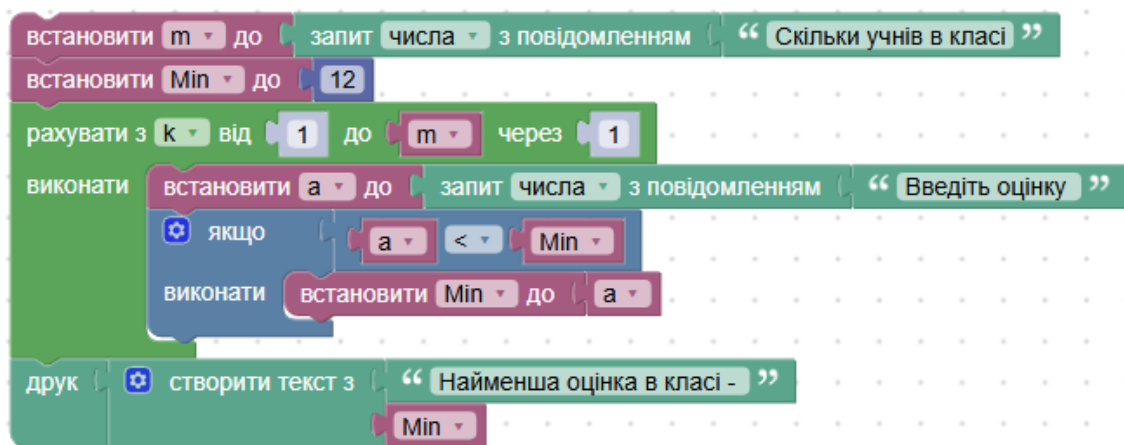


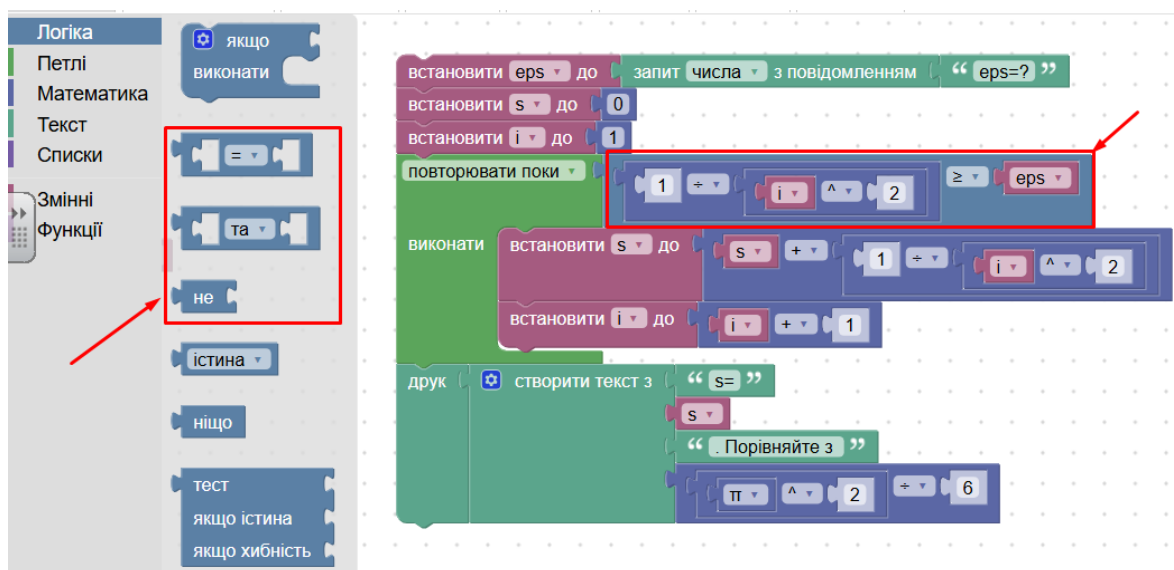
Рис. 6.6 – Пошук найменшої оцінки за контрольну роботу без використання масивів

В *циклах з передумовою* слід задавати умову за допомогою логічних блоків (див. рис. 6.7).

**Приклад 6.4.** Написати програму наближеного обчислення суми  $S = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{n^2} + \dots$ . Умовою завершення підсумовування обрати виконання нерівності  $\frac{1}{n^2} < \varepsilon$ , де  $\varepsilon$  – достатньо мале дійсне додатне число.

Порівняти результат з точним значення суми ряду  $\frac{\pi^2}{6}$ .

► На рис. 6.7 наведено код програми з використанням передумови. При обчисленні обрано  $\varepsilon = 0.0001$ . Проте, результат показує, що отримане наближене значення збігається з точним лише на двох значущих знаках. Це пов'язано з тим, що розглянутий ряд є повільно збіжним. ◀



**Результат:**

$s = 1.6349839001848923$ . Порівняйте з  $1.6449340668482264$

Рис. 6.7 – Програма наближеного обчислення суми з використанням циклу з передумовою

**Приклад 6.5.** Написати програму «Вгадай число».

► Передбачимо, що буде загадано число від 1 до 10. Залежно від уведеного користувачем числа виводяться повідомлення, допоки користувач не вгадає. Реалізується програма за допомогою вкладених поператорів передумови «повторювати поки» та повного розгалуження «якщо...інакше» (рис. 6.8). Замість використаного оператора розгалуження можна застосувати й інший. ◀

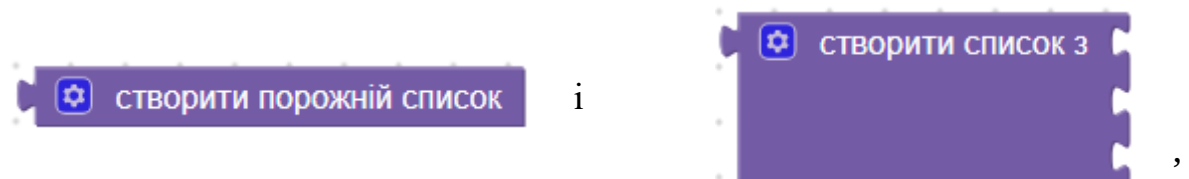


Рис. 6.8 – Програма реалізації гри «Вгадай число!»

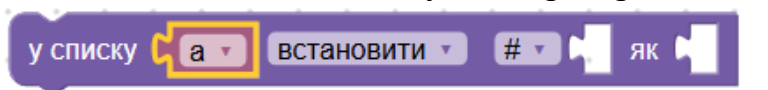
✓ Розглянемо *задачі зі списками (масивами)*.

**Приклад 6.6.** Написати програму, яка дозволяє створити масив з 5-ти перших і 5-ти останніх елементів масиву а.

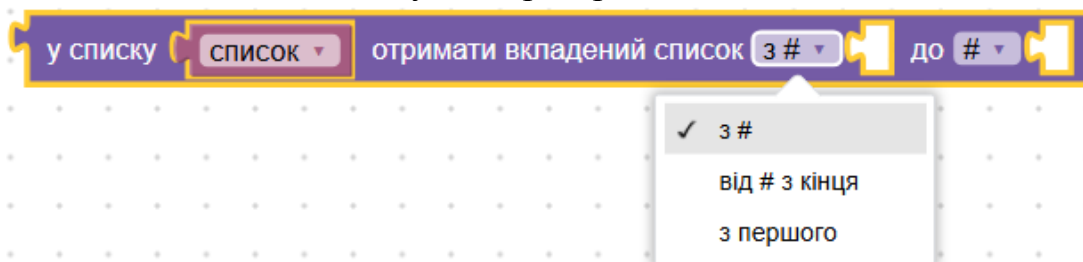
► Для створення списків використовуються блоки



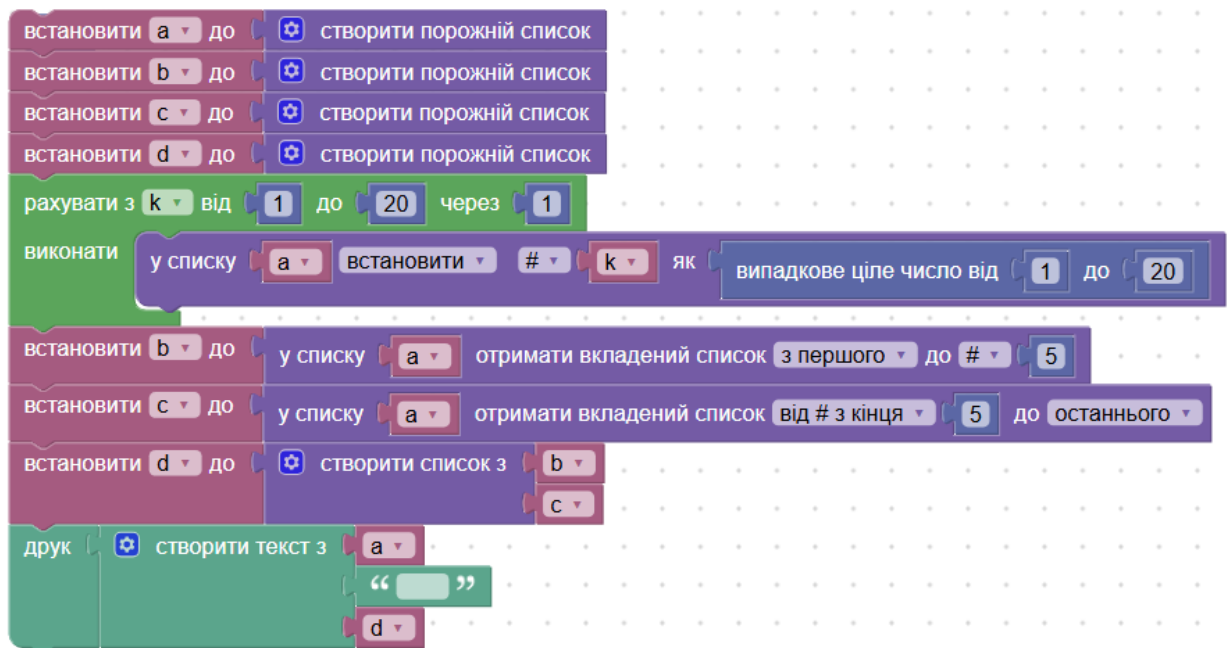
Для присвоєння значень елементам масиву – оператор



а для виділення частини списку – оператор



Реалізацію програми наведено на рис. 6.9. ◀

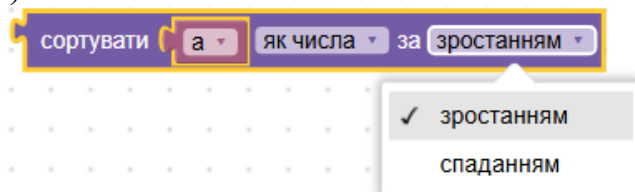


**Результат:** 4,97,22,68,52,5,61,68,41,20,49,84,85,98,56,37,61,78,79,25  
4,97,22,68,52,37,61,78,79,25

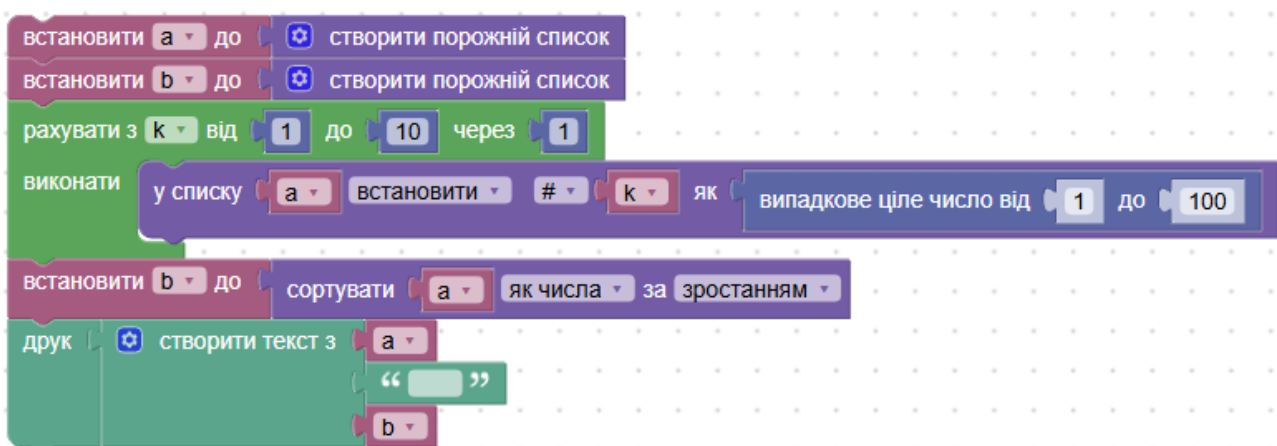
Рис. 6.9 – Програма створення масиву, що складається з частин іншого масиву

**Приклад 6.7.** Відсортувати елементи масиву за зростаннями.

► Застосуємо вбудований в Google Blockly функціонал сортування списків (масивів)



Приклад реалізації програми наведено на рис. 6.10 ◀



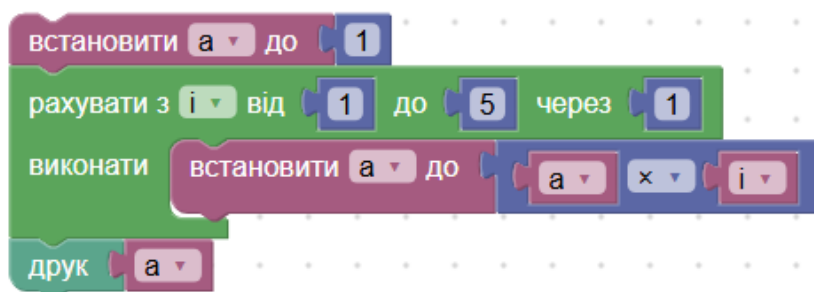
**Результат:** 35,85,75,1,83,60,65,58,35,71 1,35,35,58,60,65,71,75,83,85

Рис. 6.10 – Програма сортування масиву з використанням вбудованого функціоналу Google Blockly

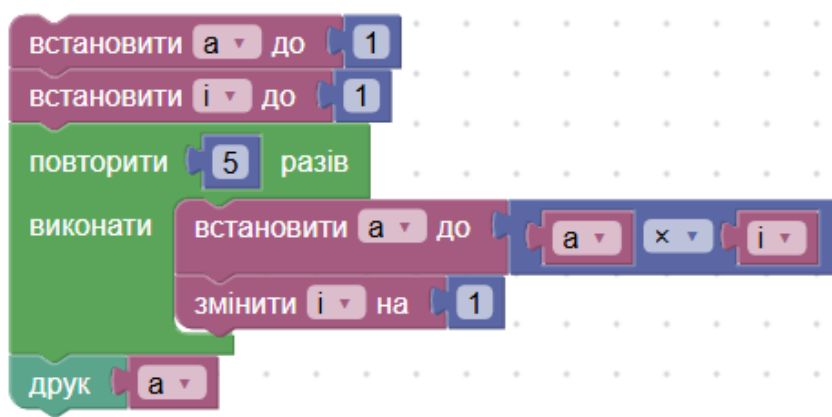
**Завдання 6.2.** Реалізувати програми з прикладів 6.2–6.7 в середовищі Google Blockly.

## Контрольні питання до лабораторної роботи №6

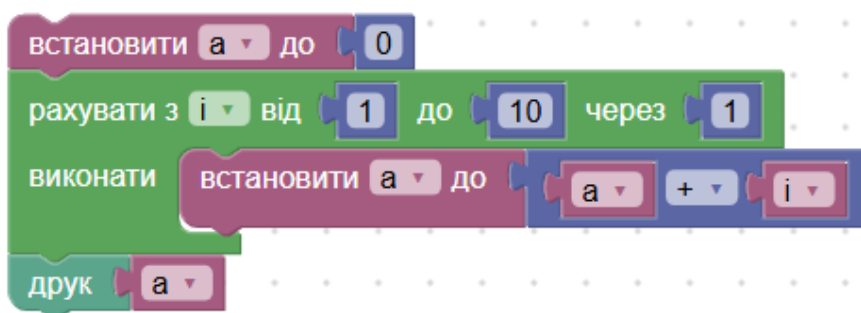
1. Що таке цикли? Які види циклів ви знаєте?
2. Що таке нескінченний цикл? Які Ви знаєте способи зупинки його виконання?
3. Поясніть частину програми. Що є результатом її виконання?



4. Поясніть частину програми. Що є результатом її виконання? Порівняйте з попереднім питанням.



5. Поясніть частину програми. Що є результатом її виконання?



6. Які особливості роботи зі списками у Google Blockly Ви знаєте?
7. Як можна поєднати два списки в один у Google Blockly?

## Лабораторна робота №7

### Тема: Робота з текстом. Створення підпрограм

**Мета:** Опанувати навички роботи з текстами та символічними змінними у Google Blockly. Навчитись використовувати підпрограми та функції при створенні програм.

### Теоретичні відомості

✓ Категорія *Текст*» (рис. 7.1).

Google Blockly надає можливості для створення програм, в яких необхідно використовувати текстові змінні.

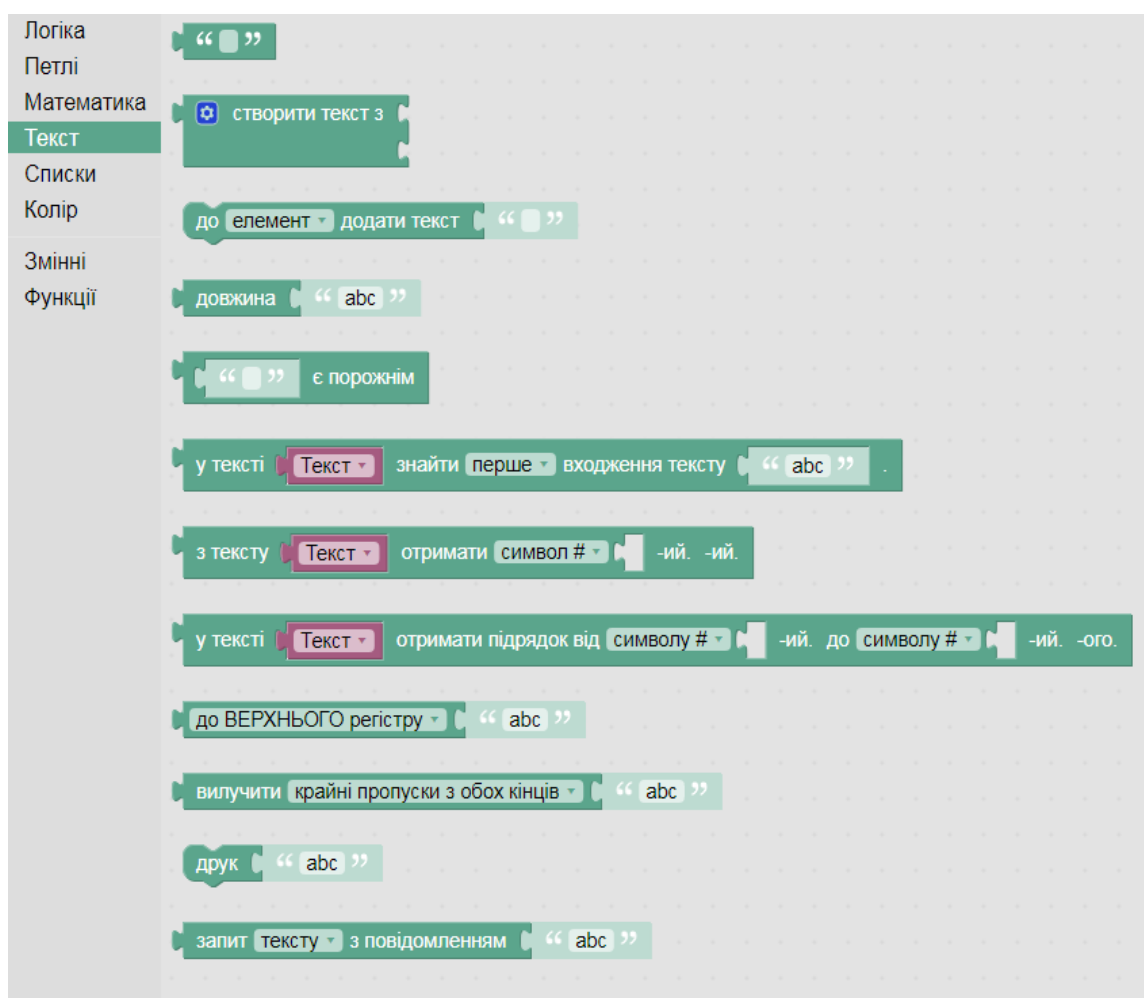


Рис. 7.1 – Команди категорії «Текст»

Ми вже використовували декілька команд з цього каталогу для друку повідомлень, але можливості Google Blockly більші. В одну дію ми можемо, наприклад, змінити регістр повідомлення або вилучити пропуски.

Звичайно ж, програми для роботи з текстом виконують задачі набагато складніші. Google Blockly надає можливість лише звернутись до певного знаку в рядку, але не змінити його, тому для збереження результатів доводиться

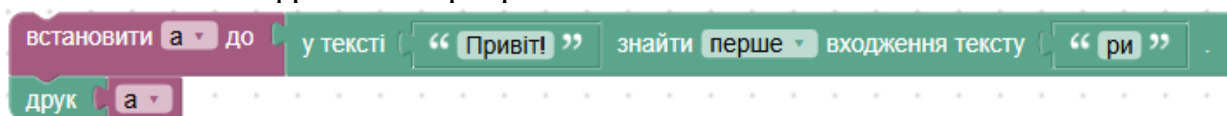
ініціалізувати новий рядок. Можна також шукати входження одного рядка до іншого, вибрати n-й елемент або цілий підрядок.

В рис. 7.2 наведено способи роботи з текстом.

| <b>Створення текстових змінних</b>                  |                                                      |
|-----------------------------------------------------|------------------------------------------------------|
| 1 спосіб:                                           |                                                      |
| 2 спосіб:                                           |                                                      |
| 3 спосіб:                                           |                                                      |
| 4 спосіб:                                           |                                                      |
| <b>Способи друку текстової змінної та тексту</b>    | <b>Знаходження номера позиції підтексту у тексті</b> |
|                                                     |                                                      |
|                                                     |                                                      |
| <b>Робота з окремими літерами та групами літер.</b> |                                                      |
|                                                     |                                                      |
| <b>Знаходження довжини</b>                          |                                                      |
|                                                     |                                                      |
| <b>Прибирання зайвих пропусків</b>                  | <b>Вибір літери</b>                                  |
|                                                     |                                                      |
|                                                     |                                                      |
|                                                     | <b>Вибір частини тексту:</b>                         |

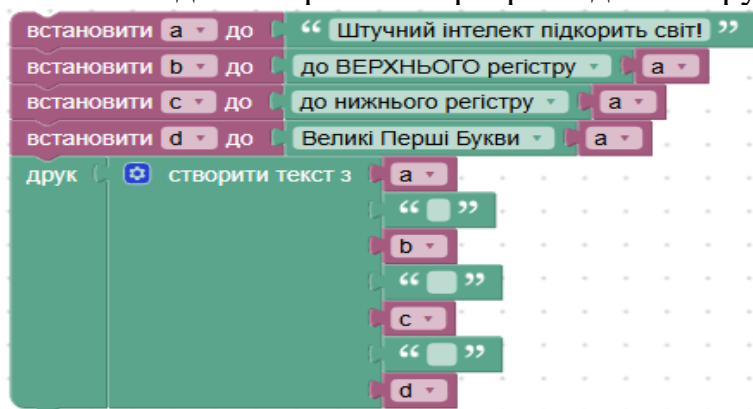
Рис. 7.2 – Функціонал деяких команд категорії «Текст»

Розглянемо фрагмент програми



Результатом її роботи буде «2». При використанні аналогічної команди пошуку підтексту в тексті у випадку, якщо текст не знайдено, отримаємо за результаті – «0».

Наведена на рис. 7.3 програма демонструє дії зі зміною регістру.



### Результат:

Штучний інтелект підкорить світ! ШТУЧНИЙ ІНТЕЛЕКТ ПІДКОРИТЬ СВІТ! штучний інтелект підкорить світ! Штучний Інтелект Підкорить Світ!

Рис. 7.3 – Програма зміни регістру в тексті

✓ Категорія «Функції» дозволяє працювати з процедурами (рис. 7.4).

Функції добре відомі вам з інших мов програмування, але нагадаємо, що їх необхідно використовувати для таких дій, що повторюються багато разів. Ідеальна програма в головному тілі має лише декілька викликів функцій, а вони вже в свою чергу знов розбиті на підфункції і так далі до найпростіших цілей.

|                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Логіка</p> <p>Петлі</p> <p>Математика</p> <p>Текст</p> <p>Списки</p> <p>Колір</p> <p>Змінні</p> <p><b>Функції</b></p> <p>до щось зробити</p> <p>блок тексту</p> <p>до щось зробити</p> <p>блок тексту</p> <p>повернути</p> <p>якщо повернути</p> | <p>до щось зробити</p> <p>Створює функцію без виводу.</p> <p>до щось зробити2</p> <p>повернути</p>                                                                                                                   |
| Команди категорії «Функції»                                                                                                                                                                                                                         | Виклик процедур та функцій                                                                                                                                                                                           |
| <p>Блоки для створення процедур та функцій</p> <p>Звернення до процедур та функцій</p>                                                                                                                                                              |                                                                                                                                                                                                                      |
| <p>до щось зробити</p> <p>назва входу: <b>x</b></p> <p>Додати до функції вхідні параметри.</p>                                                                                                                                                      | <p>до щось зробити з: <b>y, x</b></p> <p>назва входу: <b>z</b></p> <p>входи</p> <p>назва входу: <b>y</b></p> <p>назва входу: <b>x</b></p>                                                                            |
| <p>до щось зробити2</p> <p>повернути</p> <p>назва входу: <b>x</b></p> <p>входи</p> <p>дозволити дії <input checked="" type="checkbox"/></p>                                                                                                         | <p>до щось зробити2 з: <b>x, y</b></p> <p>повернути</p> <p>назва входу: <b>z</b></p> <p>входи</p> <p>назва входу: <b>x</b></p> <p>назва входу: <b>y</b></p> <p>дозволити дії <input checked="" type="checkbox"/></p> |
| <p>щось зробити</p> <p>Запустити користувацьку функцію "щось зробити".</p> <p>щось зробити з:</p> <p><b>y</b></p> <p><b>x</b></p>                                                                                                                   | <p>щось зробити</p> <p>Запустити користувацьку функцію "щось зробити" і використати її вивід.</p> <p>щось зробити з:</p> <p><b>x</b></p> <p><b>y</b></p>                                                             |

Рис. 7.4 – Команди категорії «Функції та робота з ними

В Google Blockly окремо подаються функції (повертають певне значення) та процедури (не повертають значення). Щоб внести вхідні змінні, натисніть на зірочку, перетягніть необхідну кількість блоків і дайте імена локальним змінним. Функція вміє повертати 1 значення.

### **Завдання до лабораторної роботи №7**

Ознайомитися з командами категорій «Текст» та «Функції». . Виконати завдання свого варіанту.

#### **Варіант 1**

1. Обрати відповідний текст. Замінити на знак зірочка (\*) в тексті слово, введене з клавіатури.
2. Знайти значення функції  $y = \sin x$ , розклавши її в ряд Маклорена до 20 доданка. Функція при знаходженні розкладу в ряд Маклорена повинна використовувати створені вами функції піднесення до степеня, визначення знака доданка та обчислення факторіала.

#### **Варіант 2**

1. Обрати відповідний текст. Видалить в тексті слова, що повторюються. (Якщо між ними немає розділового знаку).
2. Знайти значення функції  $y = \cos x$ , розклавши її в ряд Маклорена до 20 доданка. Функція при знаходженні розкладу в ряд Маклорена повинна використовувати створені вами функції піднесення до степеня, визначення знака доданка та обчислення факторіала.

#### **Варіант 3**

1. Обрати відповідний текст. Замініть на знак серця всі букви о/О в тексті.
2. Знайти значення функції  $y = \ln(1 + x)$ , розклавши її в ряд Маклорена до 20 доданка. Функція при знаходженні розкладу в ряд Маклорена повинна використовувати створені вами функції піднесення до степеня та визначення знака доданка.

#### **Варіант 4**

1. Обрати відповідний текст. Видалити зайві пропуски в тексті.
2. Знайти значення функції  $y = \sqrt{1 + x}$ , розклавши її в ряд Маклорена до 20 доданка. Функція при знаходженні розкладу в ряд Маклорена повинна використовувати створені вами функції піднесення до степеня, визначення знака доданка та обчислення факторіала.

#### **Варіант 5**

1. Обрати відповідний текст. Порахуйте кількість розділових знаків в тексті.
2. Знайти значення функції  $y = \frac{1}{1 + x}$ , розклавши її в ряд Маклорена до 20 доданка. Функція при знаходженні розкладу в ряд Маклорена повинна

використовувати створені вами функції піднесення до степеня та визначення знака доданка.

### Варіант 6

1. Обрати відповідний текст. Зробіть всі символи між лапками великими. («Гарі Потер» — «ГАРІ ПОТЕР»).
2. Знайти значення функції  $y = sh x$ , розклавши її в ряд Маклорена до 20 доданка. Функція при знаходженні розкладу в ряд Маклорена повинна використовувати створені вами функції піднесення до степеня, визначення знака доданка та обчислення факторіала.

### Варіант 7

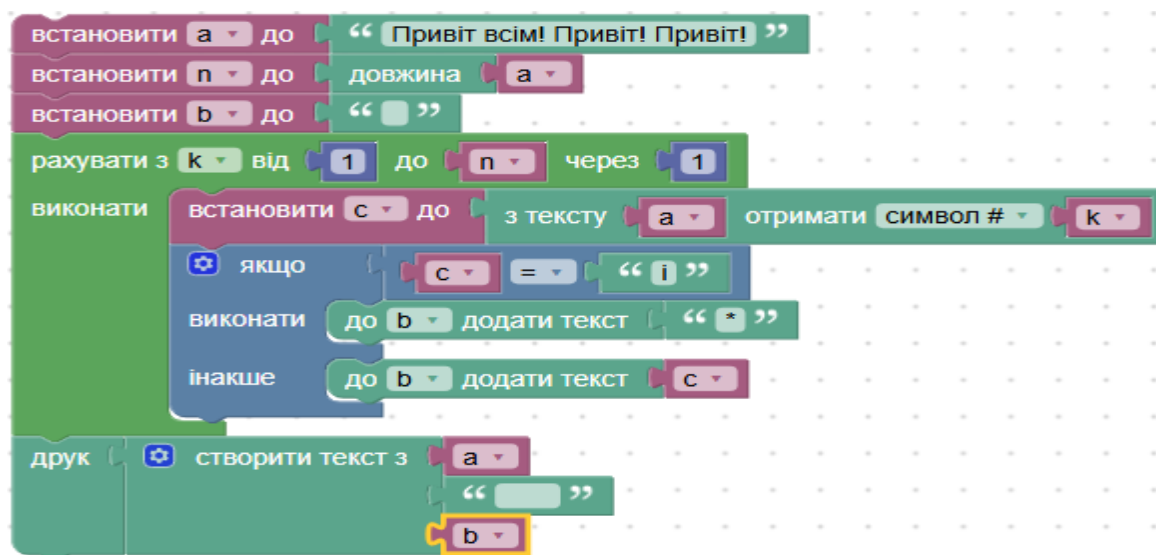
1. Обрати відповідний текст. Перевірити, чи є фраза паліндромом.
2. Знайти значення функції  $y = arctg x$ , розклавши її в ряд Маклорена, до 20 доданка. Функція при знаходженні розкладу в ряд Маклорена повинна використовувати створені вами функції піднесення до степеня та визначення знака доданка.

## Методичні рекомендації до лабораторної роботи №7

- ✓ Розглянемо задачі з *обробки текстів*.

**Приклад 7.1.** Написати програму, яка замінює символ «і» в тексті на «\*».

► На рис. 7.5 наведено код відповідної програми. Вона використовує як функціонал команд категорії «Текст», так і циклічні й розгалужені структури категорій «Петлі» та «Логіка».



**Результат:**

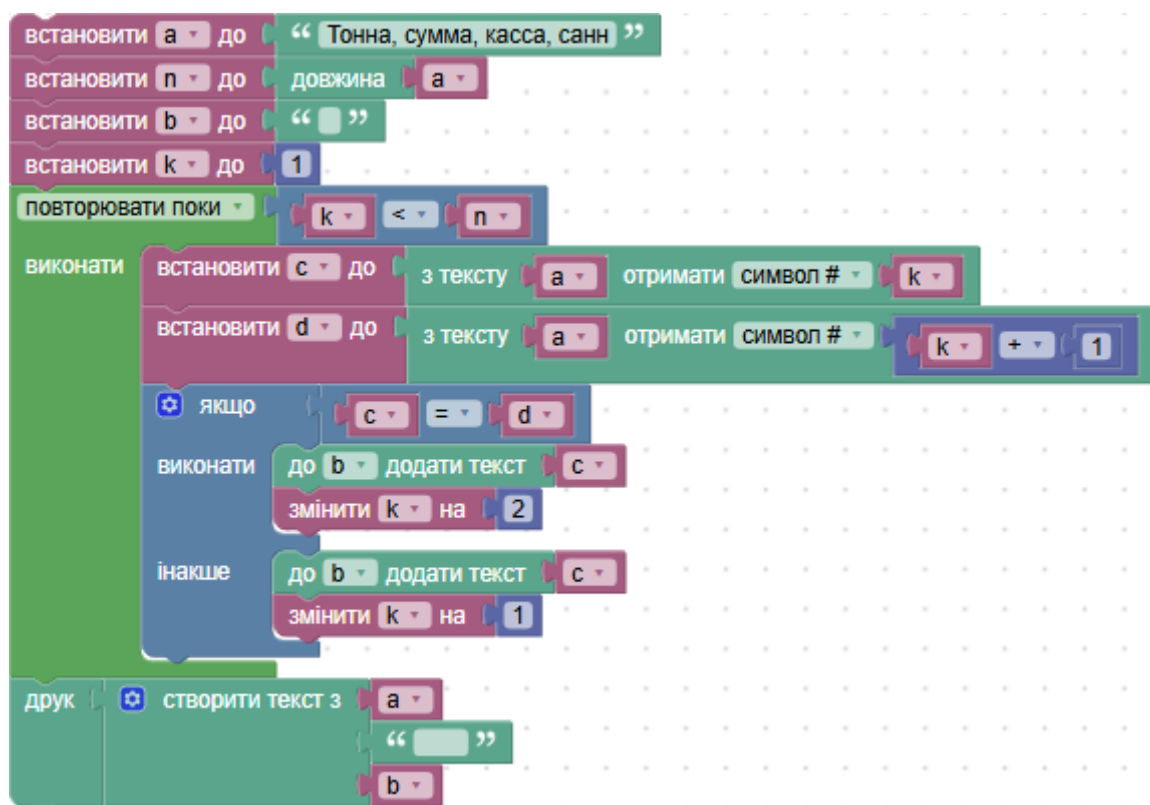
Привіт всім! Привіт! Привіт!    Прив\*т вс\*м! Прив\*т! Прив\*т!

Рис. 7.5 – Програма заміни символу «і» символом «\*» в тексті

**Приклад 7.2.** Написати програму, яка реалізує заміну подвоєних букв на одну.

► Програма на рис. 7.6 демонструє розв’язання поставленої задачі. В середині циклу з відомою кількістю повторень не рекомендовано змінювати

значення параметру циклу. Саме тому в цій програмі використано цикл з передумовою. ◀

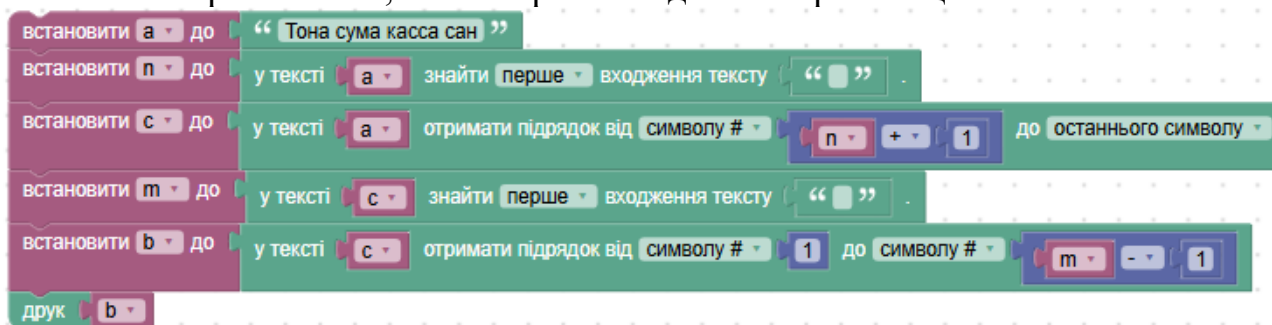


**Результат:** Тонна, сума, каса, санн    Тона, сума, каса, сан

Рис. 7.6 – Програма заміни подвоєних букв на одну

**Приклад 7.3.** Програмною реалізацією виконати завдання: знайти друге слово в тексті, якщо слова в тексті відокремлені пропусками.

► Рис. 7.7 демонструє результат виконання завдання. Зверніть увагу на блоки категорії «Текст», які використано для його реалізації! ◀

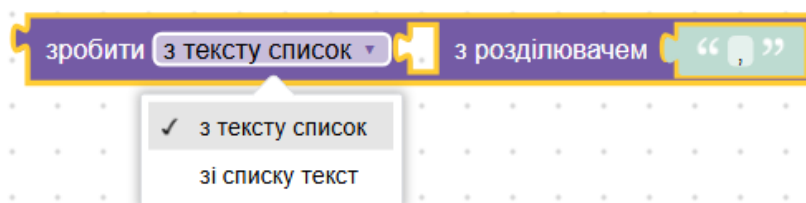


**Результат:** сума

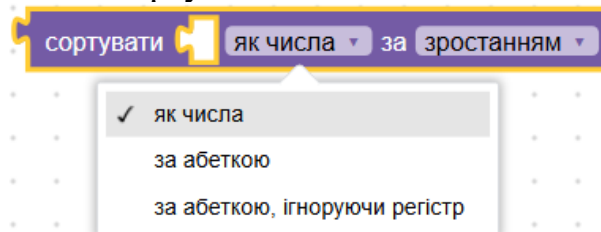
Рис. 7.7 – Програма пошуку другого слова в тексті (слова в тексті відокремлені пробілами)

**Приклад 7.4.** Програмно реалізувати сортування слів тексту за абеткою. Порівняти результати різних видів сортування.

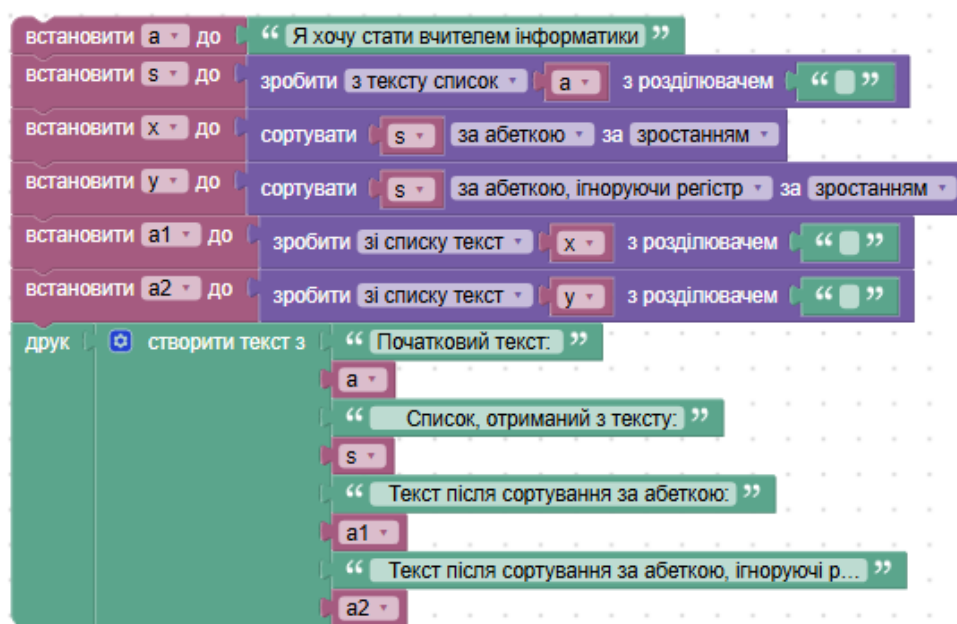
► Наведена на рис. 7.8 програма використовує спочатку перетворення тексту в список, а по завершенню роботи програми робить зворотне перетворення за допомогою блока категорії «Списки»:



Отриманий список сортується за допомогою блока тієї ж категорії:



На рис. 7.8 також наведено результат роботи програми. Зверніть увагу, що наразі українську абетку не закладено у середовище Google Blockly. ◀



**Результат:**

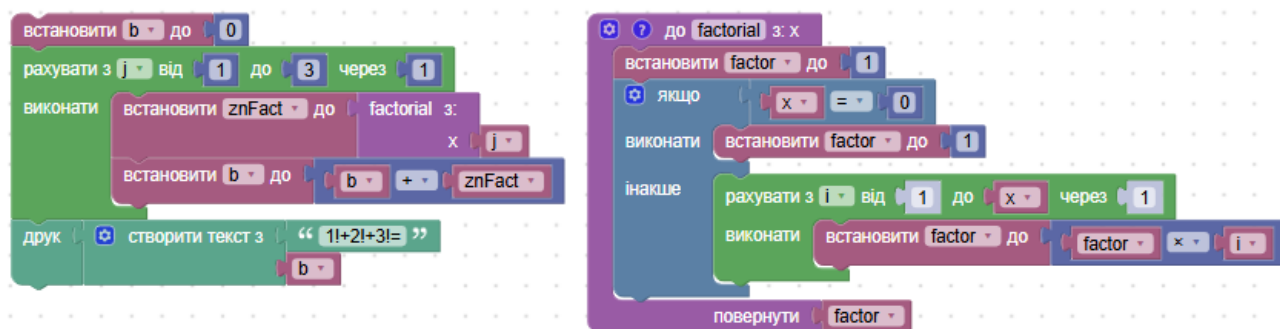
Початковий текст: Я хочу стати вчителем інформатики    Список, отриманий з тексту: Я,хочу,стати,вчителем,інформатики    Текст після сортування за абеткою: Я вчителем стати хочу інформатики    Текст після сортування за абеткою, ігноруючі регістр :вчителем стати хочу Я інформатики

Рис. 7.8 – Програма сортування слів у реченні

✓ Розглянемо задачі з використанням процедур та функцій.

**Приклад 7.5.** З використання функцій написати програму обчислення суми факторіалів.

► В програмі, поданій на рис. 7.9, процедура «factorial» отримує на вході значення змінної  $x$  (невід’ємне ціле число), а на виході – значення його факторіалі. В циклі основної програми реалізується звернення до функції. ◀



**Результат:** 1!+2!+3!=9

Рис. 7.9 – Програма сортування слів у реченні

В завданні №2 лабораторної роботи №7 спочатку потрібно знайти розвинення в степеневий ряд наданих функцій. Відповідна тема вивчалася в курсі «Математичний аналіз».

Нижче запропонуємо два способи наближеного обчислення значення функції з використанням її розвинення в ряд Маклорена. В першому способі залишимо таке ж формулювання, як і в завданнях лабораторної роботи, а в другому – знайдемо те саме значення, але з використанням іншого підходу. Причину розгляду двох способів пояснимо нижче.

**Приклад 7.6.** Знайти значення функції  $y = e^{-x}$ , розклавши її в ряд Маклорена до 20 доданка. Функція при знаходженні розкладу в ряд Маклорена повинна використовувати створені вами функції піднесення до степеня, визначення знака доданка та обчислення факторіала.

► Спочатку знайдемо розклад експоненціальної функції

$$e^a = 1 + \frac{a}{1!} + \frac{a^2}{2!} + \dots + \frac{a^k}{k!} + \dots \quad (a \in \mathbb{R}). \quad (7.1)$$

Покладемо  $a = -x$  отримаємо

$$e^{-x} = 1 - \frac{x}{1!} + \frac{x^2}{2!} + \dots + (-1)^k \frac{x^k}{k!} + \dots \quad (x \in \mathbb{R}). \quad (7.2)$$

В програмі на рис. 7.10 застосовується розклад (7.2). В ній створено окремі процедури та функції для обчислення

- факторіала невід’ємного натурального числа (factorial),
- піднесення до степеня (stepin),
- визначення знака доданку (znak).

В основній програмі здійснюється звернення до процедур та функцій в циклі, де обчислюється значення кожного доданку суми (7.2) та вони підсумовуються. ◀

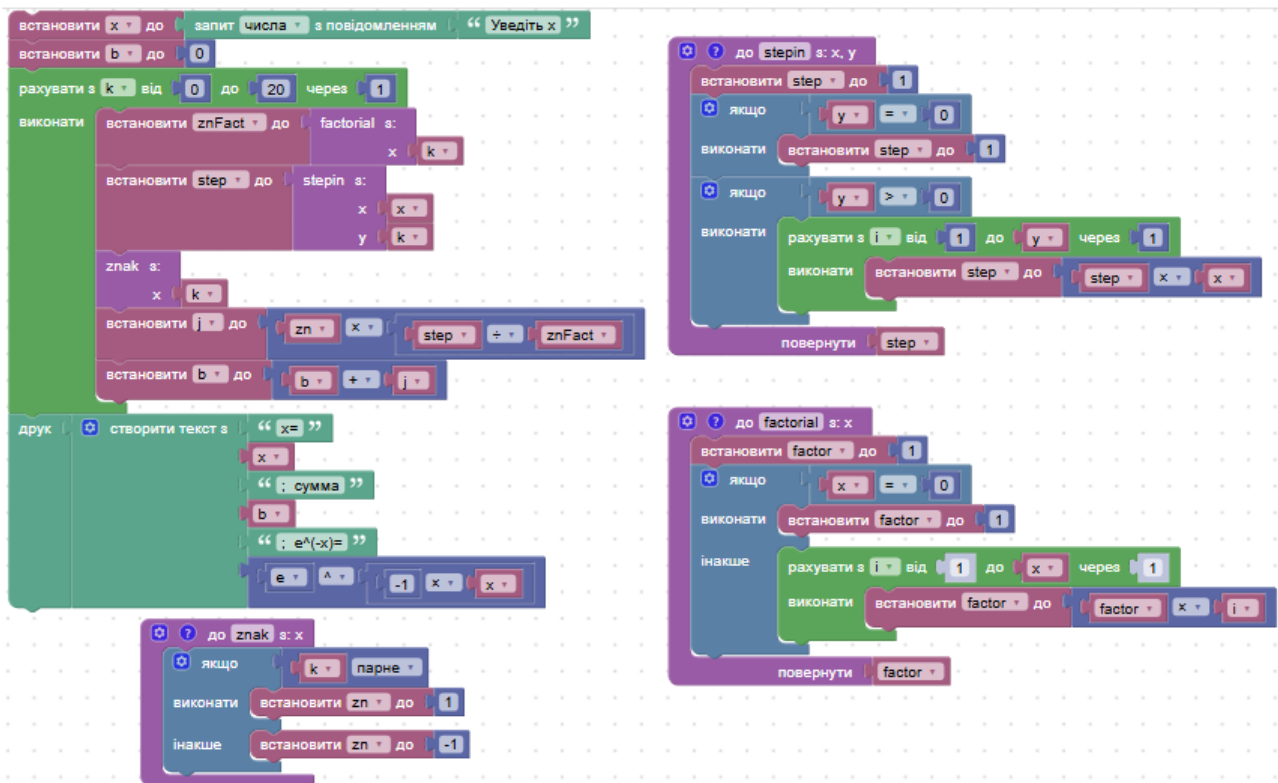
Функції, запропоновані в завданнях до лабораторної роботи, не передбачають обчислення факторіалів великих натуральних чисел. Проте, в прикладних задачах для наближеного обчислення значень функцій може виникнути наступна проблема. Для великих значень натуральних чисел неможливо обчислити їхній факторіал звичайними програмними засобами. Крім того, для значень аргумента функції, менших за 1, значення степеня буде малим числом. Множення малого числа на велике може призвести до значних помилок. Тому у другому способі використаємо таку ідею. Подамо доданок формули (7.1) у вигляді:

$$a^k = \frac{\overbrace{a \cdot a \cdot a \dots a}^{k \text{ множників}}}{\underbrace{1 \cdot 2 \cdot 3 \dots k}_{k \text{ множників}}} = \frac{a}{1} \cdot \frac{a}{2} \cdot \frac{a}{3} \cdot \dots \cdot \frac{a}{k} = \prod_{i=1}^k \frac{a}{i}, \quad (7.3)$$

що дозволить написати одну процедуру (7.1).

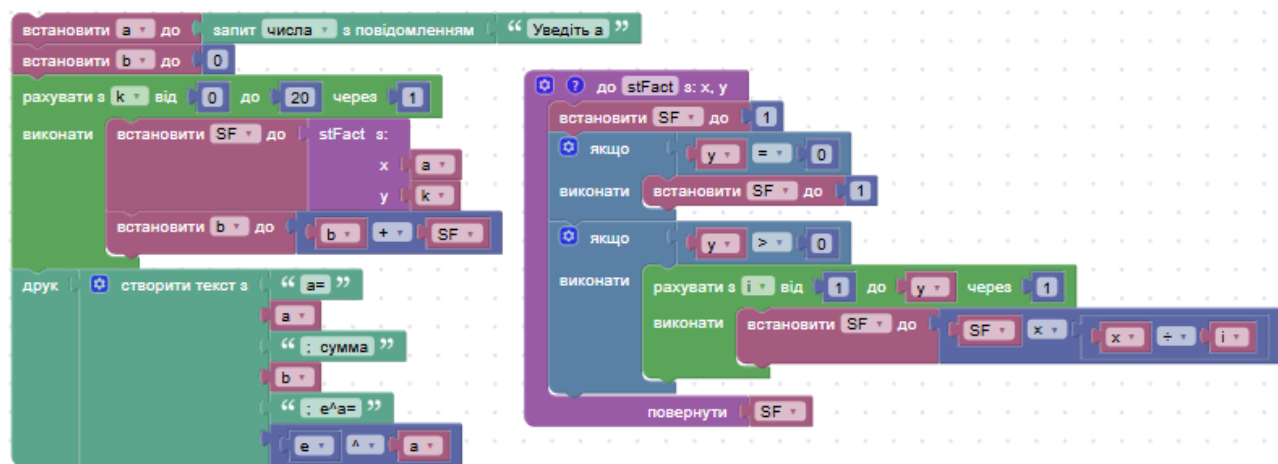
**Приклад 7.6.** Знайти значення функції  $y = e^{-x}$ , розклавши її в ряд Маклорена, до 20 доданка. При знаходженні розкладу в ряд Маклорена використовувати формули (7.1) та (7.3).

► На рис. 7.11 наведено програму, яка вирішує поставлене завдання. ◀



**Результат:** `x=2; сумма0.1353352832366504; e^(-x)=0.1353352832366127`

Рис. 7.10 – Програмна реалізація першого способу обчислення значень функції  $y = \exp(-x)$



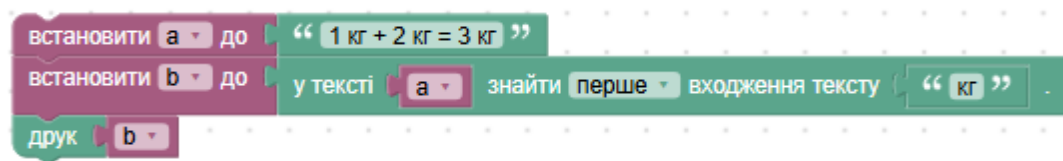
**Результат:** `a=-2; сумма0.13533528323665037; e^a=0.1353352832366127`

Рис. 7.11 – Програмна реалізація другого способу обчислення значень функції  $y = \exp(-x)$

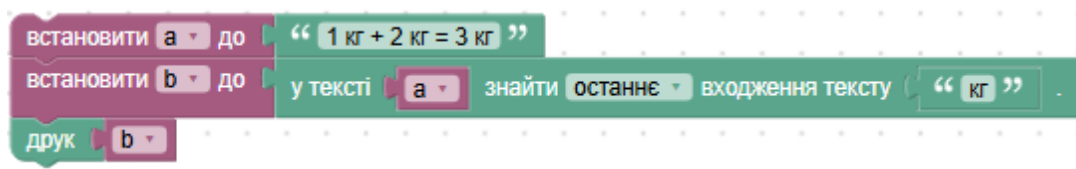
**Завдання 7.2.** Реалізувати програми з прикладів 7.1–7.6 в середовищі Google Blockly. Проаналізувати алгоритм їх роботи та принципи функціонування використаних команд.

### Контрольні питання до лабораторної роботи №7

1. Які особливості роботи з символьними змінними у Google Blockly ви знаєте?
2. Як можна поєднати два тексти в один?
3. Яку найбільшу довжину може мати текст у Google Blockly?
4. Поясніть частину програми. Що є результатом її виконання?

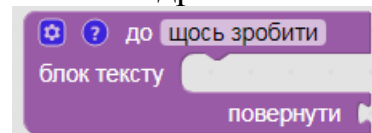
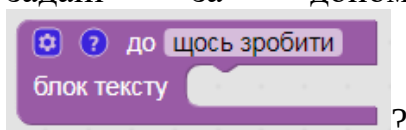


5. Поясніть частину програми. Що є результатом її виконання?



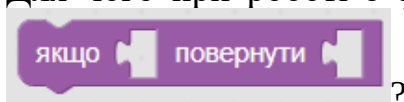
6. Які переваги та недоліки роботи з функціями у Google Blockly ви знаєте?
7. Які види функцій у Google Blockly вам відомі? Чим відрізняються функції,

задані за допомогою блоків



та

8. Для чого при роботі з функціями у Google Blockly використовують блок



## Лабораторна робота №8

## Тема: Робота з «Черепашкою»

**Мета:** Навчитись будувати зображення у візуальному середовищі розробки Google Blockly. Ознайомитись з можливостями інструмента «Черепашка». Опанувати особливості роботи з колірною палітрою.

## Теоретичні відомості

Для роботи з графічними зображеннями у візуальному середовищі розробки Google Blockly пропонується окрема група інструментів, так звана «Черепашка»<sup>1</sup>.

Користувачеві надається поле  $400 \times 400$  пікселів і розташована посередині «черепашка», що має декілька параметрів (рис. 8.1):

- напрям руху;
- колір;
- розмір пера;
- видимість пера;
- видимість черепашки.

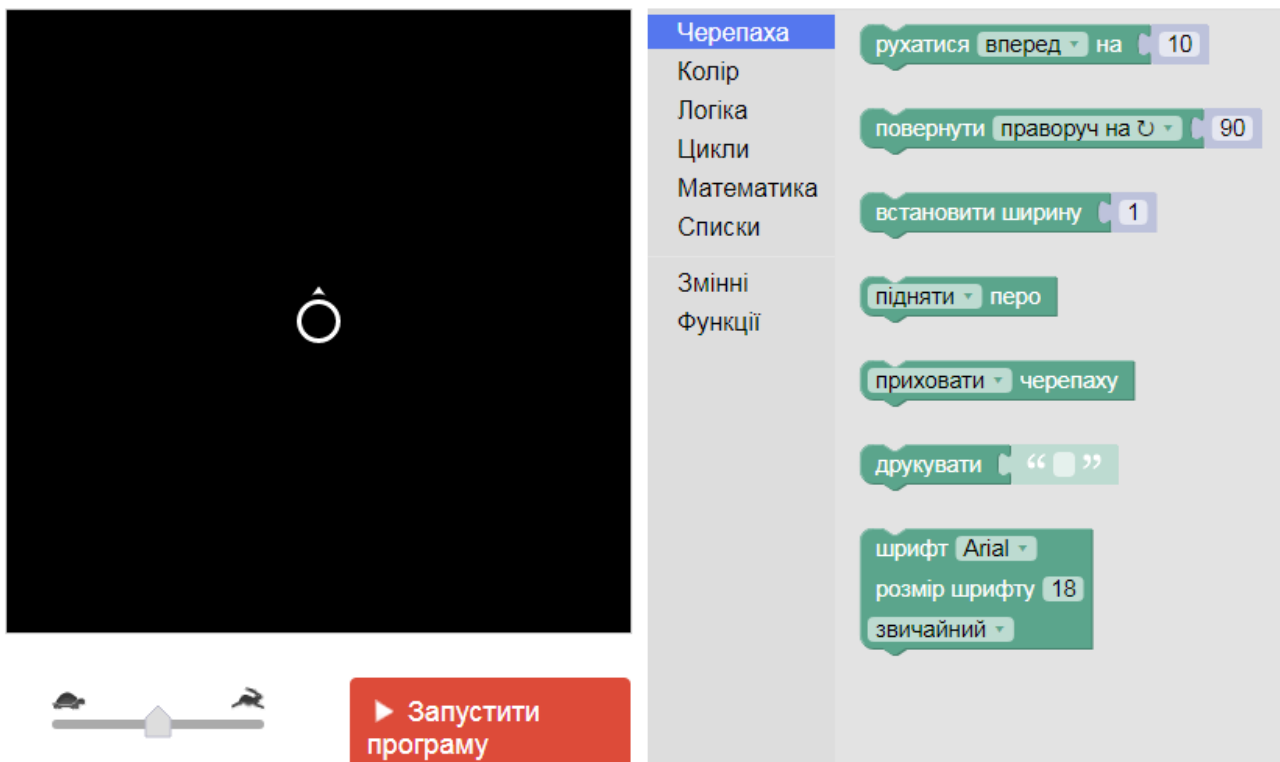
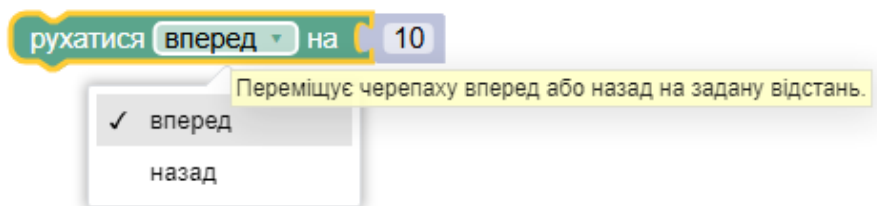


Рис. 8.1 – Команди категорії «Черепашка»

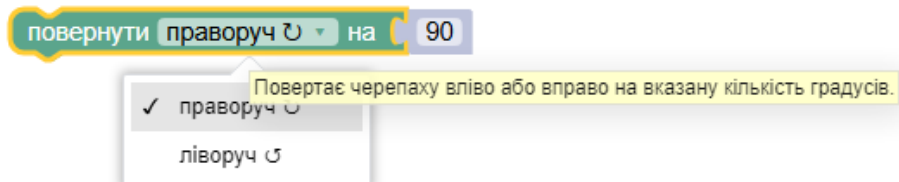
*Зверніть увагу, що «черепашка» являє собою коло зі стрілкою, яке вказує напрям її орієнтації для подальшого переміщення.*

<sup>1</sup> Ігри для майбутніх програмістів. Сайт навчальних ігор Blockly. URL : <https://blockly.games/?lang=uk>

«Черепашка» вміє рухатись вперед і назад на задану довільну кількість пікселів:



обертатися праворуч чи ліворуч на будь-яку кількість градусів.



Крім того, надається можливість друку текстових повідомлень з параметрами шрифту, його розміру та стилю. На рис. 8.2 наведено код програми, що демонструє використання такого функціоналу.

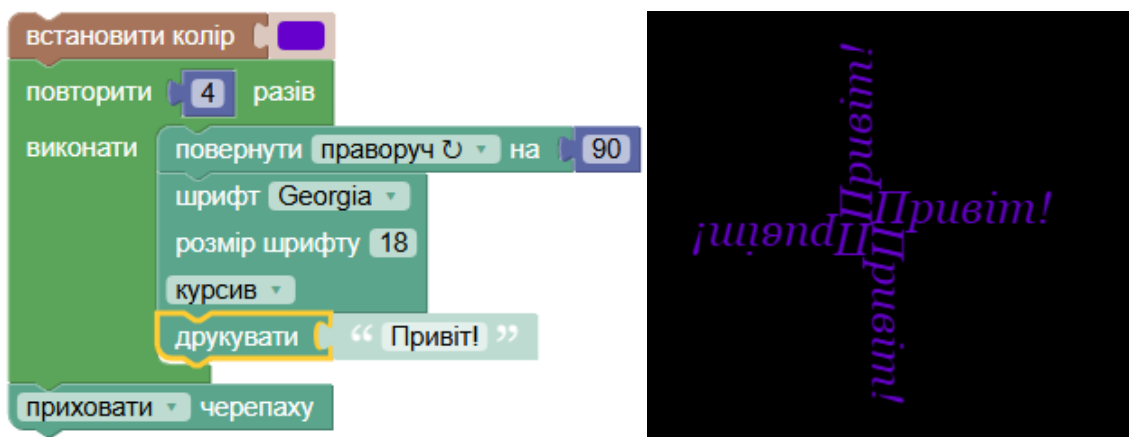
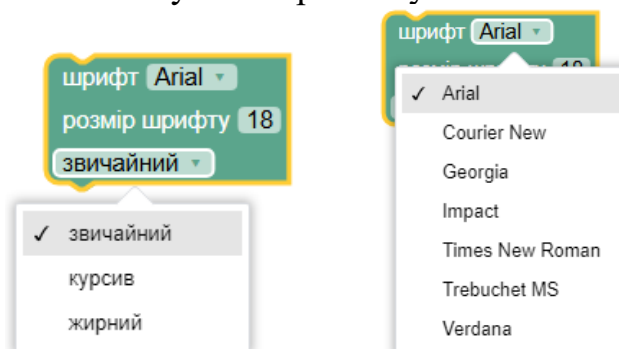


Рис. 8.2 – Програма, що використовує блок текстових шрифтів в середовищі

Друк тексту починається з того положення, де розташована черепашка, і в тому напрямку, який вона вказує. Колір тексту такий самий, як і колір пера.



Слід звернути увагу, що для написання програм, яка працюють з графічними зображеннями, можна використовувати всі раніше розглянуті блоки команд, в тому числі цикли, списки кольорів та рекурсії.

В окремій групі інструментів «Колір» користувачеві надається можливість встановлення кольору «черепахи» з бібліотеки, випадковим чином,

в режимі RGB або шляхом поєднання двох кольорів у відсотковому співвідношенні (див. рис. 8.3).

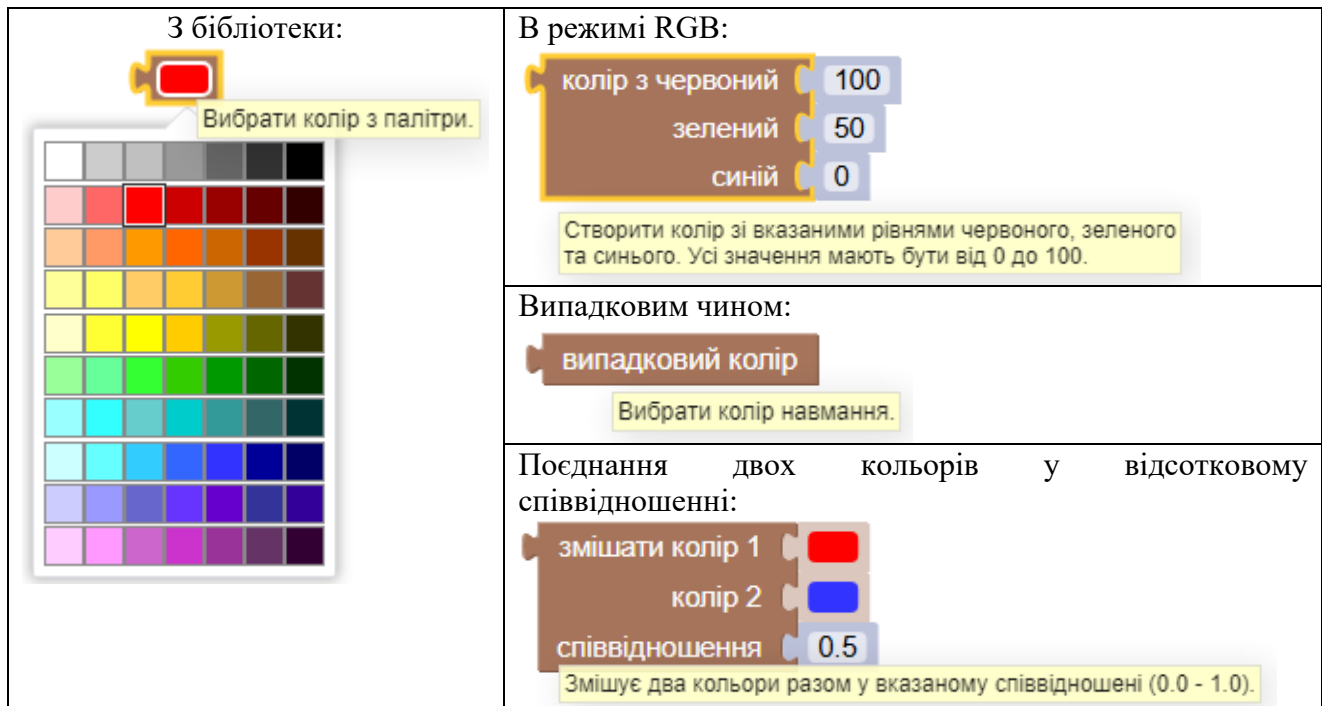


Рис. 8.3 – Функціонал команд категорії «Колір»

Головними недоліками «черепахи» є низька швидкість виконання програм та відсутність можливості контролю розташування пера на полі. Немає ані функцій, які б повертали теперішнє місцезнаходження «черепахи», ані функцій, які б його встановлювали. Для регулювання швидкості виконання дій слід скористатися відповідним бігунцем



## Методичні рекомендації до лабораторної роботи №8

**Приклад 8.1.** Реалізувати побудову круга.

► Програма на рис. 8.4 передбачає побудову 360 копій радіуса. ◀

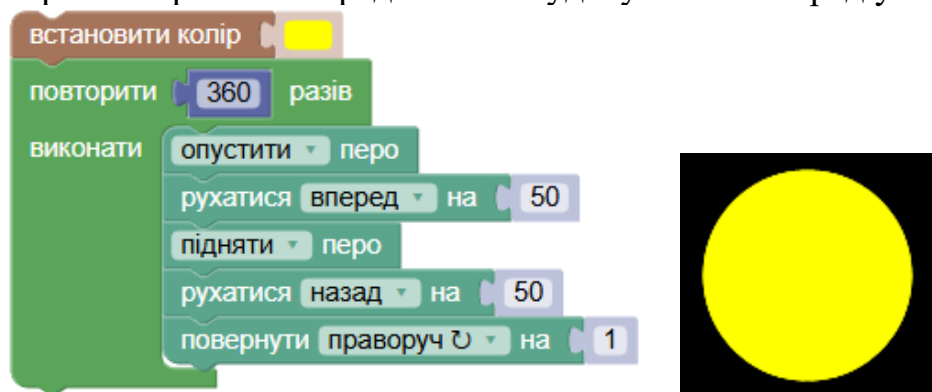


Рис. 8.4 – Програма побудови круга

Щоб повністю зафарбувати поле, можна намалювати круг достатньо великого радіуса відповідним кольором.

Аналогічно можна будувати сектор (повторювати меншу кількість разів) та півколо. Також можна копіювати круги та наносити один на інший.

**Приклад 8.2.** Реалізувати побудову трикутника.

► Програма на рис. 8.5 передбачає побудову великої кількості паралельних відрізків. ◀



**Результат:**

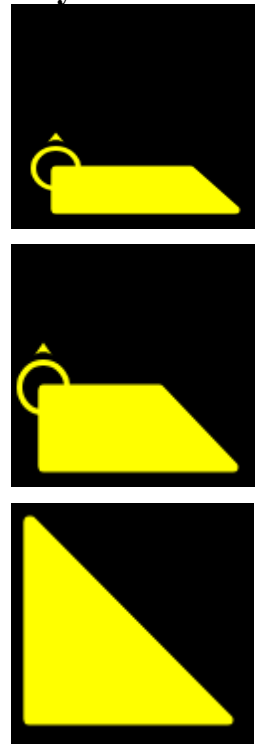


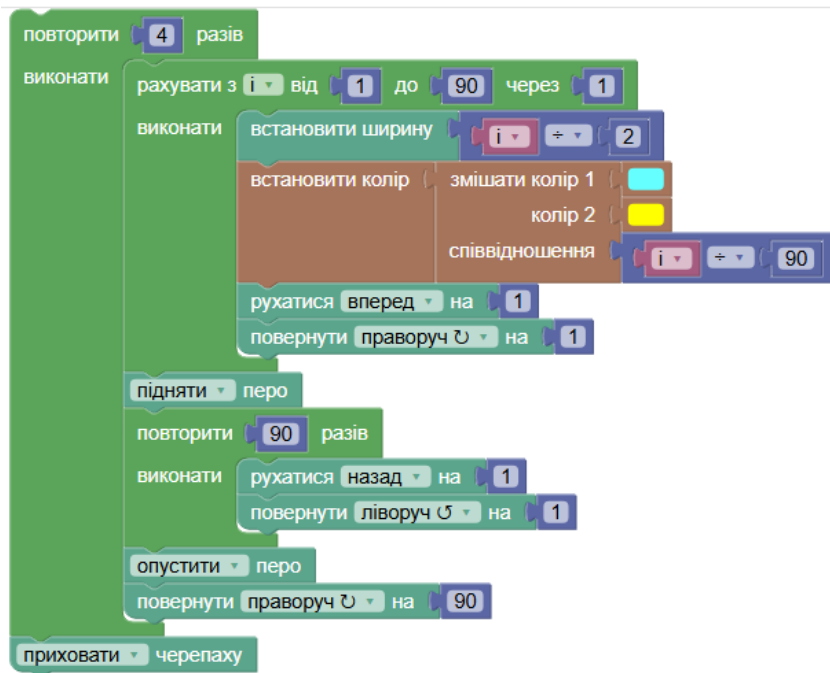
Рис. 8.5 – Побудова заповненого трикутника

Аналогічно можна будувати прямокутник та квадрат

**Завдання 8.1.** Внесіть в середовище «Ігри Blockly Черепашка» наведені в прикладах 8.1. та 8.2 програми. Також складіть програму, наведену на рис. 8.6, що демонструє можливості змішування кольорів.

Як було зазначено вище, стрілка «черепашки» демонструє її напрям руху. Тому тут можна реалізувати побудову «ліній» в полярній системі координат. Якщо вважати центр екрана полюсом полярної системи координат, то відстань, відкладена від нього, буде полярною, що визначається рівнянням лінії. Зважаючи на те, що зазначена відстань має бути невід'ємною, код програми на рис. 8.7 включає відповідну перевірку в структурі розгалуження.

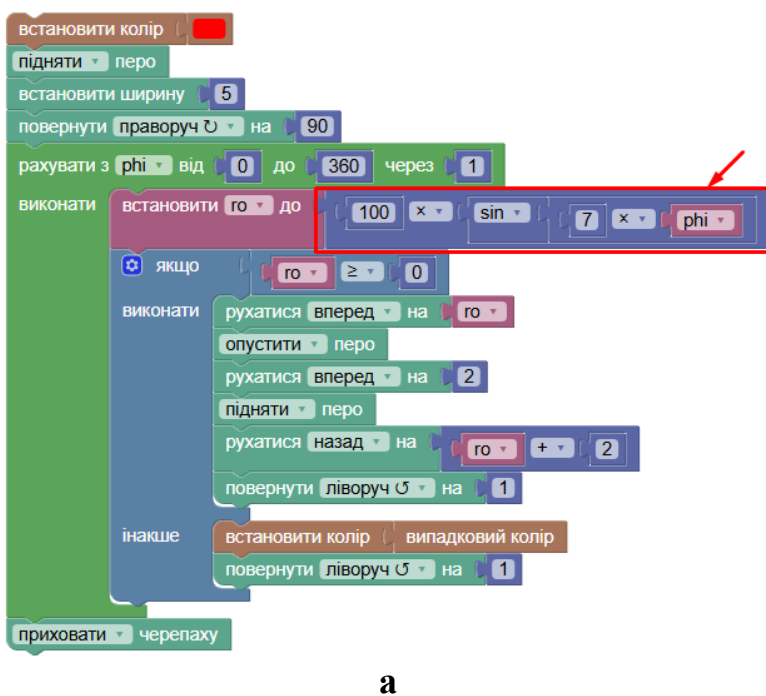
**Завдання 8.2.** Внесіть код програми з рис. 8.7,а до середовища «Ігри Blockly Черепашка», дослідіть принципи її роботи. Отримайте результат, наведений на рис. 8.7, б. Рівняння семилисника у даному випадку має вигляд  $\rho = 100 \sin 7\varphi$ . Замінюючи фрагмент, виділений червоним прямокутником, блоками, що відповідатимуть рівнянню кадріюди  $\rho = 75(1 - \cos \varphi)$ , отримаєте лінію, як на рис. 8.7, г. Скеруйте підняття та опускання пера так, щоб отримати суцільно зафарбовані фігури, як на рис. 8.7, в, д.



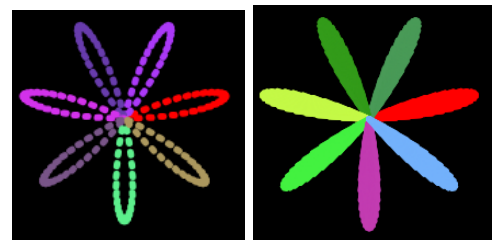
Результат:



Рис. 8.6 – Програма для демонстрації функціоналу змішування кольорів

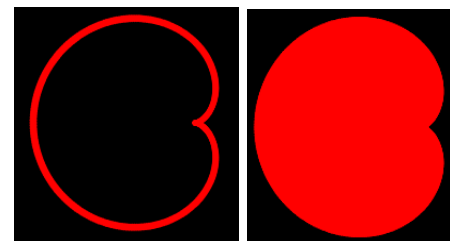


а



б

в



г

д

Рис. 8.7 – Програма побудови «лінії», заданої в полярній системі координат

### Завдання до лабораторної роботи №8

- Щоб розпочати роботу, перейдіть за посиланням:

<https://blockly-games.appspot.com/turtle>

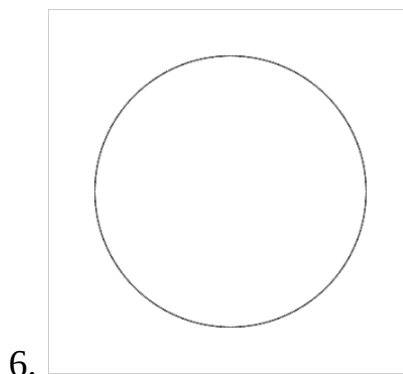
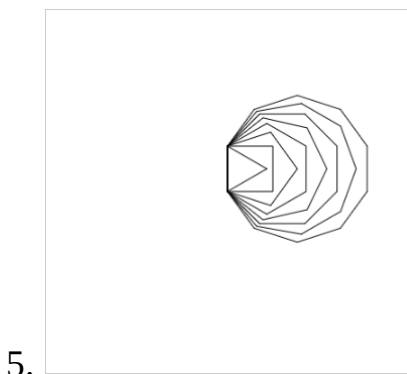
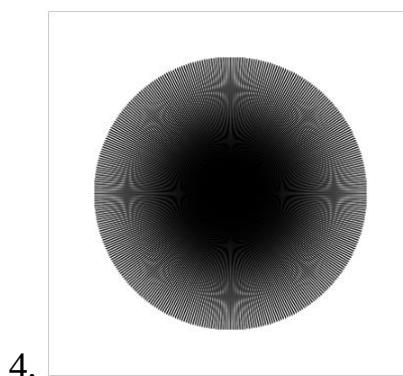
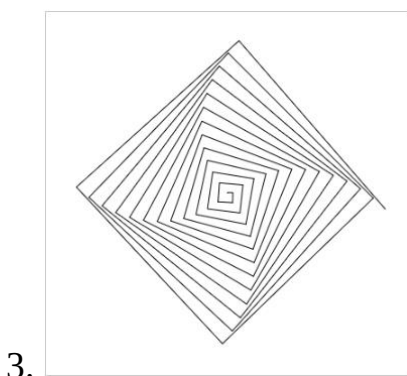
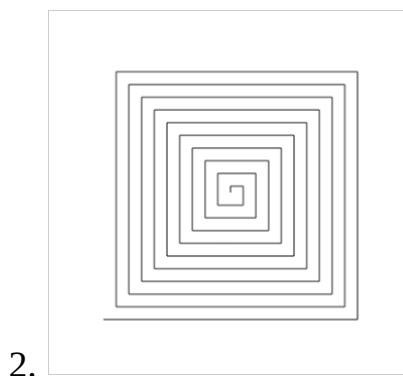
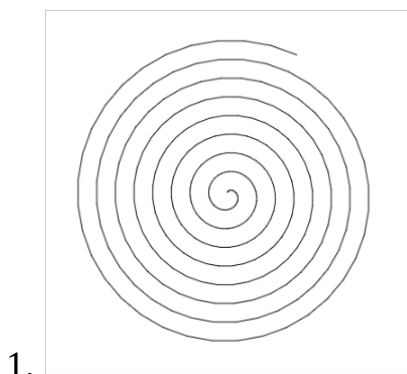
За посиланням ви знайдете легку навчальну програму з 10 завдань, і після розв'язання останньої ви зможете вільно створювати власні малюнки, а також зберігати їх до загальної галереї.

- Побудуйте пейзаж. При побудові використайте наступне:

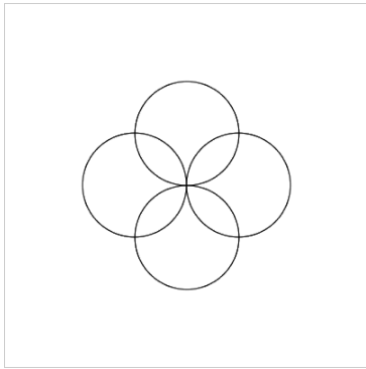
- 1) Зафарбуйте фон одним або декількома кольорами, наприклад, небо та земля (трава, сніг).
- 2) Використайте побудову трикутників, кругів та квадратів та їх частин.
- 3) Задавайте колір різними способами.
- 4) Побудуйте веселку, хмарки (за бажанням дощ або сніг).
- 5) За бажанням річку, дерева, хатинку.
- 6) Не забудьте приховати черепашку.

- Намалюйте фігуру з першої та другої групи. Розмірами фігур, їх кольором та іншими малозначущими параметрами можна знехтувати.

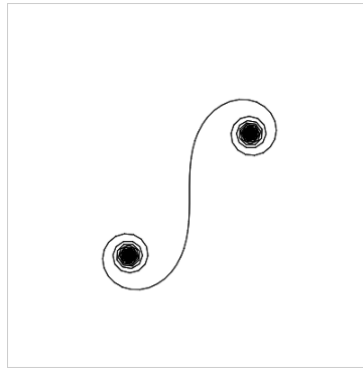
### 1 група завдань:



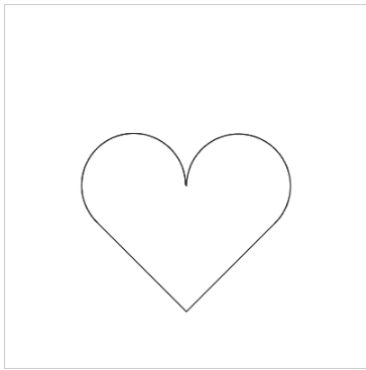
7.



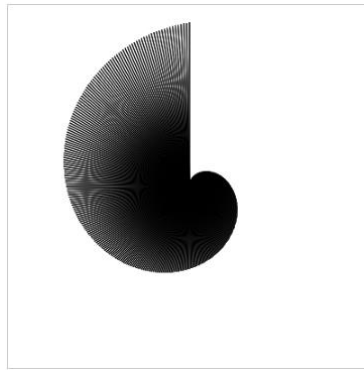
8.



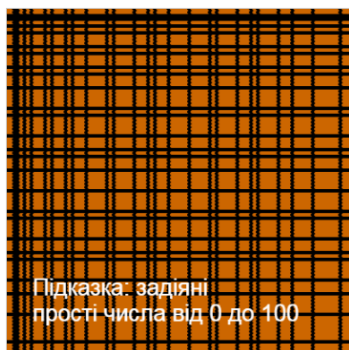
9.



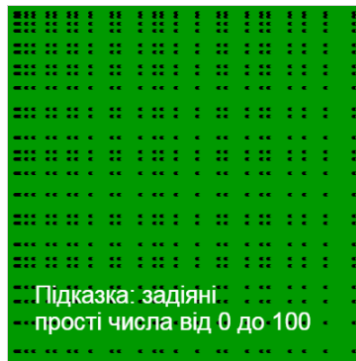
10.



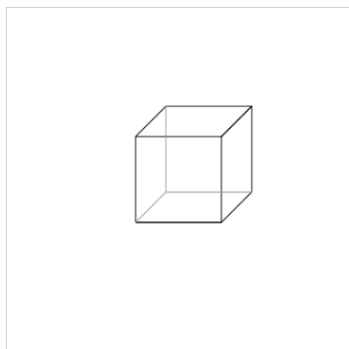
11.



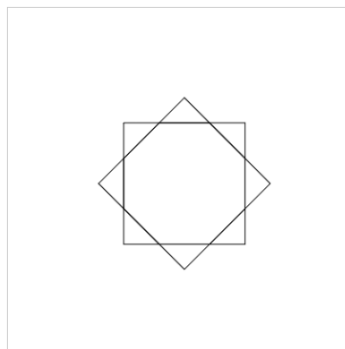
12.



13.

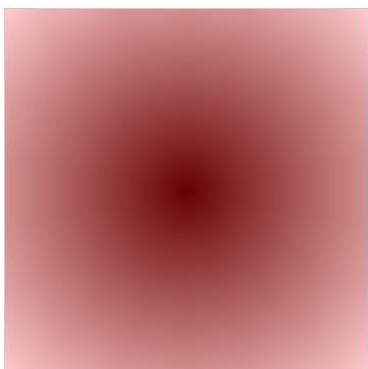


14.

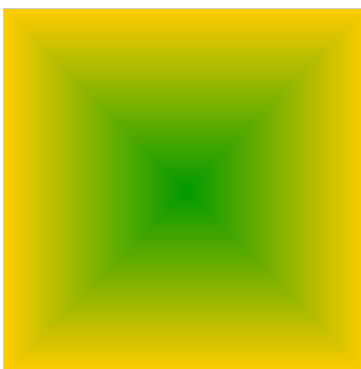


**2 група завдань:**

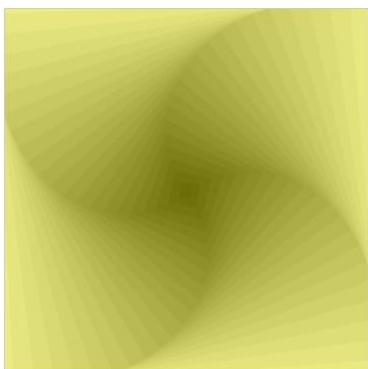
1.



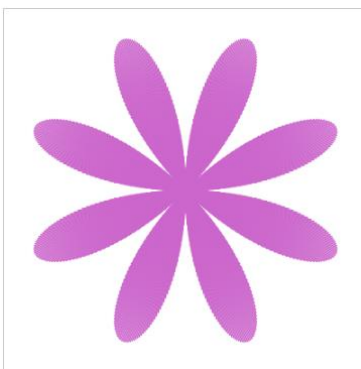
2.



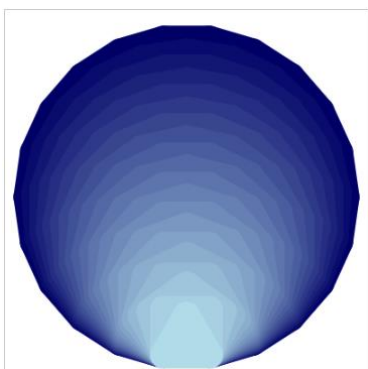
3.



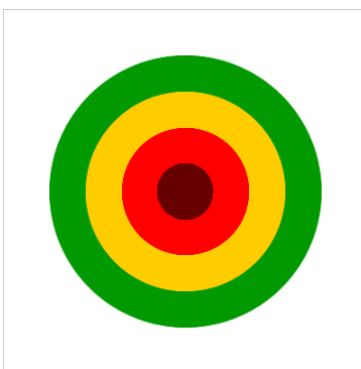
4.



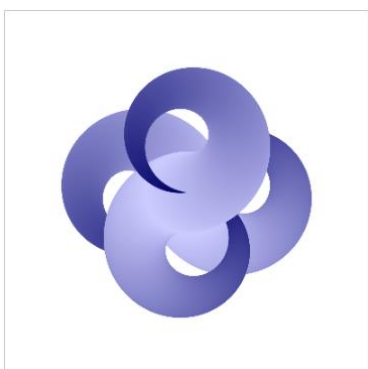
5.



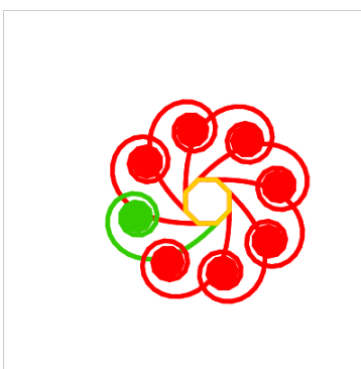
6.



7.



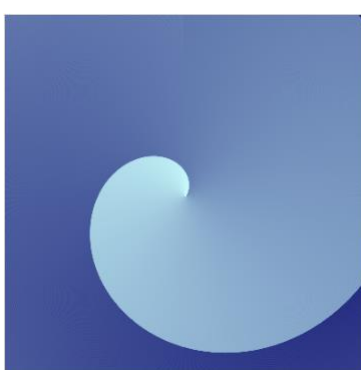
8.

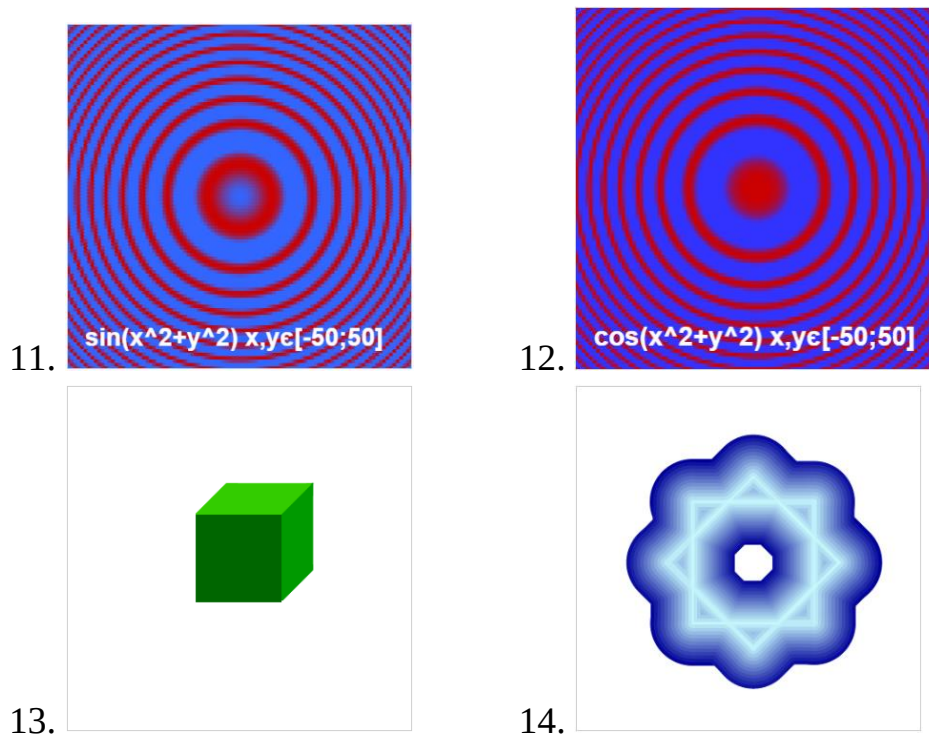


9.



10.

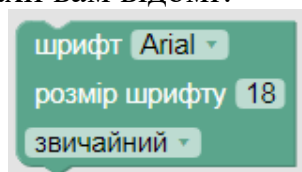




### Контрольні питання до лабораторної роботи №3

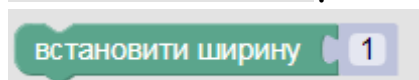
1. Які переваги та недоліки «черепахи» ви знаєте?
2. Які способи встановлення кольору пера черепахи вам відомі?

3. Для чого при малюванні використовуюсь блок



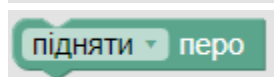
?

4. Для чого при малюванні використовуюсь блок



?

5. Для чого при малюванні використовуюсь блок



?

5. Наведіть приклад використання блоку



. Чим його

використання відрізняється від використання блоку



?

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Алудден Й., Гамбель Ф., Фігус В., Вагліасінді Ф. Програмування для дітей: Створюй відеоігри за допомогою Скретч. Харків : Ранок, 2019. 128 с.
2. Алудден Й., Гамбель Ф., Фігус В., Вагліасінді Ф. Програмування для дітей: Створюй анімації за допомогою Скретч. Харків : Ранок, 2019. 128 с.
3. Діккінс Р., Мелмот Дж., Стовелл Л. Scratch. Кодування для майбутніх програмістів. Київ : КМ-Букс, 2019. 96 с.
4. Тітова О. О., Мухін В. В. Візуальні мови програмування : методичні рекомендації до виконання лабораторних робіт для здобувачів ступеня вищої освіти бакалавра спеціальності «Інженерія програмного забезпечення» освітньо-професійної програми «Програмна інженерія». Запоріжжя : ЗНУ, 2020. 50 с.

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Кайдан Н. В., Тараненко Г. І. Мотивація освітнього процесу засобами гейміфікації. *Збірник наукових праць фізико-математичного факультету ДДПУ*. 2023. Вип. 13. С. 74–78.
2. Квак П. А., Чернікова Л. А. Програмування через реалізацію проєктів: стратегії та практика в школі. *Відкрите освітнє е-середовище сучасного університету : електронне наукове фахове видання*. 2024. № 16. С. 35–49.
3. Тітова О. О., Мухін В. В. Візуальні мови програмування : методичні рекомендації до виконання лабораторних робіт для здобувачів ступеня вищої освіти бакалавра спеціальності «Інженерія програмного забезпечення» освітньо-професійної програми «Програмна інженерія». Запоріжжя : ЗНУ, 2020. 50 с.
4. Яценко О. І., Чумак Л. М. Критерії добору середовища навчання програмування для формування ІКТ-компетентності майбутніх учителів початкової школи. *Інформаційні технології і засоби навчання*. 2020. Т. 78, № 4. С. 219-236.
5. Erol O., Çırak N. S. The effect of a programming tool scratch on the problem-solving skills of middle school students. *Educ Inf Technol*. 2022. Vol. 27. P. 4065–4086.
6. Olsson J., Granberg C. Teacher-student interaction supporting students' creative mathematical reasoning during problem solving using Scratch. *Mathematical Thinking and Learning*. 2022. Vol. 26, No. 3. P. 278–305.
7. Rosydiana E. A., Sudjimat D. A., Utama C. The Effect of Digital Learning Media Using Scratch Game Based Learning on Student Problem Solving Skills. *Jurnal Penelitian Pendidikan IPA*. 2023. Vol. 9, No. 11. P. 10010–10015.
8. Sidhu R. Coding for Kids in Scratch 3: The Complete Guide to Creating Art, Artificial Intelligence, and Computer Games for Beginners Paperback. Independently published, 2019. 149 p.

9. Su Y.-S., Shao M., Zhao L. Effect of Mind Mapping on Creative Thinking of Children in Scratch Visual Programming Education. *Journal of Educational Computing Research*. 2022. Vol. 60. No. 4. P. 906-929.
10. Wainwright M. 25 Scratch 3 Games for Kids: A Playful Guide to Coding Paperback. No Starch Press, 2019. 128 p.

**Інформаційні джерела:**

1. Вивчаємо Scratch разом. URL : <http://surl.li/hxyobb>
2. Ігри для майбутніх програмістів. *Сайт навчальних ігор Blockly*. URL : [https://blockly.games/?lang=uk\\_\\_](https://blockly.games/?lang=uk__)
3. Про візуальне програмування та Google Blockly. *Сайт прикладної математики primat.org*. URL : <http://surl.li/ukxzep>
4. Спільнота та середовище Scratch. URL : <https://scratch.mit.edu/>
5. Blockly для майбутніх програмістів. *Сайт Blockly*. URL : <http://blockly.ru/>
6. Powerful coding, block by block. URL : <https://developers.google.com/blockly/>

## СПИСОК ІЛЮСТРАЦІЙ

|                                                                                                                                          |    |
|------------------------------------------------------------------------------------------------------------------------------------------|----|
| Рис. 1.1 – Стартова сторінка візуального середовища Scratch .....                                                                        | 8  |
| Рис. 1.2 – Інтерфейс вікна середовища візуального програмування Scratch .....                                                            | 9  |
| Рис. 1.3 – Вікно обрання розширень .....                                                                                                 | 9  |
| Рис. 1.4 – Команди категорії «Рух» .....                                                                                                 | 10 |
| Рис. 1.5 – Команди категорії «Вигляд» .....                                                                                              | 10 |
| Рис. 1.6 – Команди категорії «Звук» .....                                                                                                | 11 |
| Рис. 1.7 – Команди категорії «Події» .....                                                                                               | 11 |
| Рис. 1.8 – Команди категорії «Керування» .....                                                                                           | 11 |
| Рис. 1.9 – Команди категорії «Датчики» .....                                                                                             | 12 |
| Рис. 1.10 – Команди категорії «Оператори» .....                                                                                          | 12 |
| Рис. 1.11 – Команди категорії «Змінні» .....                                                                                             | 12 |
| Рис. 1.12 – Команди категорії «Змінні» .....                                                                                             | 13 |
| Рис. 1.13 – Робота з файлами програм .....                                                                                               | 13 |
| Рис. 1.14 – Бібліотека спрайтів .....                                                                                                    | 14 |
| Рис. 1.15 – Видалення спрайта зі сцени .....                                                                                             | 14 |
| Рис. 1.16 – Бібліотека «Оберіть тло» .....                                                                                               | 15 |
| Рис. 1.17 – Код програми з додаванням привітання .....                                                                                   | 15 |
| Рис. 1.18 – Орієнтація спрайта на сцені на координатній сітці .....                                                                      | 16 |
| Рис. 1.19 – Код програми переміщення спрайта .....                                                                                       | 16 |
| Рис. 1.20 – Параметри орієнтації спрайта на сцені .....                                                                                  | 16 |
| Рис. 1.21 – Код програми, що передбачає відбиття спрайта від межі сцени .....                                                            | 16 |
| Рис. 1.22 – Програма керування рухами спрайта .....                                                                                      | 17 |
| Рис. 2.1 – Запис аудіо для подальшого відтворення спрайтом .....                                                                         | 19 |
| Рис. 2.2 – Вікно інструменту «Звуки» .....                                                                                               | 19 |
| Рис. 2.3 – Функціонал кнопки «Обрати звук» .....                                                                                         | 19 |
| Рис. 2.4 – Бібліотека звуків .....                                                                                                       | 19 |
| Рис. 2.5 – Команди категорій «Музика» (а) і «Текст у мову» (б) .....                                                                     | 20 |
| Рис. 2.6 – Використання команд категорії «Музика» .....                                                                                  | 20 |
| Рис. 2.7 – «Танок» спрайта під ударні інструменти .....                                                                                  | 23 |
| Рис. 2.8 – Опрацювання розмірів спрайта .....                                                                                            | 23 |
| Рис. 2.9 – Код програми танцю спрайта Jovi Dance .....                                                                                   | 23 |
| Рис. 2.10 – Керування музичними інструментами .....                                                                                      | 24 |
| Рис. 2.11 – Циклічне повторення звучання інструмента .....                                                                               | 24 |
| Рис. 2.12 Датчик «клавішу... натиснуто?» .....                                                                                           | 25 |
| Рис. 2.13 – Приклад коду з реакцією рухомого персонажа на зовнішній вплив .....                                                          | 25 |
| Рис. 3.1 – Команди циклічної структури .....                                                                                             | 26 |
| Рис. 3.2 – Код програми з використанням команди циклічної структури і розширення «Олівець» .....                                         | 26 |
| Рис. 3.3 – Команди розгалуженої структури .....                                                                                          | 27 |
| Рис. 3.4 – Команди очікування, зупинки, керування клонами .....                                                                          | 27 |
| Рис. 3.5 – Приклади вкладених структур .....                                                                                             | 27 |
| Рис. 3.6 – Команди категорії «Датчики»: логічні вирази, що перевіряють певну умову .....                                                 | 28 |
| Рис. 3.7 – Команди категорії «Датчики»: ідентифікатори значень, перезапуску та дозволу, постановки питань для отримання відповідей ..... | 28 |
| Рис. 3.8 – Класифікація команд категорії «Оператори» .....                                                                               | 28 |
| Рис. 3.9 – Код програми діалогу між спрайтом та користувачем з діями, що залежать від відповідей .....                                   | 31 |
| Рис. 3.10 – Додавання до програми кодів ефектів перетворення тексту в мову .....                                                         | 32 |

|                                                                                                                        |    |
|------------------------------------------------------------------------------------------------------------------------|----|
| Рис. 3.11 – Повідомлення про кількість набраних очок.....                                                              | 32 |
| Рис. 3.12 – Код програми гри спрайта з м'ячем.....                                                                     | 33 |
| Рис. 4.1 – Функціонал команд категорії «Змінні».....                                                                   | 34 |
| Рис. 4.2 – Створення змінної.....                                                                                      | 34 |
| Рис. 4.3 – Створення порожнього списку .....                                                                           | 35 |
| Рис. 4.4 – Функціонал команд підкатегорії «Списки» в категорії «Змінні» .....                                          | 35 |
| Рис. 4.5 – Складання списку оцінок учнів групи .....                                                                   | 36 |
| Рис. 4.6 – Гра для перевірки знань таблиці множення .....                                                              | 38 |
| Рис. 4.7 – Код програми підрахунку чисел, кратних 10 .....                                                             | 39 |
| Рис. 4.8 – Побудова ліній, заданих в полярній системі координат .....                                                  | 39 |
| Рис. 4.9 – Код та блок-схема програми пошуку найбільшого елемента списку та його номера .....                          | 40 |
| Рис. 5.1 – Інтерфейс середовища візуального програмування: Google Blockly.....                                         | 41 |
| Рис. 5.2 – Створення змінної в середовищі Google Blockly .....                                                         | 42 |
| Рис. 5.3 – Присвоєння значення змінній різними способами .....                                                         | 42 |
| Рис. 5.4 – Програма, яка демонструє виконання арифметичних дій з числами в Google Blockly .....                        | 43 |
| Рис. 5.5 – Команди категорії «Математика» .....                                                                        | 43 |
| Рис. 5.6 – Математичні операції вз каталогу «Математика» .....                                                         | 44 |
| Рис. 5.7 – Команди категорії «Логіка» .....                                                                            | 44 |
| Рис. 5.8 – Утворення різних варіантів структур усови.....                                                              | 45 |
| Рис. 5.9 – Програма з використанням команд категорії «Математика» .....                                                | 46 |
| Рис. 5.10 – Програма обчислення площі трикутника за формулою Герона.....                                               | 47 |
| Рис. 5.11 – Програма розгалуженої структури для визначення парності числа .....                                        | 47 |
| Рис. 5.12 – Програма розгалуженої структури для обчислення значення функції з різним заданням на різних множинах ..... | 48 |
| Рис. 6.1 – Команди категорії «Петлі».....                                                                              | 50 |
| Рис. 6.2 – Цикл з нескінченною кількістю повторень .....                                                               | 50 |
| Рис. 6.3 – Команди категорії «Списки» .....                                                                            | 51 |
| Рис. 6.4 – Програма пошуку найбільшого елемента списку випадкових чисел з визначенням місця його розміщення .....      | 52 |
| Рис. 6.5 – Програма обчислення квадрата факторіала натурального числа.....                                             | 54 |
| Рис. 6.6 – Пошук найменшої оцінки за контрольну роботу без використання масивів .....                                  | 55 |
| Рис. 6.7 – Програма наближеного обчислення суми з використанням циклу з передумовою .....                              | 55 |
| Рис. 6.8 – Програма реалізації гри «Вгадай число!» .....                                                               | 56 |
| Рис. 6.9 – Програма створення масиву, що складається з частин іншого масиву .....                                      | 57 |
| Рис. 6.10 – Програма сортування масиву з використанням вбудованого функціоналу Google Blockly .....                    | 57 |
| Рис. 7.1 – Команди категорії «Текст» .....                                                                             | 59 |
| Рис. 7.2 – Функціонал деяких команд категорії «Текст» .....                                                            | 60 |
| Рис. 7.3 – Програма зміни регістру в тексті .....                                                                      | 61 |
| Рис. 7.4 – Команди категорії «Функції та робота з ними».....                                                           | 61 |
| Рис. 7.5 – Програма заміни символу «і» символом «*» в тексті .....                                                     | 63 |
| Рис. 7.6 – Програма заміни подвоєних букв на одну.....                                                                 | 64 |
| Рис. 7.7 – Програма пошуку другого слова в тексті (слова в тексті відокремлені пробілами) .....                        | 64 |
| Рис. 7.8 – Програма сортування слів у реченні .....                                                                    | 65 |
| Рис. 7.9 – Програма сортування слів у реченні .....                                                                    | 66 |

Рис. 7.10 – Програмна реалізація першого способу обчислення значень функції  $y=\exp(-x)$  ..67

Рис. 7.11 – Програмна реалізація другого способу обчислення значень функції  $y=\exp(-x)$  ...67

Рис. 8.1 – Команди категорії «Черепаха» .....69

Рис. 8.2 – Програма, що використовує блок текстових шрифтів в середовищі.....70

Рис. 8.3 – Функціонал команд категорії «Колір».....71

Рис. 8.4 – Програма побудови круга .....71

Рис. 8.5 – Побудова заповненого трикутника .....72

Рис. 8.6 – Програма для демонстрації функціоналу змішування кольорів .....73

Рис. 8.7 – Програма побудови «лінії», заданої в полярній системі координат .....73

---

**НАВЧАЛЬНО-МЕТОДИЧНЕ ВИДАННЯ**  
(українською мовою)

**Пшенична Олена Станіславівна  
Д'яченко Наталія Миколаївна  
Красікова Ірина Володимирівна  
Тітова Ольга Олександрівна**

**ВІЗУАЛЬНЕ ПРОГРАМУВАННЯ**

Методичні рекомендації до лабораторних занять  
для здобувачів ступеня вищої освіти бакалавра  
спеціальності «Середня освіта» освітньо-професійної програми  
«Середня освіта (Інформатика)»

Рецензент *К.С. Решевська*  
Відповідальний за випуск *Г.М. Шило*  
Коректор *Н.М. Д'яченко*