

Міністерство освіти і науки України
Запорізький національний університет

О.О. Тітова, В.В. Мухін

ВІЗУАЛЬНІ МОВИ ПРОГРАМУВАННЯ

Методичні рекомендації до виконання лабораторних робіт
для здобувачів ступеня вищої освіти бакалавра
спеціальності «Інженерія програмного забезпечення»
освітньо-професійної програми «Програмна інженерія»

Затверджено
вченою радою ЗНУ
Протокол № від

Запоріжжя
2020

УДК: 004.434: 004.8(076.5)
Т454

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.

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

Для здобувачів ступеня вищої освіти бакалавра спеціальності «Інженерія програмного забезпечення» освітньо-професійної програми «Програмна інженерія».

Рецензент

С. І. Гоменюк, доктор технічних наук, професор, декан математичного факультету

Відповідальний за випуск

О. О. Тітова, кандидат технічних наук, доцент кафедри фундаментальної математики

ЗМІСТ

Вступ.....	4
Загальні рекомендації до виконання лабораторних робіт.....	6
Лабораторна робота №1 Введення у візуальні мови програмування.....	7
Лабораторна робота №2 Каталог Math. Математичні дії, функції.....	10
Лабораторна робота №3 Каталог Logic. Оператори порівняння.....	14
Лабораторна робота №4 Каталог Control. Елементи управління циклами.....	19
Лабораторна робота №5 Каталог Text. Робота з текстами та символьними змінними.....	24
Лабораторна робота №6 Каталог Lists. Елементи для роботи з масивами.....	30
Лабораторна робота №7 Каталог Color. Робота з колірною палітрою.....	34
Лабораторна робота №8 Каталог Procedures. Створення функцій і підпрограм.....	43
Рекомендована література.....	47
Використана література.....	47
Додаток А. Деякі алгоритми обробки даних	48

ВСТУП

Візуальні мови програмування – один із спеціальних курсів професійної підготовки фахівця з програмної інженерії. Візуальне програмування взагалі – це спосіб створення програм шляхом маніпулювання графічними об'єктами замість написання програмного коду в текстовому вигляді. Візуальне програмування дозволяє програмувати, використовуючи графічні або символічні елементи, якими можна маніпулювати інтерактивним чином згідно з деякими правилами, причому просторове розташування графічних об'єктів використовувати як елементи синтаксису програми. Значна частина візуальних мов програмування базується на ідеї «фігур і ліній», де фігури розглядаються як суб'єкти і з'єднуються лініями (стрілками, дугами тощо), які являють собою відношення.

Візуальні мови програмування можна вивчати на базі візуальних середовищ, таких, як Scratch, Blockly та інших подібних візуальних онлайн-середовищ. При цьому надається можливість описувати процеси в легкому для розуміння поданні і достатньому рівні абстракції. У програмуванні величезна доля зусиль витрачається на специфікацію конкретної мови програмування: синтаксис, правила написання (десь потрібна крапка з комою, десь потрібні фігурні дужки, десь взагалі ні один вислів не обходиться без дужок). Середовище Google Blockly було створено для нового стилю програмування, у якому головну роль грає візуалізація [1]. Це спеціальне середовище, в якому користувачі перетягують візуальні блоки, щоб створити додаток. Дані методичні рекомендації будуть присвячені роботі саме з середовищем Google Blockly.

Візуальне програмування підтримує основні концепції програмування: ви можете побачити всім відомі умовні оператори, блоки циклів з умовою та передумовою, математичні блоки, блоки для роботи з текстом, для створення підпрограм тощо. До переваг візуального програмування можна віднести наступне: легко вивчати, просте і гнучке, не потрібні серйозні навички програмування, просте управління, можливість експорту програми в JavaScript, Python, Dart, PHP або XML, відкритий вихідний код.

Метою вивчення навчальної дисципліни «Візуальні мови програмування» є засвоєння студентами основ створення програм для ЕОМ шляхом маніпулювання графічними об'єктами замість написання їх тексту (візуальна мова програмування Google Blockly).

Основними **завданнями** вивчення дисципліни «Візуальні мови програмування» є:

- оволодіти основами програмування у Google Blockly;
- набути навичок створення програм за допомогою візуальних мов програмування.

У результаті вивчення дисципліни студенти повинні досягти таких **результатів навчання:**

знати:

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

уміти: застосовувати візуальну мову програмування Google Blockly для написання стандартних програм, для розв'язання прикладних завдань, використовувати можливості та особливості візуальних мов програмування при створенні програмних продуктів.

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких **компетентностей**:

- здатність розв'язувати складні спеціалізовані завдання або практичні проблеми інженерії програмного забезпечення із застосуванням теорій та методів інформаційних технологій;
- здатність аналізувати предметні області, формулювати вимоги, ідентифікувати, класифікувати та описувати завдання, знаходити методи та підходи до їх розв'язання;
- здатність до самовдосконалення та самостійного професійного розвитку;
- здатність оцінювати та забезпечувати якість виконуваних робіт;
- здатність працювати незалежно і самостійно;
- здатність до логічного і алгоритмічного мислення, обґрунтування вибору методів розв'язання задач, інтерпретації отриманих результатів;
- здатність застосовувати набуті знання в практичних ситуаціях;
- здатність вільно спілкуватися державною мовою (усно та письмово).

Методичні рекомендації створені авторами на основі багаторічного досвіду викладання візуальних мов програмування студентам спеціальності «Інженерія програмного забезпечення». Автори сподіваються, що запропоноване видання стане корисним здобувачам вищої освіти, які прагнуть отримати знання та набути навичок із програмування, а також викладачам для проведення лабораторних робіт та організації самостійної роботи студентів.

ЗАГАЛЬНІ РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

Лабораторні заняття з дисципліни «Візуальні мови програмування» призначені для того, щоб студент під керівництвом викладача, особисто опрацював матеріал, вчився програмувати за допомогою візуальних мов програмування, застосовував візуальну мову програмування 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 опрацювати самостійно); виконати практичні завдання; оформити звіт з виконання лабораторної роботи, який повинен містити:

- титульний аркуш;
- тему роботи;
- хід виконання відповідного завдання;
- результати обчислень та аналіз отриманих розв'язків.

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: Введення у візуальні мови програмування

Мета: Ознайомитись з основними можливостями середовища Google Blockly, блоками, їх призначенням, зв'язком з іншими мовами програмування.

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.

Так чи інакше, код програми являє собою текст. 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. Взагалі-то причин створити візуальні мови програмування було дуже багато [1, 4, 5].




Корпорація Google створила свій варіант такої мови 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 перекладене майже 40 різними мовами, а програмування блоками інтуїтивно зрозуміле навіть дитині. Програмісту не треба писати код, йому треба лише розташувати на полі справа блоки у правильній взаємодії одне з одним. Середовище розроблене так, що написати «неправильну» програму майже неможливо, кожний блок можна розташувати лише на відведене йому місце.

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 елемента легшим.

Важливі поради:

- ✓ Для запуску програми натисніть ,
- ✓ Для збереження посилання на ваш код використайте піктограму .
- ✓  дозволяє видалити всі блоки з поля.
- ✓ Щоб видалити один блок, виділіть його та натисніть Del, або перетягніть до сміттового 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.

Завдання до лабораторної роботи

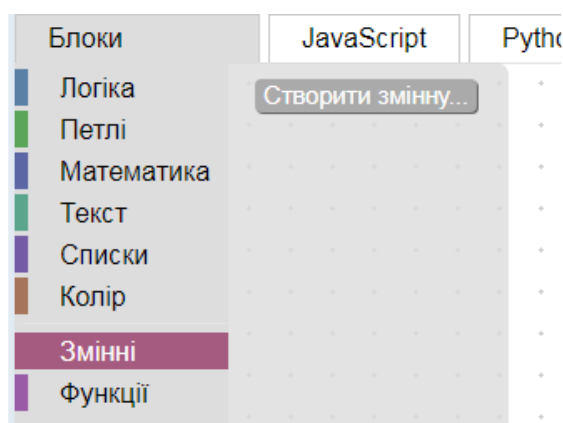
1) Починаємо роботу у візуальному середовищі (переходимо за посиланням), вивчаємо його (знайомимось) на зручній мові (укр., рос., англ.,...), яку обираємо угорі праворуч.

2) Вивчаємо типи і можливості блоків, натискаючи на них праву кнопку миші.

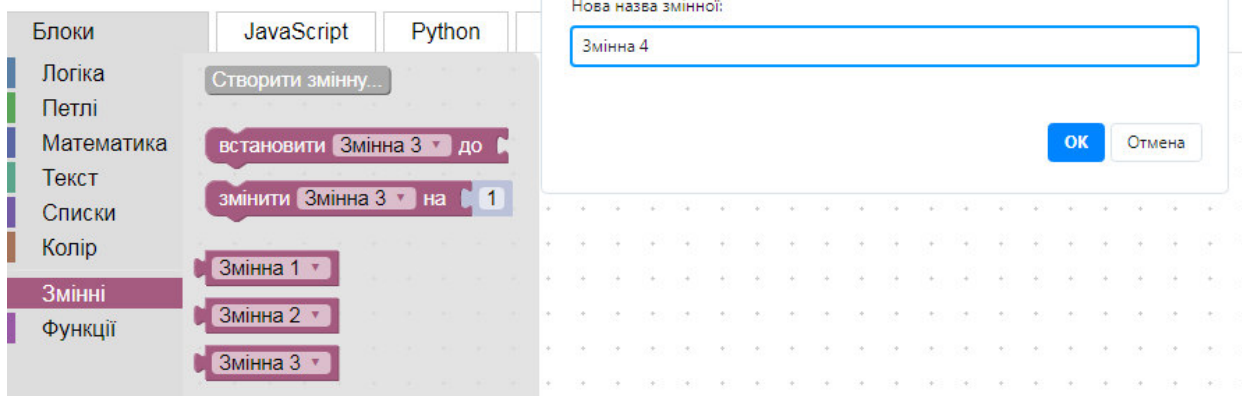
3) Виконуємо найпростіші дії за зразком:

- Створюємо змінні.

Для цього натискаємо на кнопку «Створити змінну» і надаємо їй ім'я (назву).

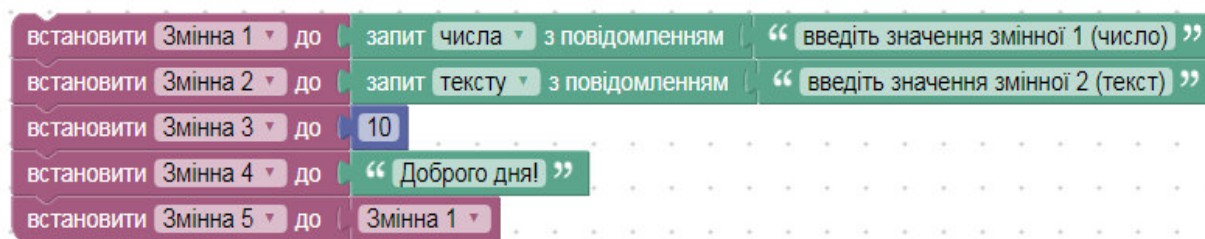


[Blockly](#) > [Demos](#) > Код

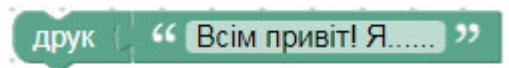


- Присвоюємо значення змінним різними способами.

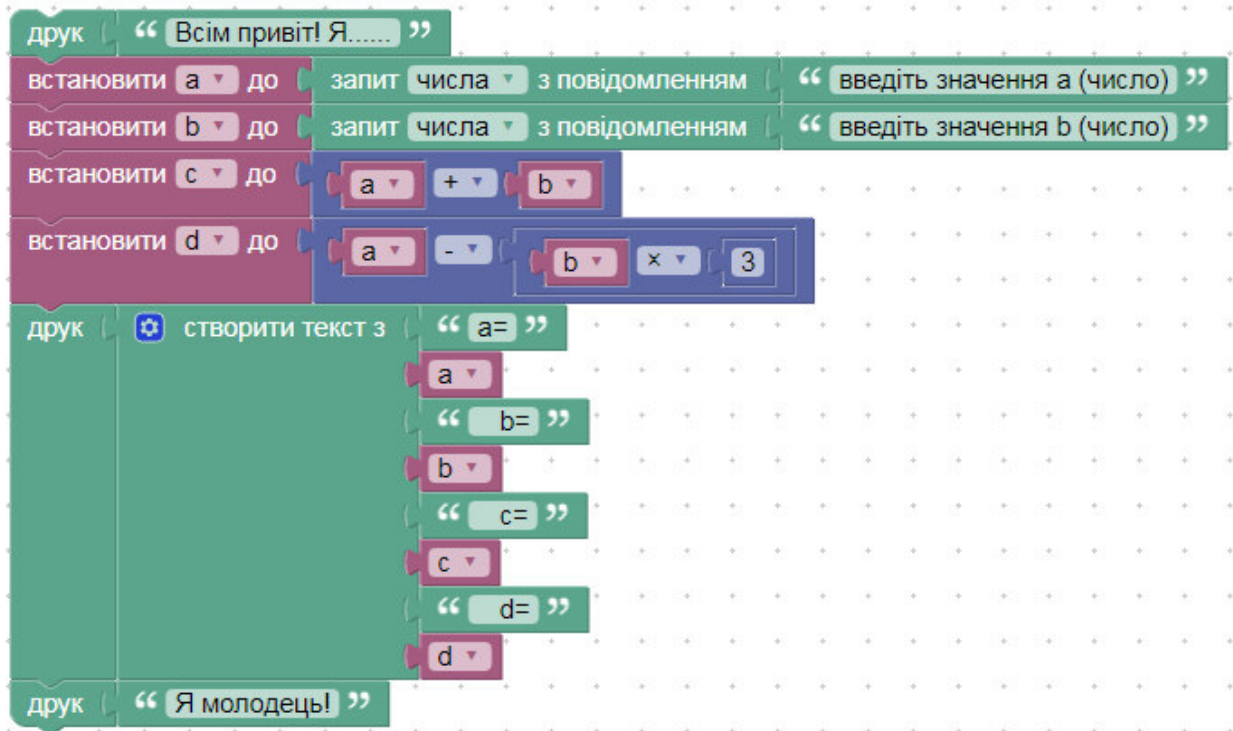
Зауважимо, що тип змінної буде визначатись значенням, яке їх присвоюємо.



- Друкуємо повідомлення про себе:



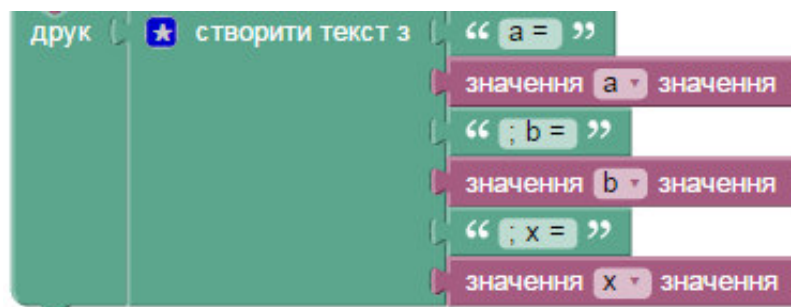
- Виконуємо дії зі змінними:



- Переглядаємо коди програм у різних мовах програмування.

? Питання для самоконтролю

1. Які переваги та недоліки візуальних мов програмування?
2. Які ви знаєте візуальні мови програмування?
3. На які типи розбито блоки в середовищі Google Blockly?
4. Як створити змінну в середовищі Google Blockly?
5. Як присвоїти значення змінним в середовищі Google Blockly?
6. Поясніть частину програми. Що є результатом її виконання?



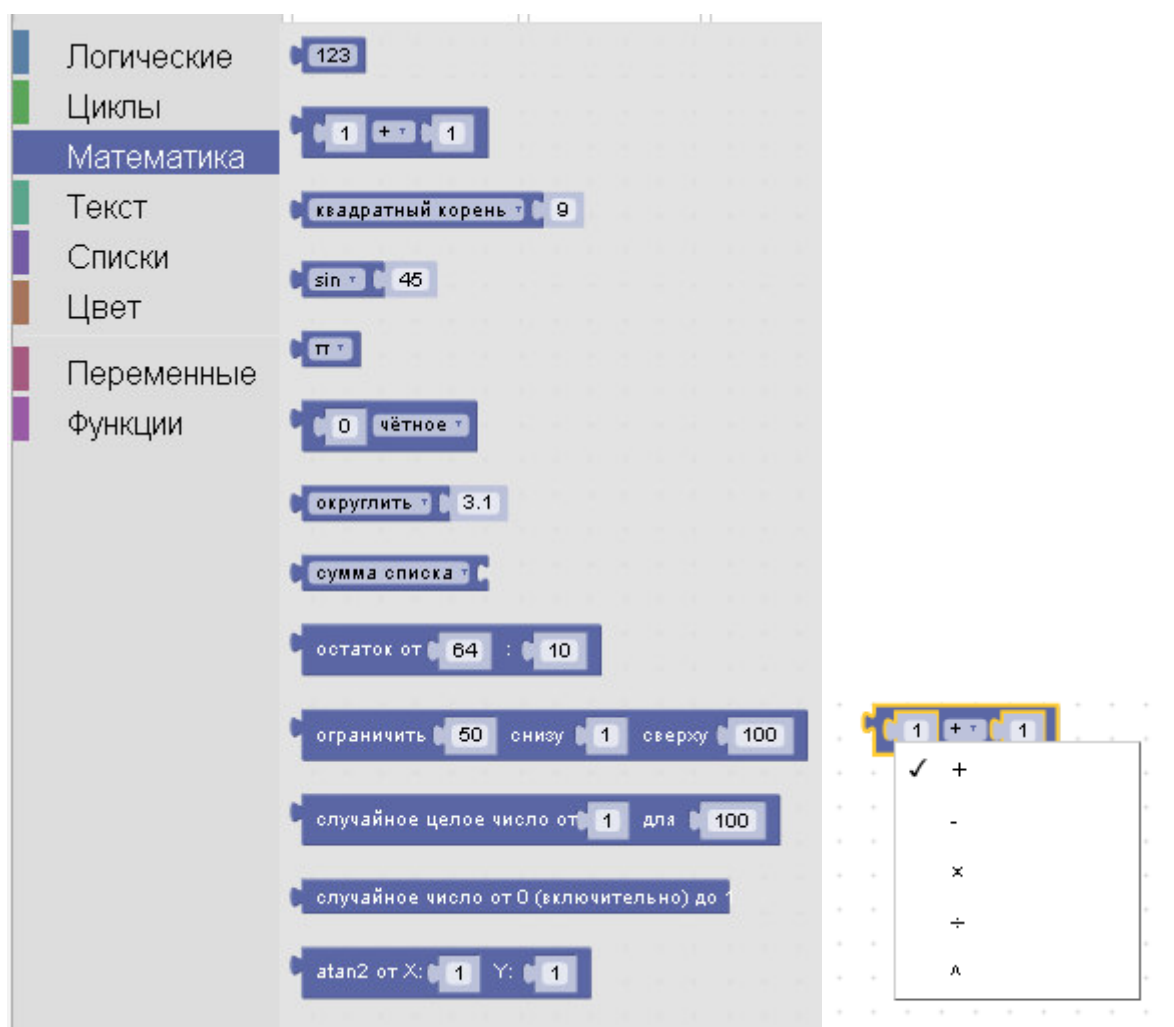
Лабораторна робота №2

Тема: Каталог Math. Математичні дії, функції

Мета: Ознайомитись з можливостями використання математичних операторів, констант та функцій каталогу Math візуального середовища розробки Google Blockly при створенні програм.

Теоретичні відомості

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

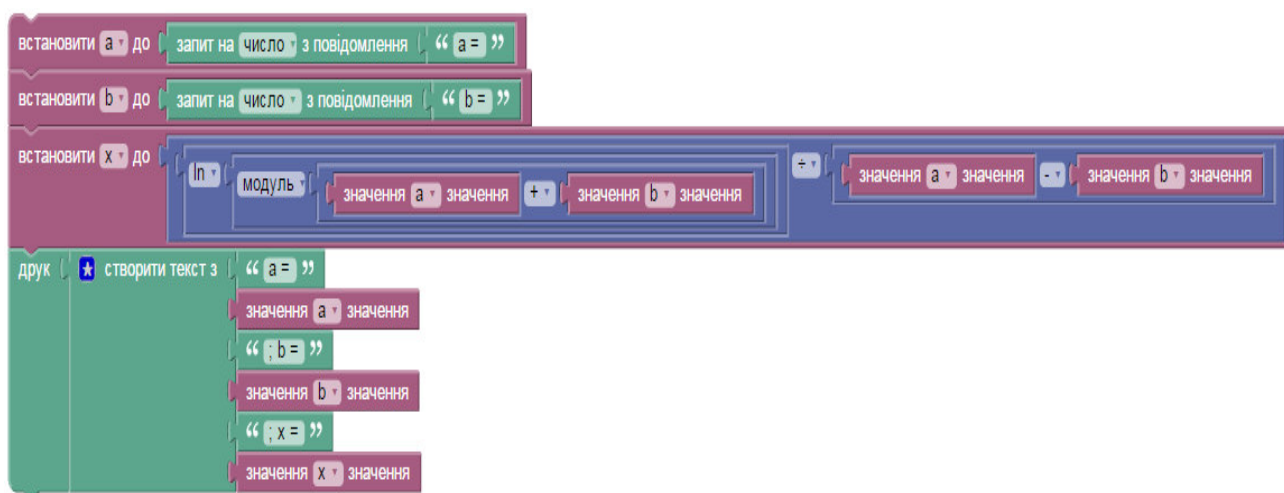


Також реалізовано деякі специфічні перевірки чи дії, які не містяться у класичних математичних бібліотеках мов програмування. Наприклад, програмісту надається можливість в одну дію перевірити число на парність/непарність, додатність/від'ємність, чи воно просте, ціле або ділиться на інше число.

Візуальне середовище розробки Google Blockly надає такі математичні константи як π , e , ∞ , φ та ін. Реалізовані тригонометричні функції та робота з логарифмами і багато інших математичних дій. Є функції, що дозволяють округляти числа до найближчого цілого, створювати псевдовипадкові числа та навіть дроби.

Особливістю роботи з візуальним середовищем розробки Google Blockly є те, що в кожній дії можуть приймати участь не більше ніж 2 змінні, тому складні формули доводиться розбивати на кроки. Для того, щоб розбити будь-яку формулу на кроки, слід проаналізувати порядок дій і скласти вираз з модулів таким чином, щоб зовні опинились дії, які повинні бути виконані в останню чергу.

Приклад 2.1. Обчислимо значення виразу $x = \frac{\ln|a+b|}{a-b}$



Результат:

a = 10; b = 20; x = -0.34011973816621555

Завдання до лабораторної роботи

- Обчисліть значення виразу:

$$A = \frac{\sqrt[3]{\sin^2 45^\circ - 3.14 + \ln 147} \cdot (2^{1.5} - [48,2])}{8! + \sqrt{100} - \frac{\pi}{3}}$$

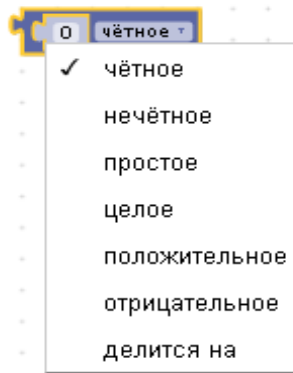
- Виконайте завдання свого варіанту:

1. Задано координати по вісі x та y двох точок. Обчисліть відстань між ними та відстань від кожної з точок до початку координат.
2. Задано координати по вісі x, y та z двох точок. Обчисліть відстань між точками та відстань від кожної точки до початку координат.

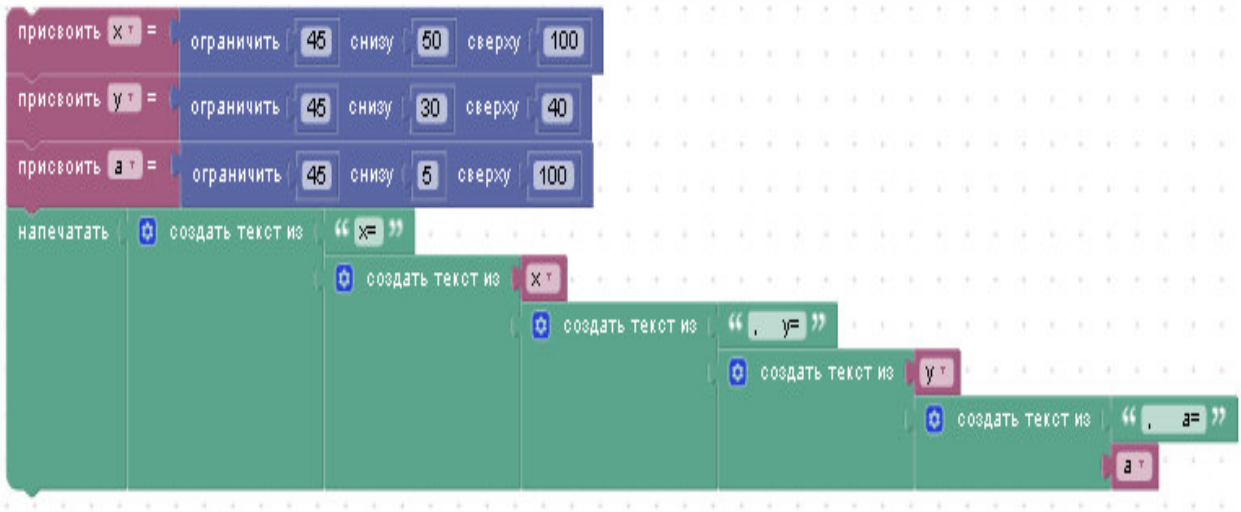
3. Задано радіус кола та міра центрального кута. Обчисліть довжину кола й площу круга, площу сектора та сегмента, довжину сторін вписаного та описаного в коло правильного трикутника.
4. Відома довжина сторони правильного трикутника. Обчисліть його периметр, площу, висоту, радіуси вписаного та описаного кіл.
5. 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.
6. Задано дві сторони трикутника та кут між ними. Обчисліть міри його невідомих кутів та сторони, периметр та площу фігури.
7. Написати програму для розв'язку квадратного рівняння $ax^2 + bx + c = 0$. Виконати перевірку коректності коефіцієнтів.
8. Обчисліть об'єм та площу повної поверхні циліндра та конуса за відомими радіусом та висотою.
9. Вектор має свій початок в точці з координатами (0;0), а координати кінця вказує користувач. Обчисліть довжину вектора та кути, які він утворює з осями координат.
10. Задано відстань, швидкість течії та власну швидкість катера. Обчисліть скільки часу знадобиться катеру, щоб пройти задану користувачем відстань за течією та проти неї.
11. Відома довжина ребра куба. Обчисліть площу його поверхні, об'єм, а також радіуси, об'єми та площі поверхонь вписаної та описаної куль.
12. Відоме значення $\operatorname{tg} \frac{x}{2}$. За допомогою формул тригонометрії знайти $\sin x$, $\cos x$, $\operatorname{tg} x$, $\operatorname{ctg} x$, $\operatorname{sec} x$, $\operatorname{cosec} x$.
13. Відоме значення $\sin x$. За його допомогою обчисліть $|\cos x|$, $|\operatorname{tg} x|$, $|\operatorname{ctg} x|$. Реалізуйте перевірку коректності значення сінуса.
14. Відоме значення $\cos x$. За його допомогою обчисліть $|\sin x|$, $|\operatorname{tg} x|$, $|\operatorname{ctg} x|$. Реалізуйте перевірку коректності значення косінуса.
15. Відомий радіус описаного навколо правильного семикутника кола. Обчисліть, використовуючи лише це значення, довжину сторони, периметр та площу семикутника.

? Питання для самоконтролю

1. Які особливості математичних операторів у Google Blockly ви знаєте?
2. Які математичні функції реалізовано в середовищі Google Blockly?
3. Поясніть що є результатом виконання кожної дії:



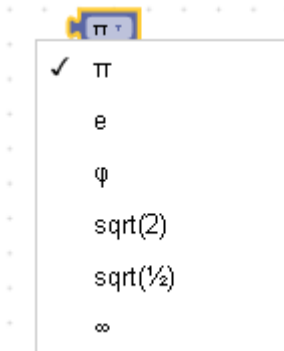
4. Поясніть, що є результатом виконання програми:



5. Поясніть, що є результатом виконання операції:



6. Поясніть значення кожної з констант:



7. Поясніть, що є результатом виконання програми:



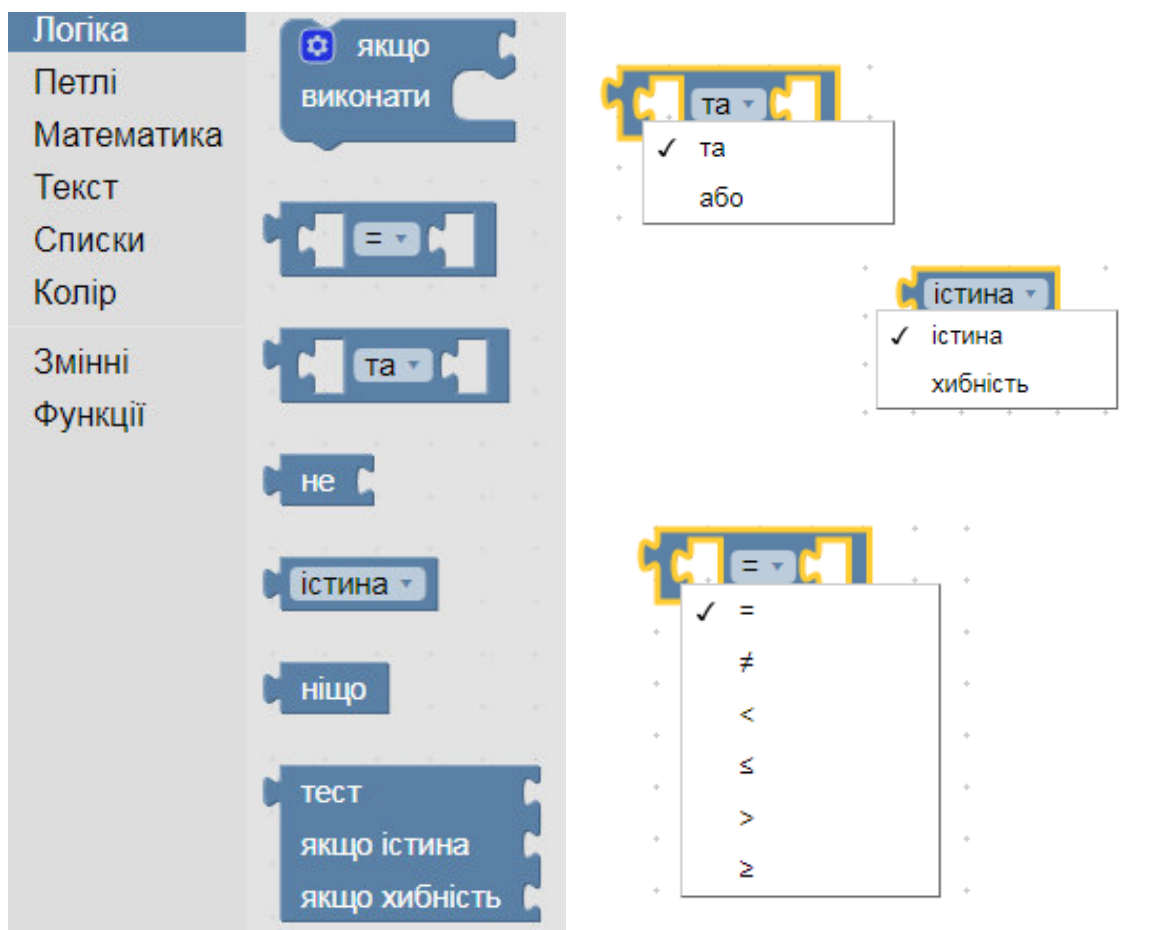
Лабораторна робота №3

Тема: Каталог Logic. Оператори порівняння

Мета: Ознайомитись з особливостями логічних блоків. Навчитись використовувати оператори порівняння для створення програм.

Теоретичні відомості

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.

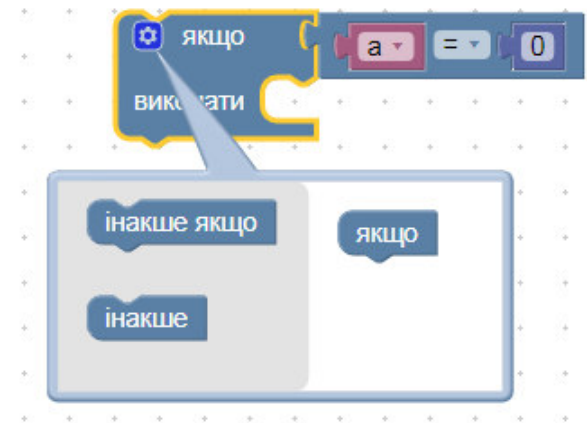


Слід звернути увагу, що в каталозі лежить спрощена версія оператора ЯКЩО ... ТО. Щоб використовувати повну і навіть комплексну версію цього оператора натисніть на зірочку зліва вгорі і додайте стільки пунктів ІНАКШЕ ЯКЩО і ІНАКШЕ, скільки вам необхідно.

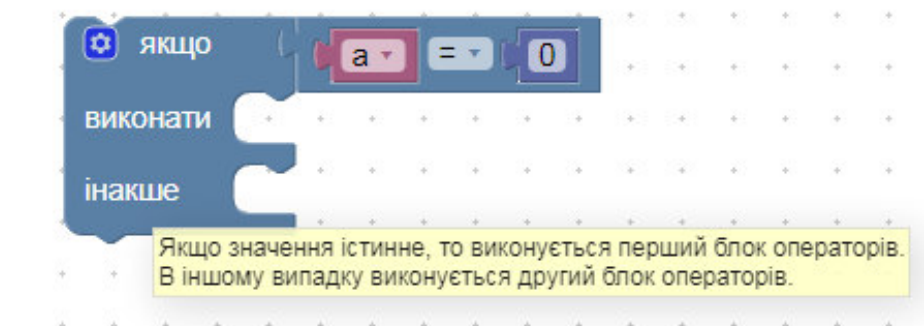
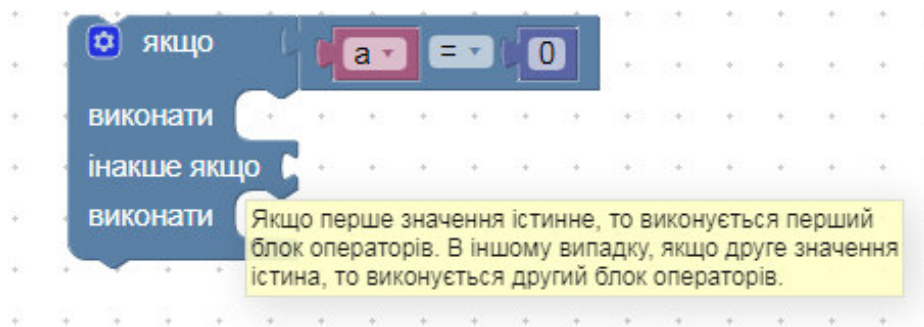
Стандартний варіант:



Вибір:

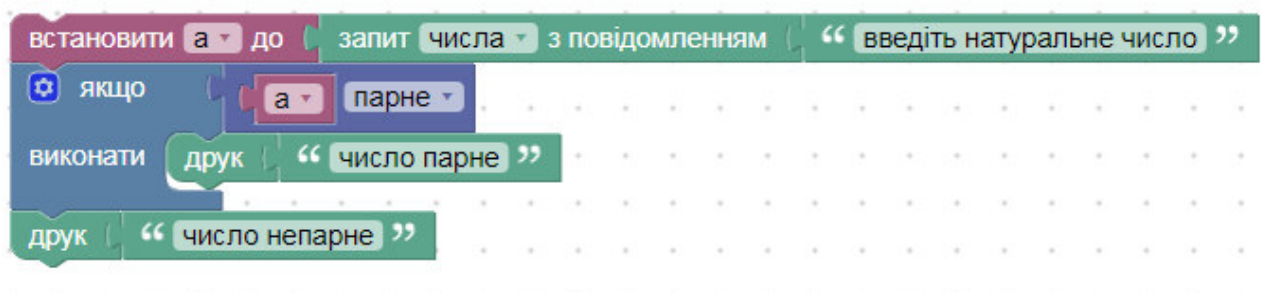


Результат:



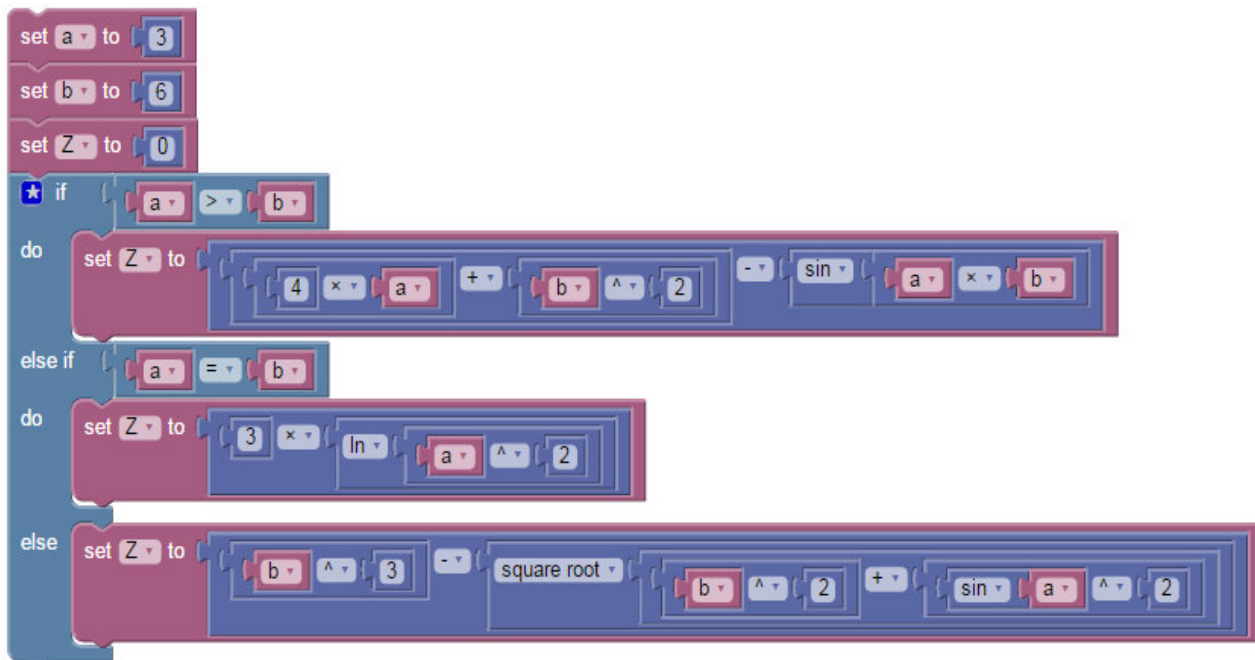
Приклади програм

Приклад 3.1. Перевірити на парність введене користувачем натуральне число.



Приклад 3.2. Обчисліть значення виразу:

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



Завдання до лабораторної роботи

- Обчисліть значення функції:

$$f(x) = \begin{cases} \cos x, & x \leq -4; \\ -\sin x, & -4 < x \leq 0; \\ \sqrt{x+2}, & 0 < x < 10; \\ 1/2^x, & 10 \leq x \leq 20; \\ x^2, & x > 20. \end{cases}$$

- Виконайте завдання свого варіанту:

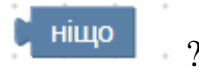
1. Відомі координати точки на осях x та y. В наступній послідовності перевірте виконання умов: чи лежить точка на перетині осей x та y; чи лежить точка на одній з координат та на якій; чи лежить точка в одному з чотирьох квадрантів та в якому.
2. Відомі координати точки x, y та z. В наступній послідовності перевірте виконання умов: чи лежить точка на перетині трьох осей; чи лежить вона на одній з осей та на якій. При невиконанні двох умов розрахуйте, відстань до якої з осей найменша.

3. Відомі три числа. Перевірте, чи є вони послідовними членами арифметичної чи геометричної прогресії. Виведіть на друк тип прогресії та її крок.
4. Відомі три числа. В наступній послідовності перевірте виконання умов: чи дорівнюють всі числа одне одному; чи є серед чисел два рівних і які. У випадку невиконання обох умов виведіть на друк відповідне повідомлення.
5. Відомі три числа. Перевірити, чи є друге за значенням (не за порядком вводу) серед них середнім арифметичним двох інших.
6. 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.
7. Написати програму для розв'язання наступної задачі: визначити, чи пройде цеглина розміром $X \times Y \times Z$ в прямокутне віконце розміром $A \times B$.
8. 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.
9. Відомий номер місяця та дня. Визначити, скільки днів усього в цьому місяці. Врахувати фактор високосного року під час розрахунку лютого.
10. Вводять номер і день місяця. Перевірити дані, що вводяться на коректність. Врахувати, що в різних місяцях різна кількість днів. Вважається, що рік є не високосним.
11. Відомі довжини трьох сторін. Перевірити, чи можна з них побудувати трикутник. Додатково перевірити, чи не є цей трикутник рівностороннім чи рівнобедреним.
12. Відомі три кути. Перевірити, чи існує трикутник з такими кутами. Додатково перевірити, чи не буде він правильним, рівнобедреним чи прямокутним.
13. Користувач вводить порядковий день тижня. Програма повинна вивести на друк інформацію: чи є цей день буднім, або вихідним. Перевірити коректність вводу.
14. Написати програму-тест з такими п'ятьма питаннями, щоб відповідями були числа. Програма повинна перевірити вірність відповіді користувача і в кінці вивести його оцінку.
15. Відомі координати лівого нижнього та правого верхнього кутів прямокутника, а також координати точки. Визначити, чи розташована точка на одній зі сторін прямокутника, між ними чи поза фігурою.

? Питання для самоконтролю

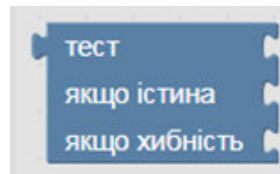
1. Які значення може приймати логічна змінна?
2. Які операції порівняння ви знаєте?
3. Наведіть приклади операторів (блоків), результатом виконання яких є істина або хибність.
4. Які оператори (блоки) реалізують логічні І, НЕ та АБО?

5. В яких випадках використовують блок



?

6. В яких випадках використовують блок



?

Лабораторна робота №4

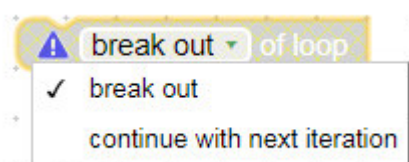
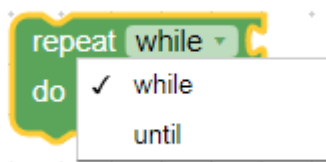
Тема: Каталог Control. Елементи управління циклами

Мета: Навчитись створювати циклічні програми у візуальному середовищі розробки Google Blockly.

Теоретичні відомості

В українському перекладі відповідний каталог був названий «Петлі», і в ньому можна знайти чотири види циклів:

1. Зі сталою кількістю повторень.
2. Цикл з передумовою.
3. Цикл по змінній.
4. Спеціальній цикл по змінній для роботи зі списками.



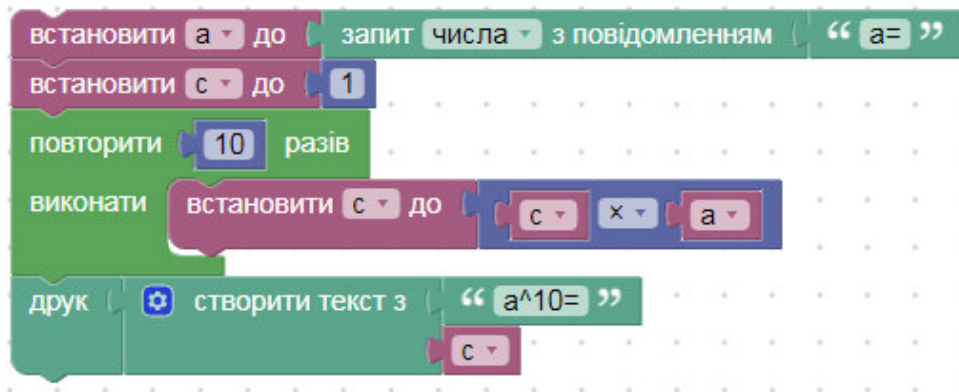
Якщо виникає необхідність роботи з циклом з післяумовою, слід зробити нескінченний цикл з умовою виходу з нього (вона також є у каталозі).

Як і в інших мовах програмування, ітератор циклу повинен бути перерахованим типом, а саме – цілим числом.

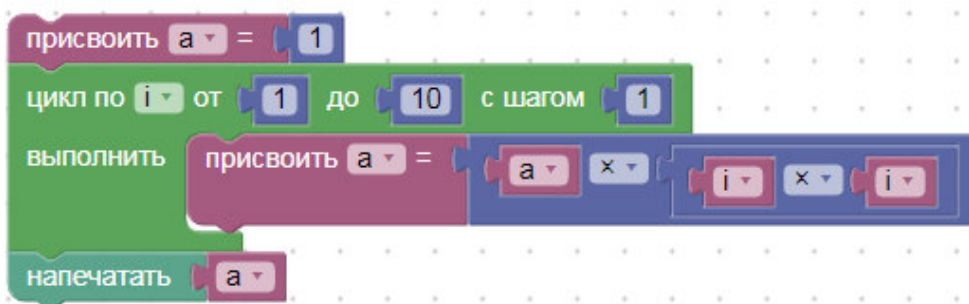
Цикли можна включати одне в одного.

Приклади програм

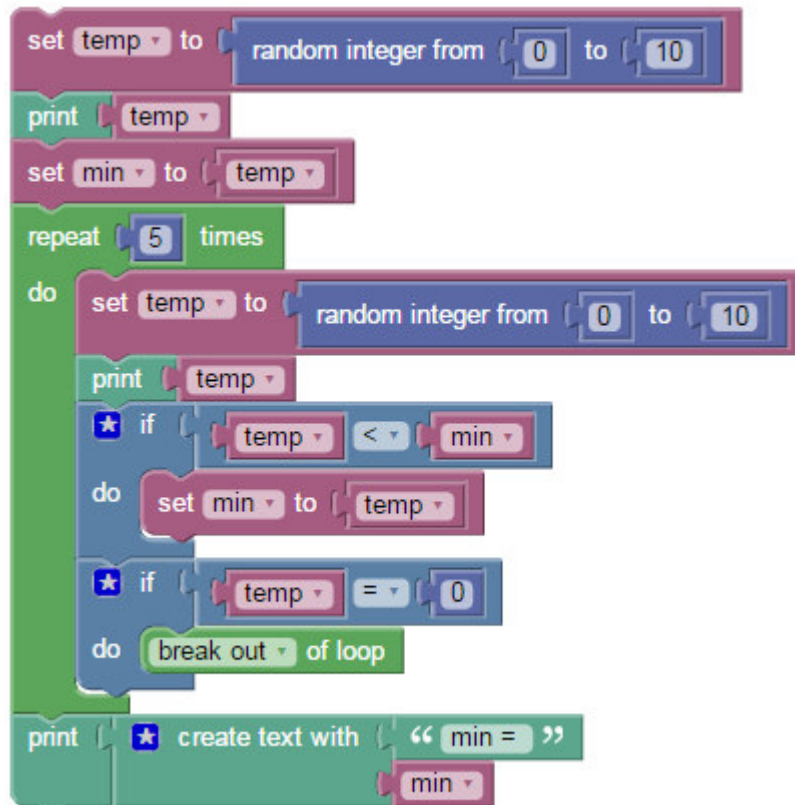
Приклад 4.1. Обчислення степеня числа.



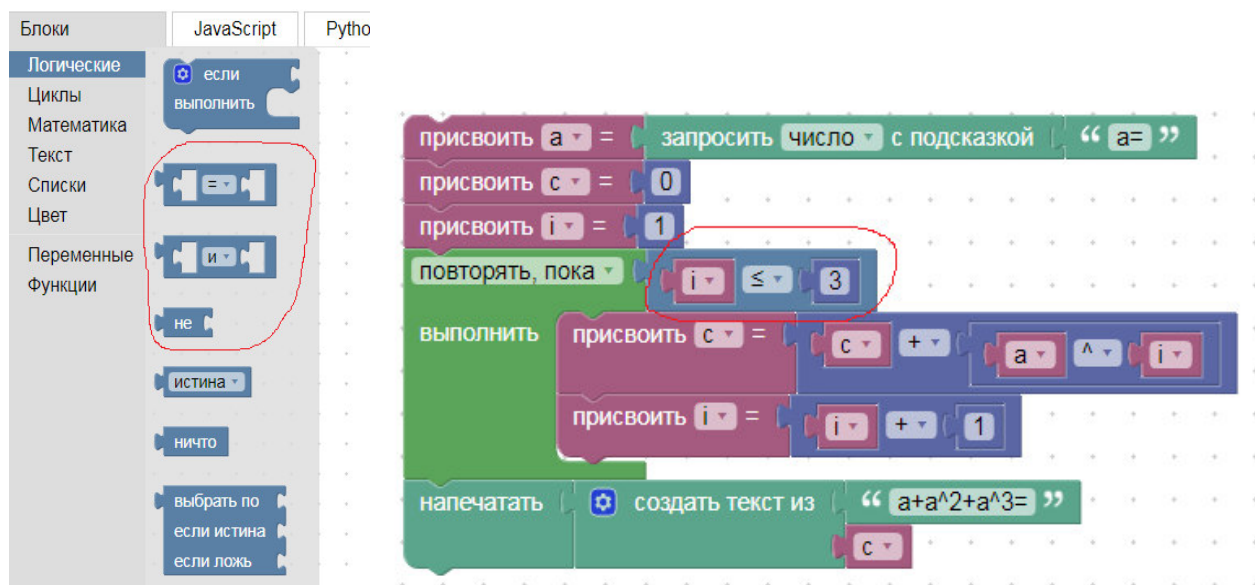
Приклад 4.2. Обчислення квадрата факторіала.



Приклад 4.3. Знаходження найменшого з чисел.



В циклах з умовою слід задавати умову за допомогою логічних блоків.



Завдання до лабораторної роботи

Слід виконати по одному завданню з кожної частини.

Частина 1. Цикли (відома кількість повторень)

1. Відомі два натуральних числа x та y ($x < y$). Вивести на друк суму простих чисел в діапазоні від x до y .
2. Написати програму для розрахунку факторіалу довільного числа.
3. Знайти найбільший спільний дільник двох натуральних чисел.
4. Відоме натуральне число N , порахувати $1 + 2 + \dots + N$.
5. Відоме натуральне число N , порахувати 2^N .
6. Написати програму для розв'язку наступної задачі: клітина ділиться на дві кожні три години. Визначити, скільки клітин стане через N годин.
7. З клавіатури вводиться кількість студентів, а потім їх оцінки з математичного аналізу напередодні екзаменів. Не використовуючи масивів розрахуйте їх середній бал.
8. З клавіатури вводиться кількість студентів, а потім їх оцінки з математичного аналізу напередодні екзаменів. Не використовуючи масивів розрахуйте кількість студентів, які не отримали допуск (< 50 балів).
9. Відома кількість курсів на спеціальності і кількість студентів на кожному з них. Не використовуючи масив визначити курси, на яких найбільша й найменша кількість студентів.

10. Відома кількість курсів на спеціальності і кількість студентів на кожному з них. Не використовуючи масив визначити середню кількість студентів на курсі.
11. Знайти добуток двох чисел, використовуючи лише додавання.
12. Написати програму для піднесення до квадрату натурального числа, використовуючи закономірність:

$$1^2 = 1$$

$$2^2 = 1 + 3$$

$$3^2 = 1 + 3 + 5$$

$$\dots$$

$$n^2 = 1 + 3 + 5 + \dots + 2n - 1.$$
13. Використовуючи дві допоміжні змінні, але не використовуючи масив, знайти N-е число послідовності Фібоначчі.
14. Дано два натуральних числа a і b. Знайти суму всіх кратних 3 чисел, які належать проміжку [a; b].
15. Знайти кількість «щасливих» 4-значних квитків. Квиток вважається «щасливим», якщо сума двох перших та двох останніх цифр співпадає. За умовою номери квитків змінюються від 1000 до 9999.

Частина 2. Цикли (невідома кількість повторень)

1. Знайти найменше просте трьохзначне число.
2. Знайти найбільше просте двозначне число.
3. Написати програму для розв'язку наступної задачі: сумувати числа, що вводяться з клавіатури, доки не зустрінеться нуль.
4. Знайти найменше число, факторіал якого є 5-значним числом.
5. Знайти найбільше число, факторіал якого є 4-значним числом.
6. Клітина ділиться на дві кожні три години. Визначити, через який проміжок часу кількість клітин сягне числа N, яке вводиться з клавіатури.
7. Знайти найбільше двозначне число, яке було б кратним сумі своїх цифр.
8. Знайти найменше тризначне число, яке було б кратним сумі своїх цифр.
9. Вивести на друк перші 20 простих чисел.
10. Сумувати числа, що вводяться з клавіатури, доки не зустрінеться два однакових числа поспіль.
11. 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 100 чи більше доларів.
12. В перший день спортсмен пробіг 10 км, кожного наступного дня він збільшував відстань на 10%. На який день спортсмен пробіжить 300 км?

13. На кожній парі фізкультури студент отримує 2 бали за присутність, кожні 5 пар студент проводить розминку і додатково отримує 3 бали, а кожні 9 пар проходить залікова здача нормативів, яка приносить додаткові 5 балів. Скільки пар треба відвідати студенту, щоб заробити 75 балів?
14. Знайти найменше автоморфне число, яке більше за N.
15. Знайти перше число послідовності Фібоначчі, яке більше за N.

? Питання для самоконтролю

1. Що таке цикли? Які види циклів ви знаєте?
2. Що таке нескінченний цикл? Яким чином його можна позбутись?
3. Поясніть частину програми. Що є результатом її виконання?

```

присвоить a = 1
цикл по i от 1 до 10 с шагом 1
  выполнить
    присвоить a = a * i
напечатать a
  
```

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

```

присвоить i = 1
присвоить a = 1
повторить 10 раз
  выполнить
    присвоить a = a * i
    присвоить i = i + 1
напечатать a
  
```

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

```

присвоить a = 0
цикл по i от 1 до 10 с шагом 1
  выполнить
    присвоить a = a + i
напечатать a
  
```

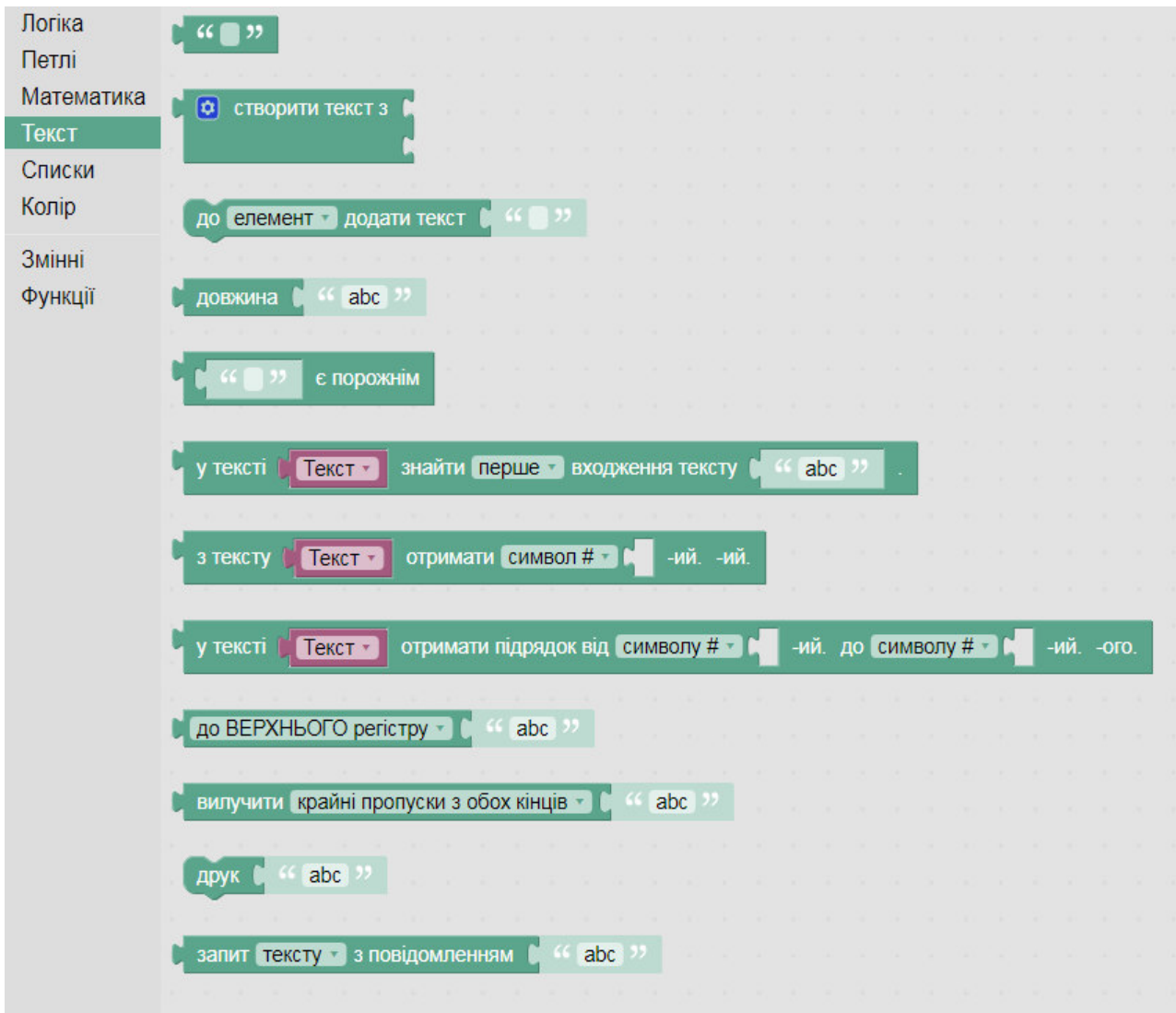

Лабораторна робота №5

Тема: Каталог Text. Робота з текстами та символічними змінними

Мета: Опанувати навички роботи з текстами та символічними змінними у Google Blockly.

Теоретичні відомості

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



Ви вже використовували декілька команд з цього каталогу для друку повідомлень, але можливості Google Blockly трохи більші. В одну дію ви можете 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, тож вам доведеться для збереження результатів

завести новий рядок. Ви можете шукати входження одного рядка до іншого, вибрати n-й елемент або цілий підрядок.

Приклади програм

Приклад 5.1. Створення текстової змінної.

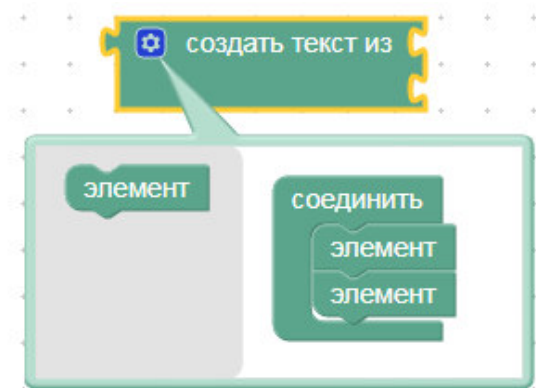
1 спосіб:



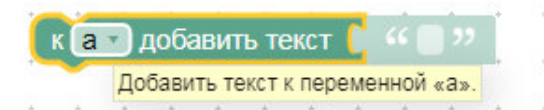
2 спосіб:



3 спосіб:



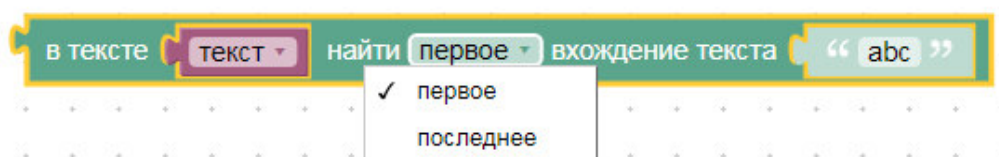
4 спосіб:



Приклад 5.2. Друк текстової змінної (тексту).

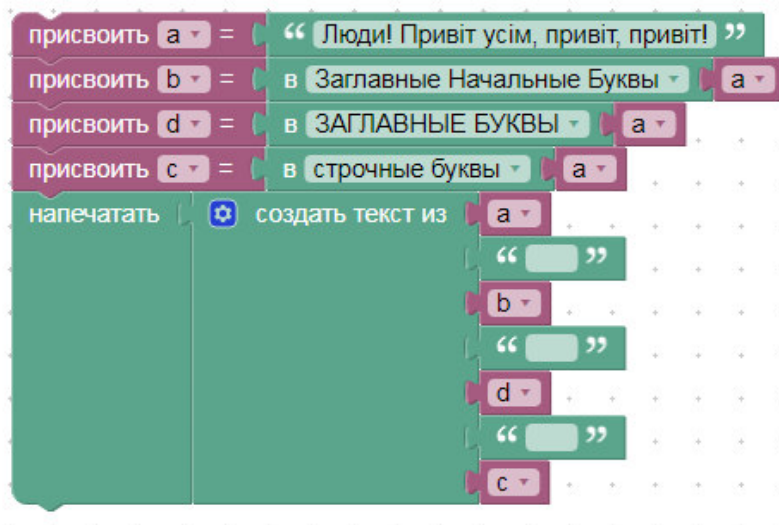


Приклад 5.3. Знаходження позиції підтексту у тексті.



Значення дорівнює нулю, якщо текст не знайдено.

Приклад 5.4. Зміна регістру.



Результат:

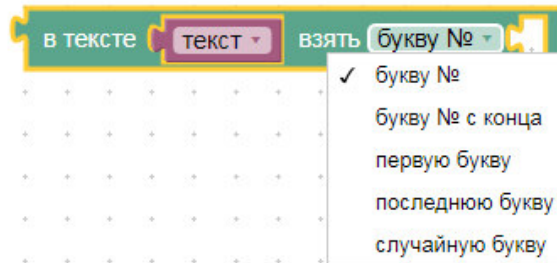
Люди! Привіт усім, привіт, привіт! Люди! Привіт Усім, Привіт, Привіт!
ЛЮДИ! ПРИВІТ УСІМ, ПРИВІТ, ПРИВІТ! люди! привіт усім, привіт, привіт!

Приклад 5.5. Робота з окремими літерами та групами літер.

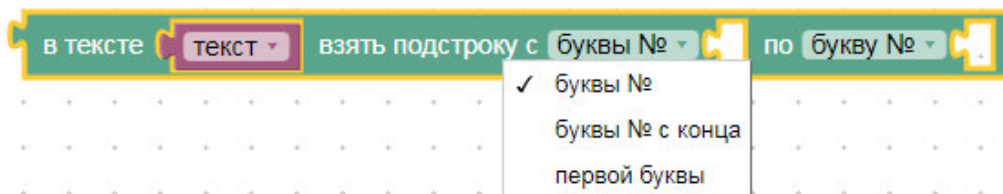
Знаходження довжини:



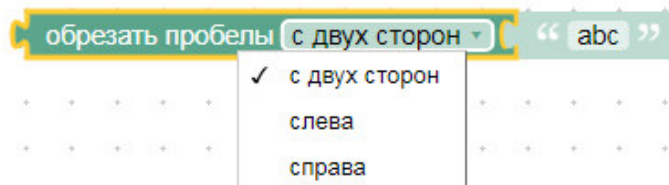
Вибір літери:



Вибір частини тексту:



Прибирання зайвих пропусків:



Приклад 5.6. Заміна символів.

```
присвоить a = " Люди! Привіт усім, привіт, привіт! "  
присвоить n = длина a  
присвоить b = ""  
цикл по i от 1 до n с шагом 1  
  выполнить  
    присвоить f = в тексте a взять букву № i  
    если f = " i "  
      выполнить к b добавить текст "*"   
    иначе к b добавить текст f  
напечатать создать текст из a  
  " "  
  b
```

Результат:

Люди! Привіт усім, привіт, привіт! Люди! Прив*т ус*m, прив*t, прив*t!

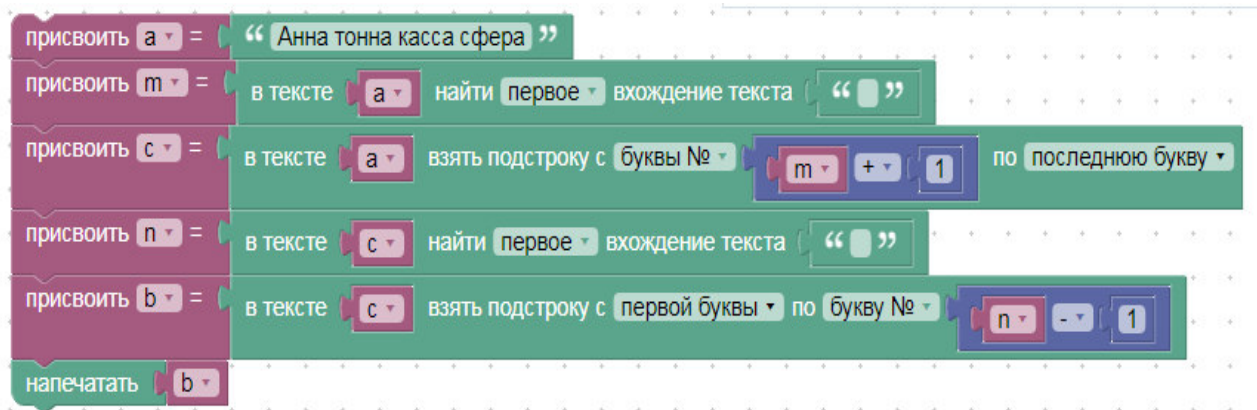
Приклад 5.7. Заміна подвоєних букв на одну.

```
присвоить a = " Анна, тонна, касса "  
присвоить n = длина a  
присвоить b = ""  
цикл по i от 1 до n с шагом 1  
  выполнить  
    присвоить f = в тексте a взять букву № i  
    присвоить c = в тексте a взять букву № i + 1  
    если f = c  
      выполнить к b добавить текст f  
      присвоить i = i + 1  
    иначе к b добавить текст f  
напечатать b
```

Результат:

Ана, тона, каса

Приклад 5.8. Знайти друге слово в тексті, якщо слова в тексті відокремлені пропусками.



Результат:

тонна

Завдання до лабораторної роботи

- Задано текст:

Математика – царица наук (Гаусс). Математика – это выдумка без обмана (Генденштейн). Математика ум в порядок приводит (Ломоносов). Математика – это искусство называть разные вещи одним и тем же именем (Пуанкаре). Математика – это язык природы (Гиббс).

Виконайте наступні завдання:

- 1). Порахувати скільки разів у ньому зустрічається слово Математика?
- 2). Видалити з тексту всі слова, що стоять у дужках разом із дужками.
- 3). Замінити знаки – на ***.
- 4). Записати слова Математика великими літерами.
- 5). Порахувати кількість слів у тексті.

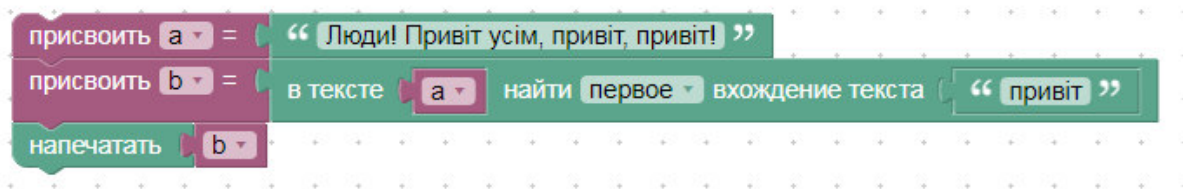
- Виконайте завдання свого варіанту. Доберіть такий вхідний текст, щоб він відповідав проблематиці завдання. Виведіть текст на друк до і після внесення змін.

1. Замінити на пропуски всі цифри в тексті.
2. Замінити на знак зірочка (*) в тексті слово, введене з клавіатури.
3. Видалити в кожному слові останню літеру.
4. Видалити зайві пропуски в тексті.
5. Перевірте, чи однакові кількість лівих(<) та правих(>) кутових дужок в тексті.
6. Порахуйте кількість розділових знаків в тексті.
7. Порахуйте кількість пар однакових знаків, що стоять в тексті поруч. (AA – одна пара, AAA – 2 пари).

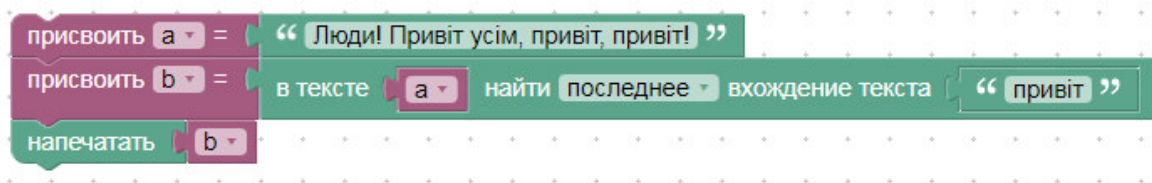
8. Видалить в тексті слова, що повторюються. (Якщо між ними немає розділового знаку).
9. Зробіть всі символи між лапками великими. («Гарі Потер» — «ГАРІ ПОТЕР»).
10. Замініть на знак серця всі букви o/O в тексті.
11. Перевірити, чи є фраза паліндромом.
12. Порахувати кількість слів в тексті.
13. Порахувати середню довжину слова в тексті.
14. Текст являє собою математичне рівняння, в якому використовуються літеральні представлення математичних констант (π , e і т.д.). Замінити їх на відповідні числа (точність не важлива).
15. Кожне слово тексту виведіть на друк в окремому повідомленні.

? Питання для самоконтролю

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?



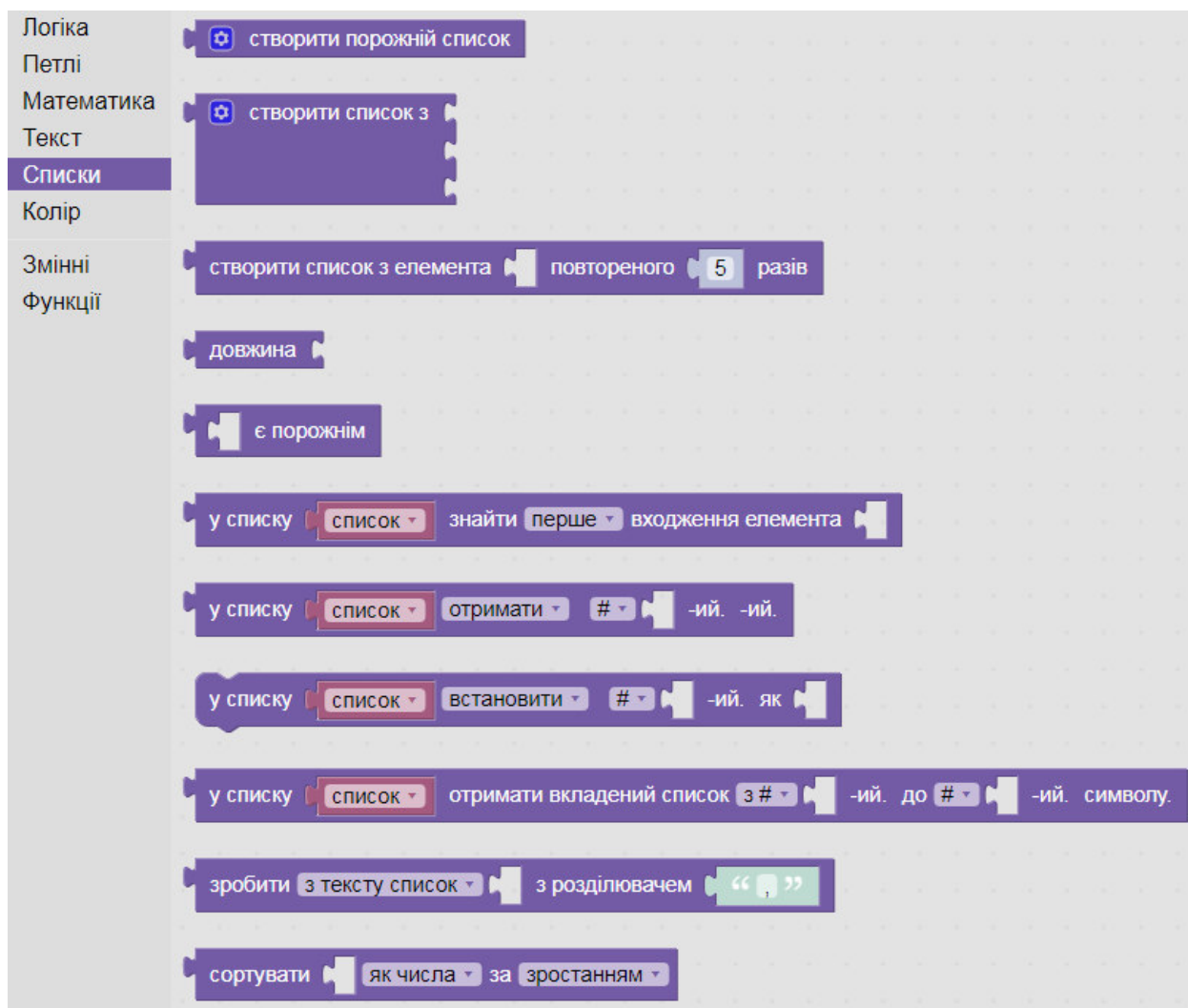
Лабораторна робота №6

Тема: Каталог Lists. Елементи для роботи з масивами

Мета: Ознайомитись з можливостями створення масивів різних типів даних та роботою з ними (звернення до елементів, друк елементів та ін.). Навчитись реалізовувати алгоритми сортування інформації (числових масивів та символьних масивів) у Google Blockly, розглянути можливості їх використання.

Теоретичні відомості

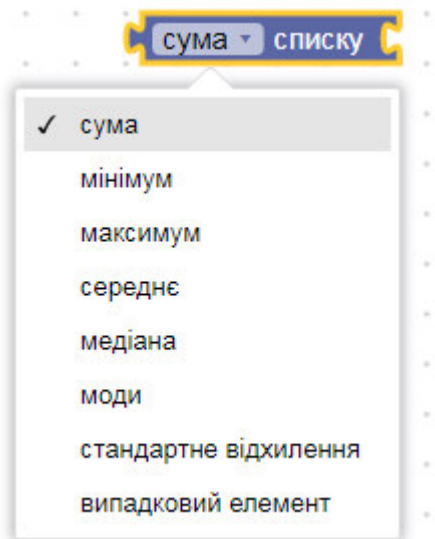
Google Blockly дозволяє створювати списки з будь-яких однорідних елементів, в тому числі з чисел, символів та кольорів. Вам треба лише привласнити змінній значення списку (пустого, або попередньо заповненого однаковими елементами). Відповідний каталог має вигляд:



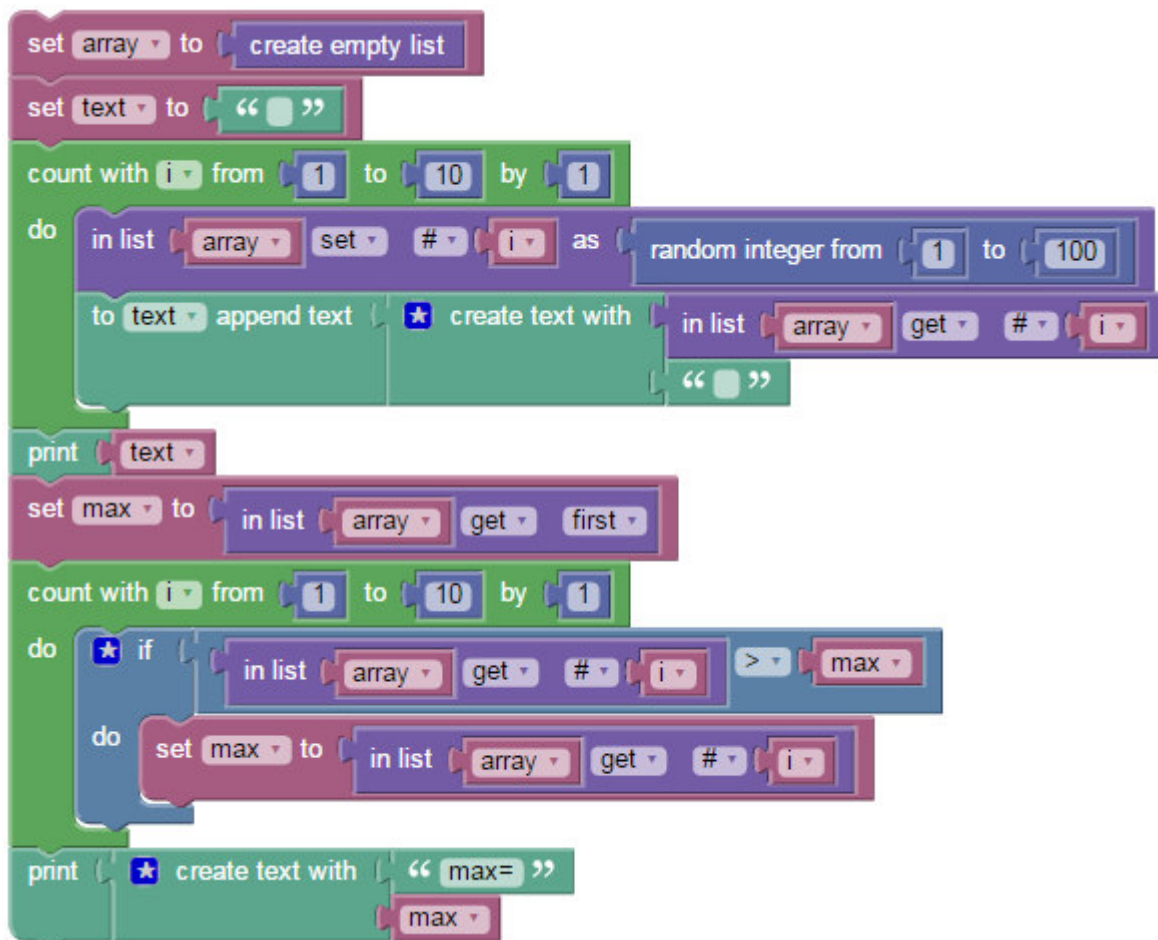
Надалі робота зі списками мало чим відрізняється від роботи з масивами у будь-якій іншій мові програмування.

Каталог «Списки» надає нам можливість в одну дію дізнатися чи є список порожнім, яку він має довжину, знайти входження певного елементу чи виділити підсписок. Ви можете 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 каталозі Math.



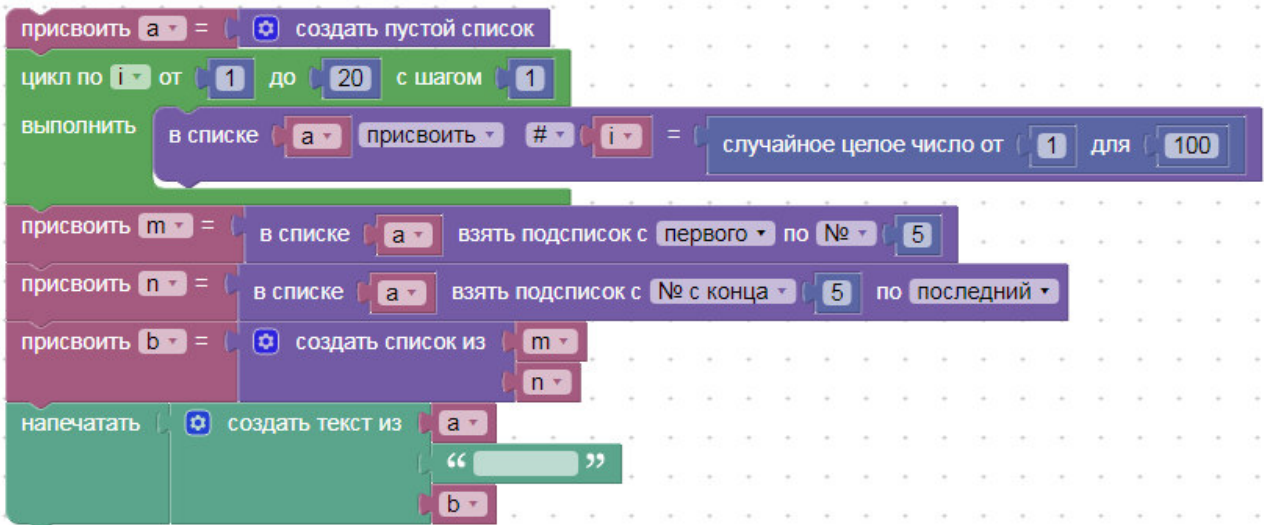
Приклад 6.1. Знаходження найбільшого елементу в списку випадкових цілих чисел.



Завдання до лабораторної роботи

- Виконайте завдання за зразком.

Створити масив з 5-ти перших і 5-ти останніх елементів масиву а.



Відсортувати слова тексту по алфавіту (порівняти результати різних видів сортування).

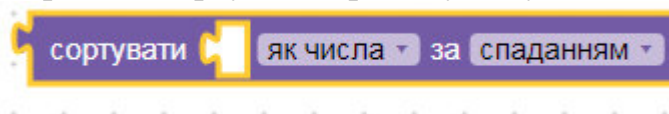


- Створіть масив з 20 випадкових цілих чисел. Виведіть його на друк до та після внесення змін. При виконанні роботи доцільно ознайомитись з матеріалом Додатку А. Виконайте завдання свого варіанту:

1. Відсортуйте масив «бульбашкою» за зростанням.
2. Відсортуйте масив «бульбашкою» за зменшенням.
3. Відсортуйте парні елементи масиву «бульбашкою».
4. Відсортуйте масив так, щоб всі елементи, які дорівнюють 0, опинилися в кінці, а інші елементи не змінили порядку розташування відносно одне одного.
5. Відсортуйте масив методом швидкого сортування.
6. Для впорядкованого масиву, що зростає, реалізуйте алгоритм бінарного пошуку.
7. Для впорядкованого масиву, що зменшується, реалізуйте алгоритм бінарного пошуку.
8. Відсортуйте масив методом сортування вставками.
9. Відсортуйте масив методом сортування вибором.
10. Відсортуйте масив методом Шелла за зростанням.
11. Відсортуйте масив методом Шелла за зменшенням.
12. Відсортуйте масив методом пірамідального сортування.
13. Відсортуйте масив методом Хора.
14. Відсортуйте методом «бульбашки» елементи масиву, що стоять на парних позиціях.
15. Відсортуйте п'ять елементів за сім порівнянь.

? *Питання для самоконтролю*

1. Які особливості роботи зі списками у Google Blockly ви знаєте?
2. Як можна поєднати два списки в один у Google Blockly?
3. Які алгоритми сортування даних вам відомі?
4. Який із алгоритмів сортування реалізує вбудована функція?



Лабораторна робота №7

Тема: Каталог Color. Робота з колірною палітрою

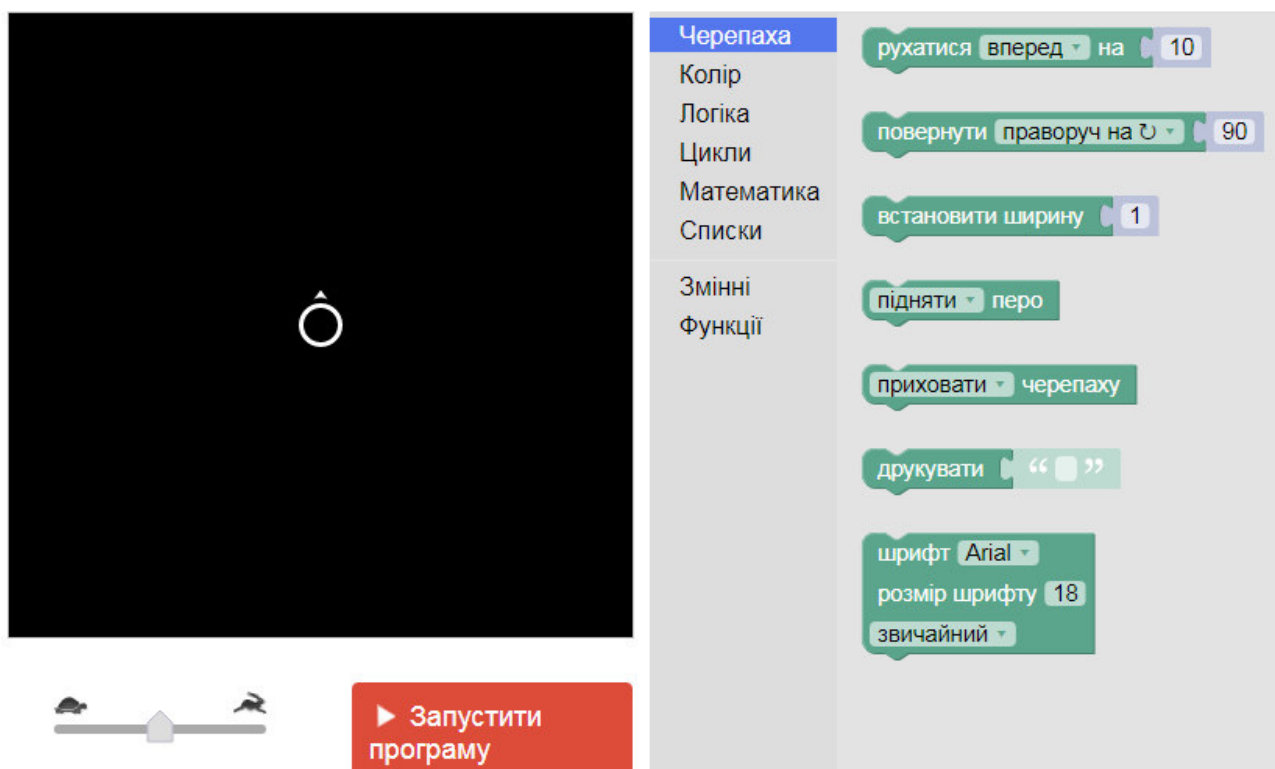
Мета: Навчитись будувати зображення у візуальному середовищі розробки Google Blockly. Ознайомитись з можливостями інструмента «Черепаша». Опанувати особливості роботи з колірною палітрою.

Теоретичні відомості

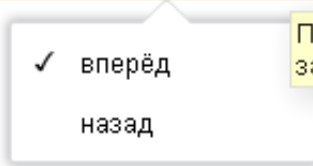
Для роботи з графічними зображеннями у візуальному середовищі розробки Google Blockly пропонується окрема група інструментів, так звана «Черепаша» [1].

Користувачеві надається поле 400x400 пікселів і розташована посередині «черепаша», у якої є декілька параметрів:

1. Напрямок руху;
2. Колір;
3. Розмір пера;
4. Видимість пера;
5. Видимість черепахи.

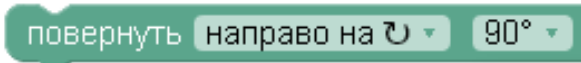


«Черепаша» вміє рухатись вперед і назад на задану довільну кількість пікселів:



Переместить черепашку вперёд или назад на заданное расстояние.

Також обернутися направо чи наліво на будь-яку кількість градусів.

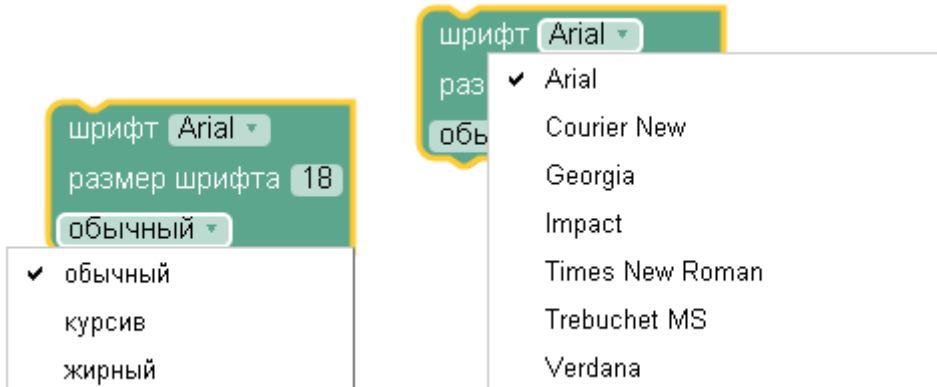


Повернуть черепашку налево или направо на заданный угол в градусах.

Також надається можливість друку текстових повідомлень з параметрами шрифту, його розміру та стилю.



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, що працюють з графічними зображеннями, можна використовувати всі раніше вивчені блоки команд, в тому числі корисним буде використання циклів, списків 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.

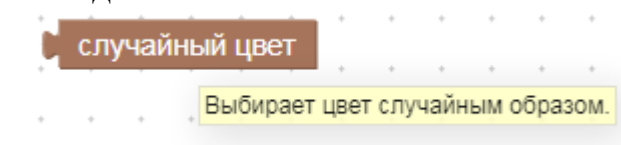
З бібліотеки:



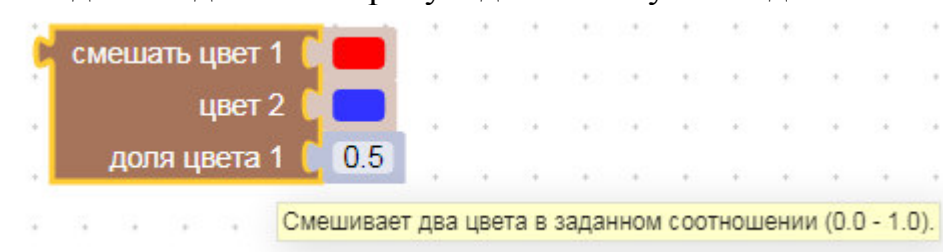
В режимі RGB:




Випадковим чином:



Поєднання двох кольорів у відсотковому співвідношенні:



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 .

Приклади програм

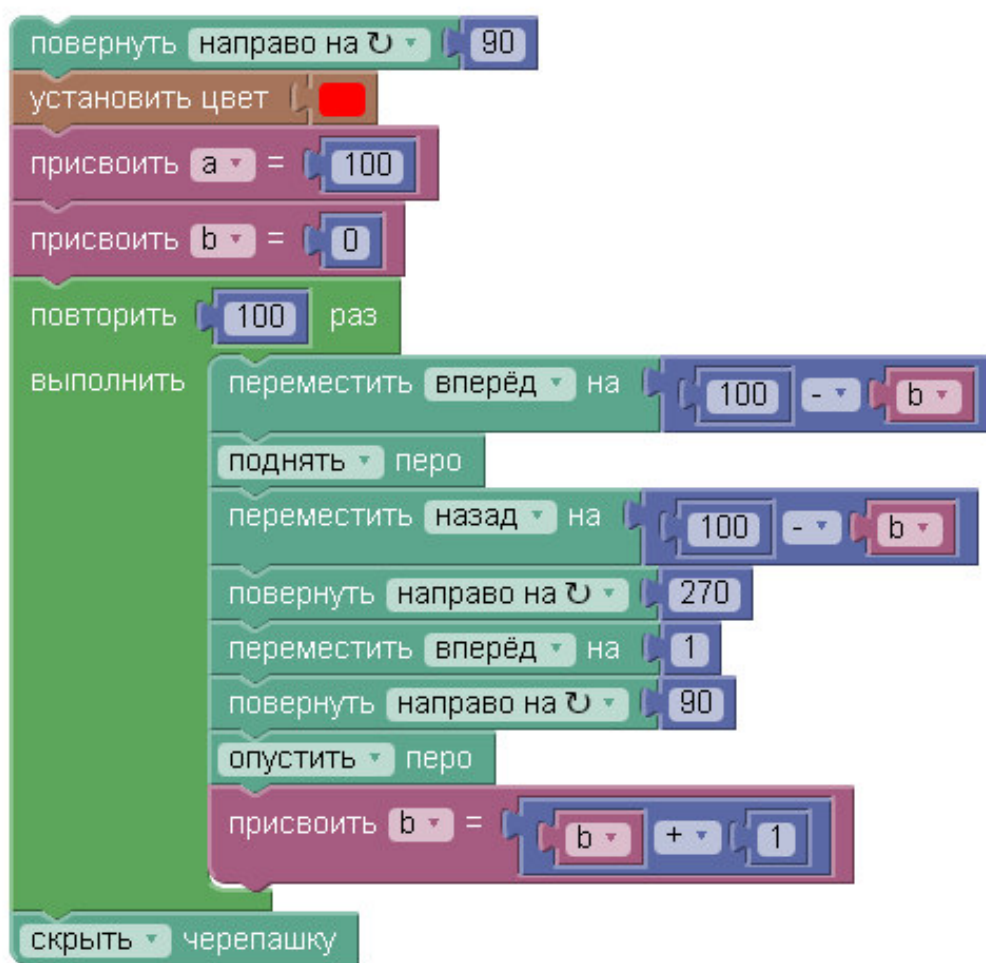
Приклад 7.1. Побудова круга (360 копій радіуса).



Щоб повністю закрасити поле, можна намалювати круг достатньо великого радіусу відповідним кольором.

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

Приклад 7.2. Побудова трикутника (побудова великої кількості відрізків)



Результат:

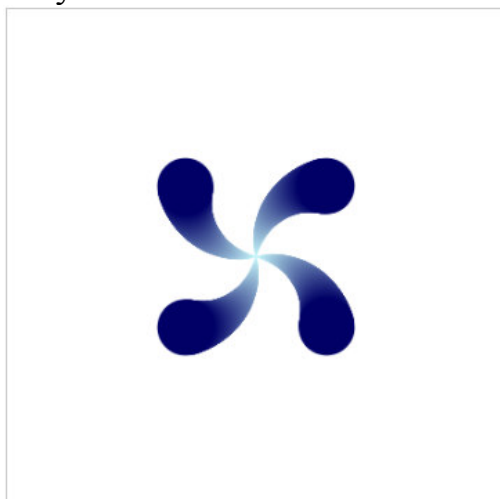


Аналогічно можна будувати прямокутник та квадрат

Приклад 7.3. Змішування кольорів

```
повторити 4 разів
зробити
  рахувати з i від 1 до 90 через 1
  зробити
    встановити ширину значення i значення ÷ 2
    встановити колір суміш колір 1 колір 2 співвідношення значення i значення ÷ 90
    рухатися вперед на 1
    повернути вправо на 1
  руку вгору
  повторити 90 разів
  зробити
    рухатися назад на 1
    повернути ліворуч на 1
  руку вниз
  повернути вправо на 90
приховати черепаху
```

Результат:



Завдання до лабораторної роботи

- Щоб розпочати роботу, перейдіть за посиланням:

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

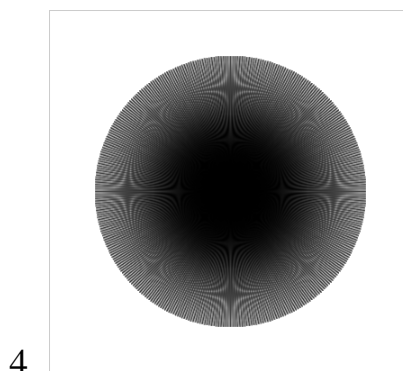
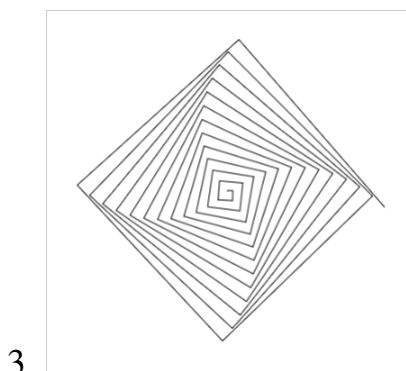
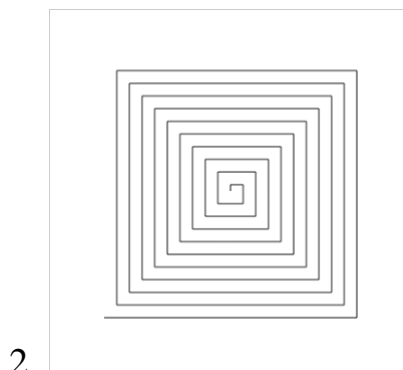
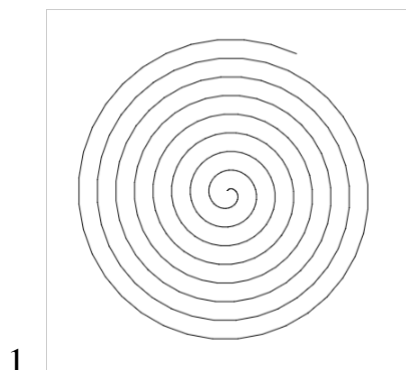
За посиланням ви знайдете легку навчальну програму з 10 завдань, і після розв'язання останньої ви зможете вільно створювати власні малюнки, а також зберігати їх до загальної галереї [1].

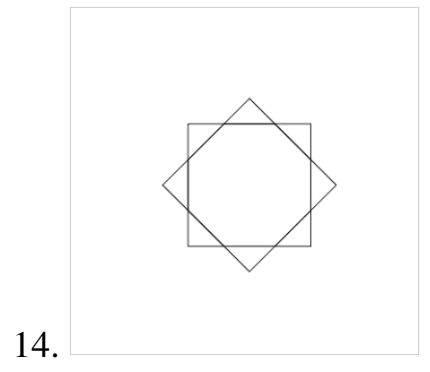
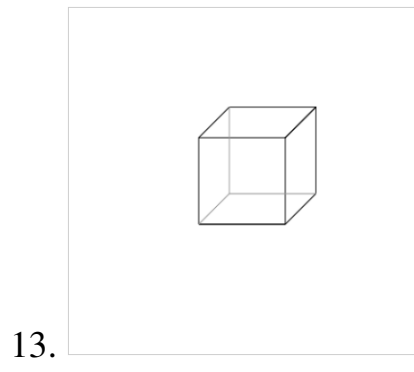
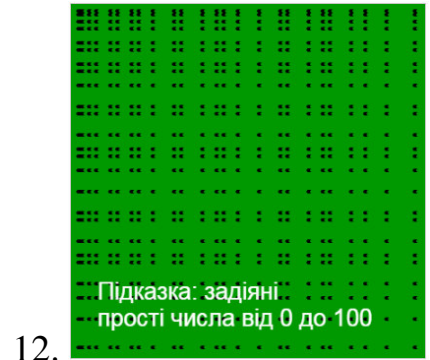
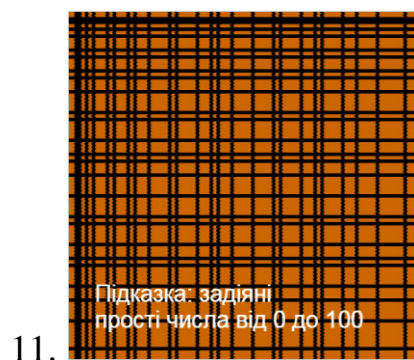
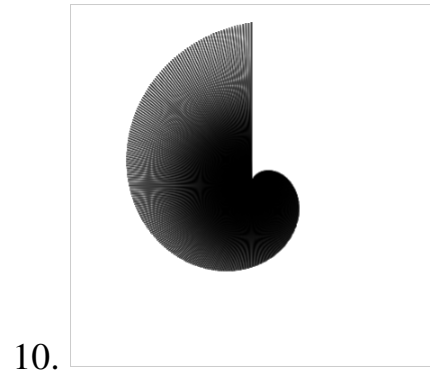
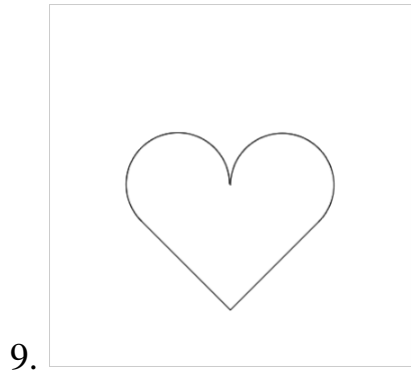
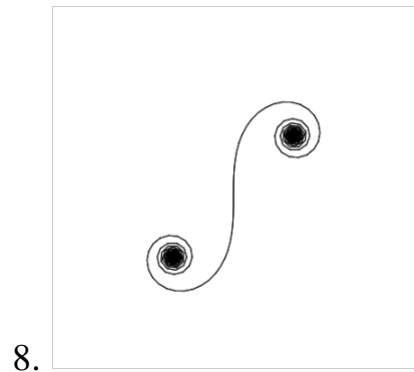
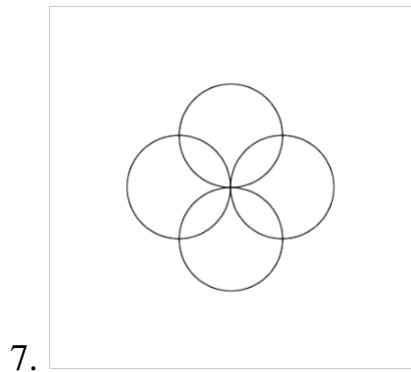
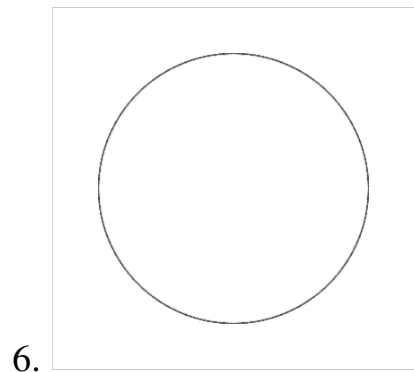
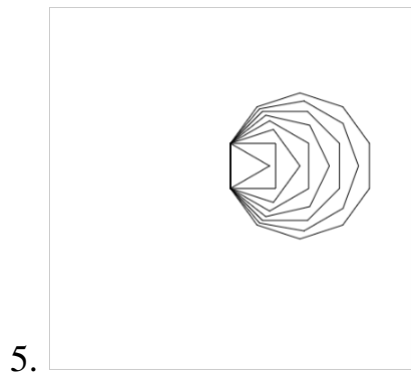
- Побудуйте пейзаж. При побудові використайте наступне:

- 1) Закрасьте 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.

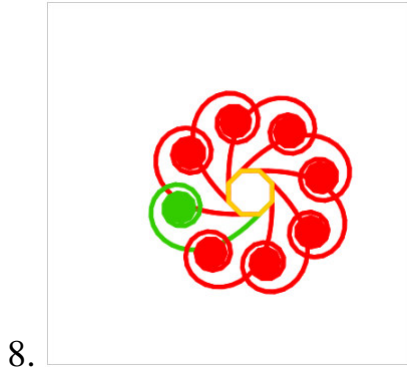
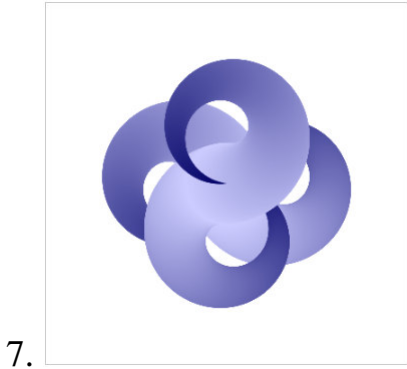
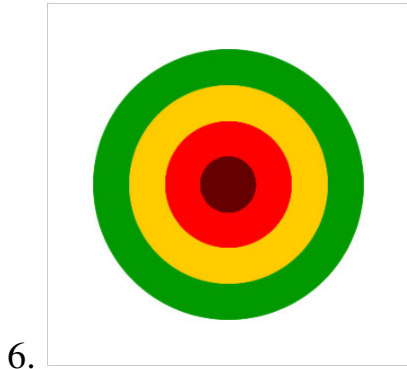
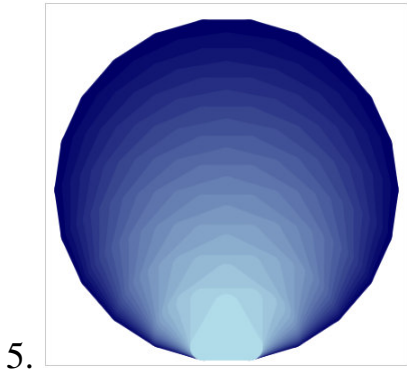
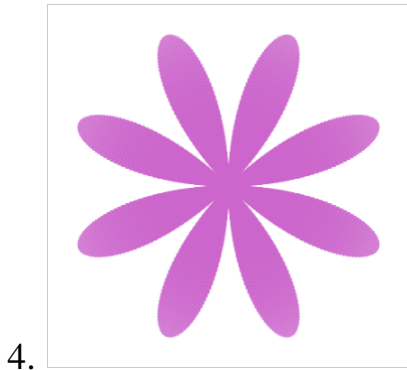
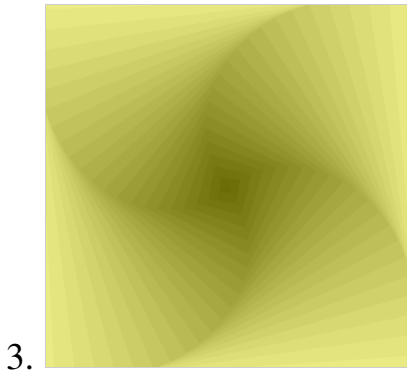
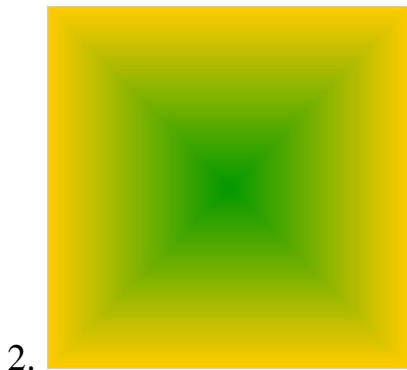
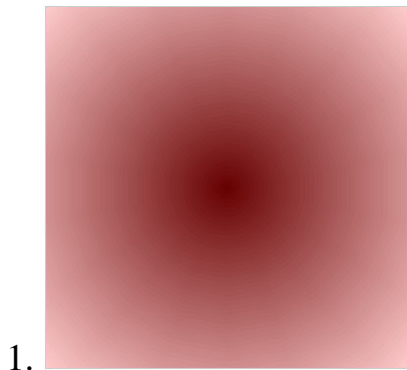
- Намалюйте фігуру з першої та другої групи. Розмірами фігур, їх кольором та іншими малозначущими параметрами можна знехтувати.

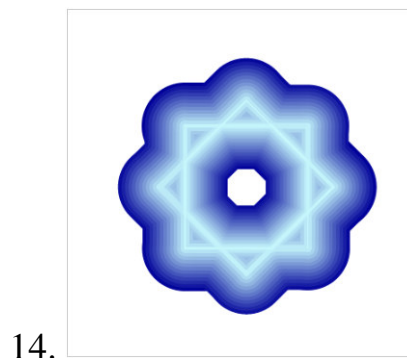
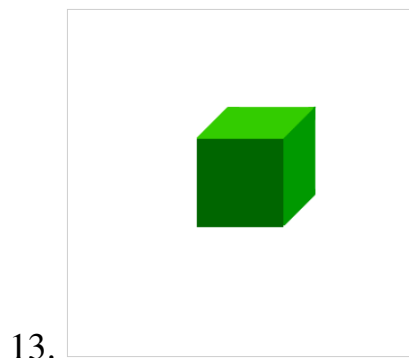
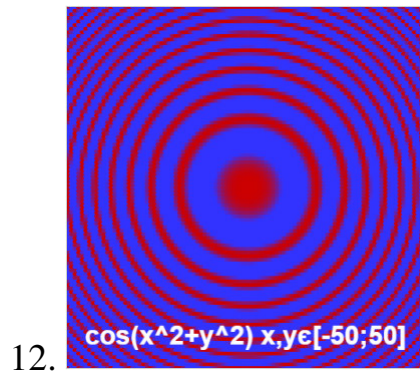
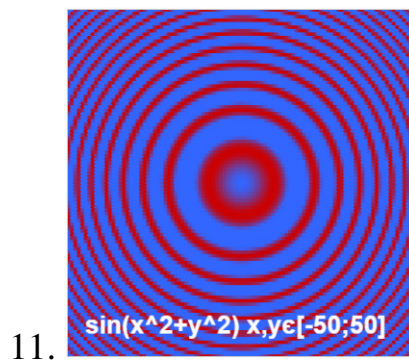
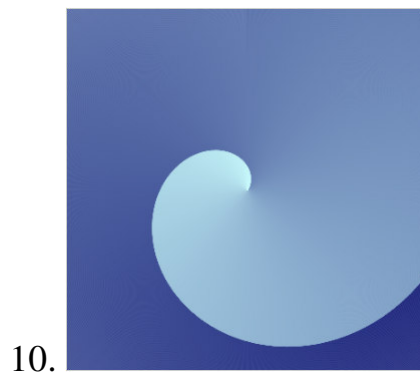
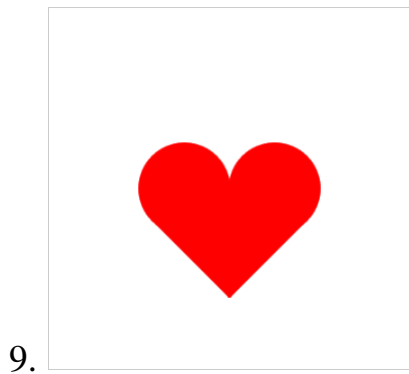
1 група завдань:





2 група завдань:

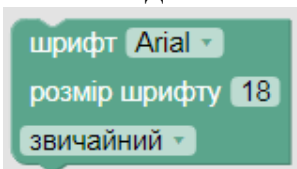




? Питання для самоконтролю

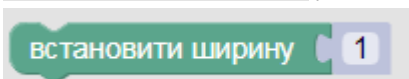
1. Які переваги та недоліки «черепашки» ви знаєте?
2. Які способи встановлення кольору пера черепахи вам відомі?

3. Для чого при малюванні використовується блок



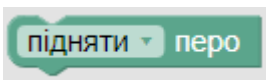
?

4. Для чого при малюванні використовується блок



?

5. Для чого при малюванні використовується блок



?

5. Наведіть приклад використання блоку



. Чим його

використання відрізняється від використання блоку



?

Лабораторна робота №8

Тема: Каталог Procedures. Створення функцій і підпрограм

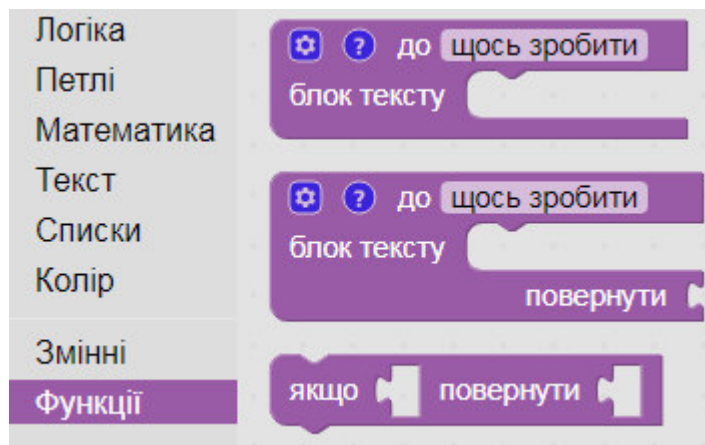
Мета: Навчитись використовувати підпрограми та функції при створенні програм.

Теоретичні відомості

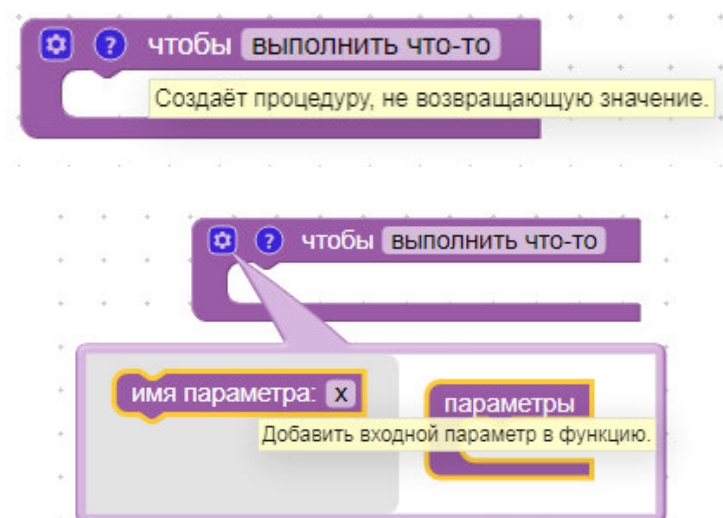
Функції добре відомі вам з інших мов програмування, але нагадаємо, що їх необхідно використовувати для таких дій, що повторюються багато разів. Ідеальна програма в головному тілі має лише декілька викликів функцій, а вони вже в свою чергу знов розбиті на під функції і так далі до найпростіших цілей.

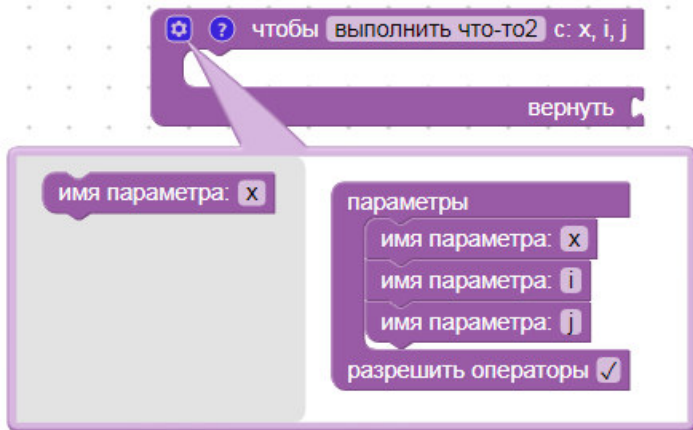
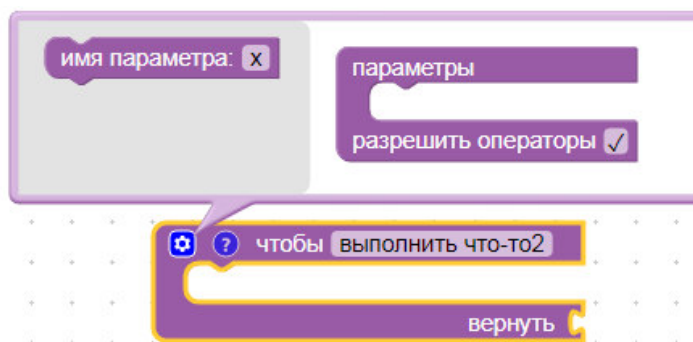
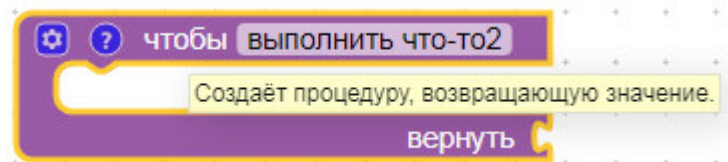
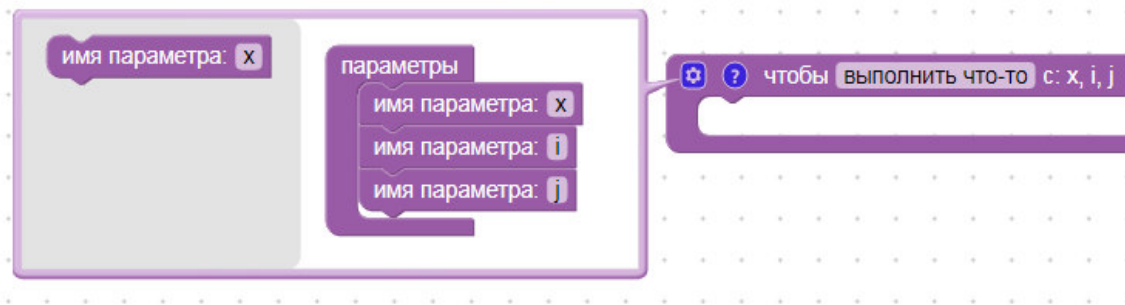
В Google Blockly окремо подаються функції (повертають певне значення) та процедури (не повертають значення). Щоб внести вхідні змінні, натисніть на зірочку, перетягніть необхідну кількість блоків і дайте імена локальним змінним. Функція вміє повертати 1 значення.

Блок безпосереднього виклику функції лежить все в тому ж каталозі «Функції».

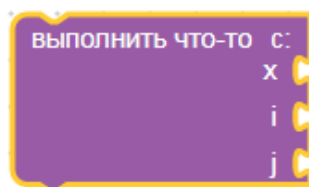
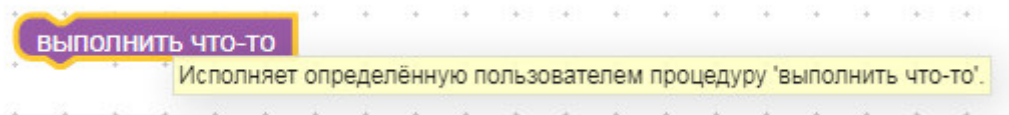


Блоки для створення процедур та функцій:





Звернення до процедур та функцій:



выполнить что-то2

Исполняет определённую пользователем процедуру 'выполнить что-то2' и возвращает вычисленное значение.

выполнить что-то2 c:
x
i
j

Приклады програм

Приклад 8.1. Обчислення факторіалу числа b

Результат:

7!=5040

Приклад 8.2. Обчислення суми факторіалів

Результат:

1!+2!+3!=9

Завдання до лабораторної роботи

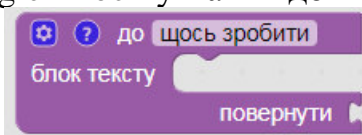
Знайти значення функції, розклавши її в ряд Маклорена, з точністю до 20 доданку, а також з наперед заданою точністю (задає користувач з клавіатури). Функція при знаходженні розкладу в ряд Маклорена повинна використовувати створені вами функції піднесення до степеня, визначення знаку доданку та обчислення факторіалу.

1. $y = e^x$
2. $y = \sin x$
3. $y = \cos x$
4. $y = \operatorname{ch} x$
5. $y = \operatorname{sh} x$
6. $y = \ln(1+x)$
7. $y = \sqrt{1+x}$
8. $y = \frac{1}{1-x}$
9. $y = \arcsin x$
10. $y = \operatorname{arctg} x$
11. $y = \operatorname{arsh} x$
12. $y = \operatorname{arth} x$
13. $y = e^{2x}$
14. $y = \sin 6x$
15. $y = \cos 2x$

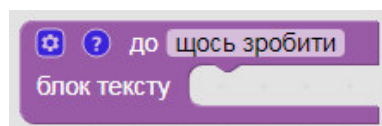
? Питання для самоконтролю

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

задані за допомогою блоків

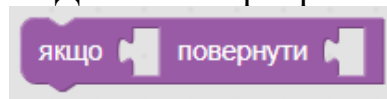


та



?

3. Для чого при роботі з функціями у Google Blockly використовується блок



?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. Ігри для майбутніх програмістів. *Сайт навчальних ігор Blockly*. URL : <https://blockly.games/?lang=uk>
2. Blockly для майбутніх програмістів. *Сайт Blockly*. URL : <http://blockly.ru/>
3. Караванова Т. П. Основи алгоритмізації та програмування: 750 задач з рекомендаціями та прикладами. Київ : ФОРУМ, 2002. 287 с.
4. Зубенко В. В., Омельчук Л. Л. Програмування. Поглиблений курс. Київ : Видавничо-поліграфічний центр «Київський університет», 2011. 623 с.

Додаткова:

1. Візуальне програмування з Google Blockly як основа знань для майбутньої професії програміста. URL : http://spec.vntu.edu.ua/conf/pdf/conf_141-147.pdf
2. Семионенков М. Н. Графическая среда программирования Blockly (Блокли). *Информатика*. №3. 2014. С. 32-40. URL : <https://drive.google.com/file/d/0B151k8zMTIuvSW1ZenI2UG85a1U/edit?pli=1>
3. Google Blockly – кандидат №1 на роль основного языка программирования. *Сайт кафедри АТЕП національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського»*. URL : <https://atep.kpi.ua/post-32/>
4. Blockly – Вікіпедія. URL : <https://uk.wikipedia.org/wiki/Blockly>
5. Кнут Д. Искусство программирования: в 3 т. Москва : Вильямс, 2007. Т. 3. Сортировка и поиск. 450 с.
6. Про візуальне програмування та Google Blockly. *Сайт прикладної математики primat.org*. URL : http://primat.org/publ/nachinajushhim/pro_vizualne_programuvannja_ta_google_blockly/66-1-0-1497
7. Современное визуальное программирование: Google Blockly. *Блог IT новин, оглядів та інтерв'ю*. URL : <http://bloggerator.org/page/sovremennoe-vizualnoe-programmirovanie-google-blockly-vpl>

ВИКОРИСТАНА ЛІТЕРАТУРА

1. Blockly для майбутніх програмістів. *Сайт Blockly*. URL : <http://blockly.ru/>
2. Зубенко В. В., Омельчук Л. Л. Програмування. Поглиблений курс. Київ : Видавничо-поліграфічний центр «Київський університет», 2011. 623 с.
3. Кнут Д. Искусство программирования: в 3 т. Москва : Вильямс, 2007. Т. 3. Сортировка и поиск. 450 с.
4. Blockly – Вікіпедія. URL : <https://uk.wikipedia.org/wiki/Blockly>
5. Современное визуальное программирование: Google Blockly. *Блог IT новин, оглядів та інтерв'ю*. URL : <http://bloggerator.org/page/sovremennoe-vizualnoe-programmirovanie-google-blockly-vpl>

ДОДАТОК А. ДЕЯКІ АЛГОРИТМИ ОБРОБКИ ДАНИХ

Наведемо основні алгоритми обробки числових даних у масивах, а саме алгоритми сортування та пошуку [2, 3].

➤ Послідовний пошук у масиві

Послідовний (лінійний) пошук у масиві передбачає послідовний перегляд усіх його елементів у порядку їхнього розташування, поки не знайдеться елемент, для якого виконується та чи інша умова. Якщо достеменно невідомо, чи є такий елемент у масиві, то необхідно стежити за тим, щоб пошук не вийшов за межі масиву.

➤ Бінарний пошук.

Цей метод застосовують виключно для впорядкованих масивів. Бінарний пошук (інші назви – дихотомічний, логарифмічний, пошук діленням навпіл) полягає в тому, що ключ порівнюється з ключем середнього елемента масиву. Якщо ці значення рівні, то шуканий елемент знайдено, в іншому випадку пошук продовжується в одній із двох половин списку (якщо ключ більше ключа середнього елемента – то в правій половині, інакше – у лівій). Відповідна половина знову ділиться навпіл і т. д. Якщо шуканий елемент існує в масиві, то на певному кроці середина відрізка з ним рано чи пізно співпаде. Якщо ж ні, то межі відрізка пошуку перетнуться і цикл пошуку слід перервати.

➤ Бульбашкове сортування.

Серед методів сортування масивів найвідомішим і найпростішим для реалізації й розуміння є обмінний алгоритм бульбашкового сортування (англ. – bubble sort). Алгоритм полягає в неодноразовому проходженні масиву, що сортується. За кожне проходження всі пари сусідніх елементів послідовно порівнюються, і якщо порядок у парі неправильний, то елементи її міняються місцями. Проходження повторюються доти, доки на черговому кроці не виявиться, що обміни більше не потрібні, отже, масив відсортовано. При проходженні алгоритму елемент, що стоїть не на своєму місці, «спливає» на потрібну позицію, як бульбашка у воді, звідки й походить назва алгоритму. Алгоритм сортування «бульбашкою» ефективний лише для невеликих масивів.

➤ Сортування вставленням.

Метод сортування вставленням простий у реалізації, ефективний на невеликих наборах даних або на частково відсортованих даних, алгоритм його стійкий. На кожному кроці алгоритму вибираємо один з елементів і вставляємо його на потрібну позицію в раніш відсортованому початку масиву доти, доки набір вхідних даних не буде вичерпано. В початковий момент відсортований масив пустий. Вибір чергового елемента зі вхідного підмасиву – довільний.

➤ Сортування злиттям.

В основу алгоритм покладено ідею принципу «розділяй і володій», суть якого полягає в: 1) розбитті задачі на менші підзадачі; 2) розв'язанні вибраних підзадач; 3) розв'язанні шляхом комбінації отриманих результатів загальної задачі. Для задачі сортування ці три етапи можуть виглядати так. Спочатку ми розбиваємо масив навпіл, потім сортуємо кожну половину окремо. Після цього здійснюємо злиття двох відсортованих масивів. Процес можна зробити

рекурсивним. Рекурсивне розбиття задачі на менші буде відбуватися доти, доки розмір підмасиву не досягне одиниці, а такий масив є впорядкованим.

➤ Швидке сортування.

Алгоритм швидкого сортування був розроблений Чарльзом Хоаром. Цей алгоритм, як і попередній, базується на принципі «розділяй і володій». Його трьома етапами є: 1) вибір деякого елемента, що називається опорним; 2) поділ масиву на два підмасиви шляхом перестановки його елементів так, щоб усі елементи, менші чи рівні опорному, виявилися ліворуч від нього, а всі елементи, більші за нього – праворуч; 3) рекурсивне впорядкування вибраних підмасивів. Базою рекурсії є підмасиви, що складаються з одного чи двох елементів. Алгоритм завжди завершується, оскільки за кожну ітерацію він ставить принаймні один елемент на його остаточне місце. Слід уважно обирати метод визначення опорної точки (за можливості з урахуванням природи масиву, що сортується). Швидке сортування, на відміну від сортування злиттям, не вимагає додаткової пам'яті.

➤ Сортування вибором.

Алгоритм полягає в наступному. Знаходять номер найменшого елемента в масиві. Проводять обмін цього значення зі значенням першої невідсортованої позиції (обмін не потрібен, якщо найменший елемент вже знаходиться на даній позиції). Слід безпосередньо вставляти мінімальний елемент в першу невідсортовану позицію, не змінюючи порядок інших елементів. Далі сортуємо «хвіст» масиву, вилучивши із розгляду вже відсортовані елементи.

➤ Пірамідалне сортування.

Цей алгоритм можна вважати удосконаленням сортування «бульбашкою», оскільки елемент «спливає» багатьма шляхами. Сортування «пірамідою» використовує бінарне сортуєче дерево. Елементи масиву розташовують у вигляді сортуєчого дерева. Далі вилучають елементи з корня по одному за раз і перебудовують дерево. Процес продовжують доти, доки в сортуєчому дереві не залишиться один елемент.

➤ Сортування Шелла.

При сортуванні Шелла спочатку порівнюють і сортують між собою значення, які розташовані один від одного на певній відстані. Потім процедуру повторюють для менших відстаней, а завершують при відстані, рівній 1.

Зауважимо, що різні підходи до сортування мають різні характеристики. Незважаючи на те, що деякі способи сортування можуть бути в середньому кращими за інші, жоден алгоритм не є ідеальним для всіх випадків. Тому широкий набір алгоритмів сортування – корисний додаток у інструментарії будь-якого програміста. Незважаючи на те, що зазвичай оптимальним є швидке сортування, воно не є кращим у всіх випадках. Наприклад, при сортуванні дуже малих списків (менше 100 елементів) додатковий обсяг роботи, пов'язаний із рекурсивними викликами швидкого сортування, може нівелювати його переваги. У таких рідких випадках один із простих методів сортування може працювати швидше. До того ж, якщо відомо, що список уже майже впорядкований або є вимоги щодо стійкості алгоритму, то якийсь інший алгоритм може виявитися прийнятнішим за швидке сортування.

Навчально-методичне видання
(українською мовою)

Тітова Ольга Олександрівна

Мухін Віталій Вікторович

ВІЗУАЛЬНІ МОВИ ПРОГРАМУВАННЯ

Методичні рекомендації до виконання лабораторних робіт
для здобувачів ступеня вищої освіти бакалавра
спеціальності «Інженерія програмного забезпечення»
освітньо-професійної програми «Програмна інженерія»

Рецензент *С.І. Гоменюк*

Відповідальний за випуск *О.О. Тітова*

Коректор *О.О. Тітова*