

Тема. Створення лінійних алгоритмів і програм

Мета:

Формування предметних компетентностей:

уміння розробляти та програмно реалізовувати лінійний алгоритм для розв'язування поставленої задачі, створювати й використовувати текстові поля й написи для організації введення/виведення даних.

Розвиток ключових компетентностей:

спілкування іноземними мовами (уміння оперувати ключовими словами мови програмування, розуміти повідомлення системи);

інформаційно-цифрова компетентність (уміння ведення діалогу “людина - технічна система”: розуміння принципів побудови інтерфейсу, роботи з діалоговими вікнами, налаштування параметрів середовища тощо);

обізнаність та самовираження у сфері культури (розуміння поняття “дружнього” інтерфейсу);

уміння вчитися впродовж життя (уміння оцінювати обсяг і якість виконаної практичної роботи, рівень досягнення навчальних цілей);

екологічна грамотність і здорове життя (свідоме дотримання правил безпеки життєдіяльності під час роботи з комп'ютерною технікою).



Теоретичний матеріал

Існує велика кількість алгоритмів, у яких всі команди повинні виконуватись послідовно одна за одною. Згадайте, наприклад, алгоритми заварювання чаю чи приготування бутерброду.

Алгоритм, у якому всі команди виконуються послідовно одна за одною, називається лінійним алгоритмом. Програма, що є реалізацією лінійного алгоритму, називається лінійною.

На цьому уроці ми спробуємо навчитися створити програму, яка буде обчислювати математичні операції та програму, яка буде малювати квадрат.

Спочатку нам потрібно згадати основні етапи створення програми.

Створення програм включає в собі декілька етапів:

1. Постановка завдання.

На цьому етапі потрібно:
збирати інформацію про завдання;
сформулювати умову завдання;
визначити зв'язки між тим, що дано й що потрібно визначити;
описати дані.

2. Моделювання

3. Побудова алгоритму

Задача №1

Дано два цілих ненульових числа. Знайти суму, різницю, добуток і частку цих чисел.

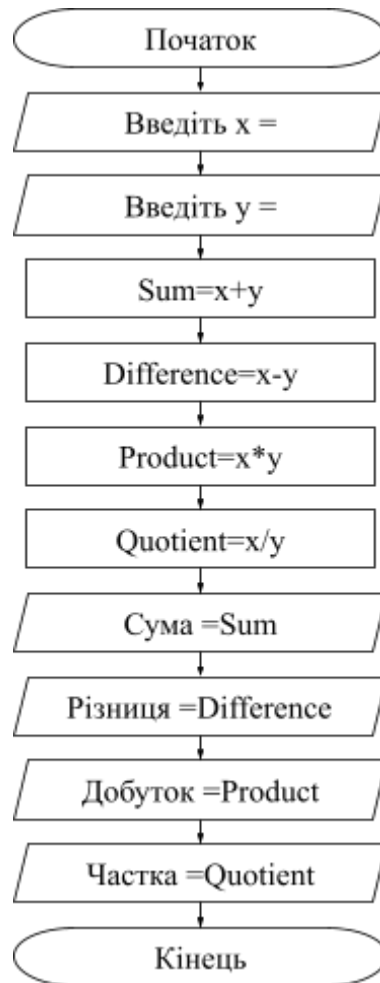
Розглядаючи дану задачу на першому етапі, нам потрібно вказати, що дано й що потрібно визначити. А також позначити тип даних, з яким будемо працювати.

У даній задачі у нас є два числа, значення яких користувач вводить самостійно. Значить ці два числа є змінними. В умові задачі ім'я цих змінних не дано, тоді ім'я змінної задається самостійно. Ми візьмемо x , y за змінні, які будуть мати тип даних цілі числа.

На другому етапі будемо блок-схему. Дана схема допоможе визначити, які команди нам потрібні та в якому порядку потрібно буде їх виконати.

Останній третій етап - створення програми, під час якого в Scratch нам знадобляться команди груп: Вигляд, Керування, Події, Датчики, Оператори, Змінні.

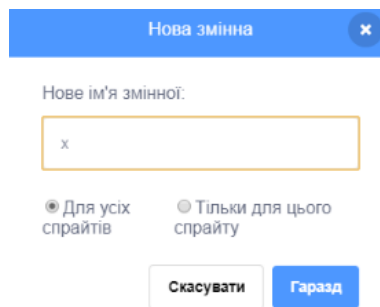
Блок-схема



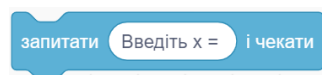
Далі спробуємо написати програму на Python. Спочатку подивимось, як команди Scratch замінюються на команди Python

	
Першою командою, яка буде відповідати блок-схемі “Початок”	
У Scratch з групи “Події” -  . Ця команда нам знадобиться для запуску програми.	Дана команда в Python зараз нам не потрібна. Програму написати можна відразу.
Далі за блок-схемою бачимо, що нам потрібно, щоб користувач надав змінним x, y значень.	

Для цього спочатку в групі “Змінні” створюємо дві змінні x , y для усіх спрайтів.

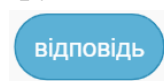


Для того щоб користувач зміг ввести значення змінної потрібно його запитати. Нам знадобляться команда із групи “Датчики”



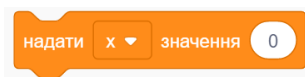
Відповідь

потрібно помістити в змінну. У групі “Датчики” беремо команду



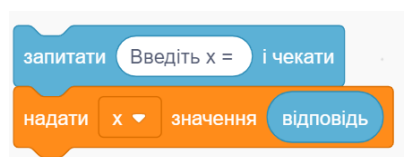
та розміщуємо її в .

Дана команда

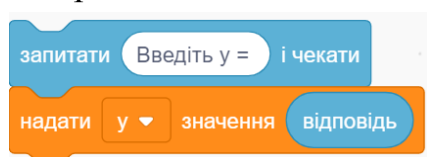


знаходиться в групі “Змінні”.

З'єднавши дані команди, користувач може вводити значення змінної для певної змінної.



Додаємо ті самі команди тільки тепер для іншої змінної.



В Python додатково створювати змінну, для подальшого використання, не потрібно. Якщо вам потрібна нова змінна для зберігання даних, то просто її пишемо та розміщуємо її потрібне значення.

$x=5$

Знак “=” означає присвоювання.

Для того щоб користувач зміг ввести самостійно значення,

використовуємо функцію `input()`.

Таким чином отримуємо команду $x=input()$. Із даною змінною

обчислення не зможемо виконати.

Оскільки на відмінно від Scratch в

Python потрібно вказати тип даних даної змінної. За стандартом всі

змінні приймають тип даних - рядковий (`str`). Тоді при обчисленні

суми даного числа отримаємо помилку при обчисленні. Функція

`int()` допоможе перевести

рядковий тип даних в цілі числа. У

дану функцію потрібно помістити

функцію `input()`. Тоді

правильною командою буде

$x=int(input())$ тобто змінній

x присвоюємо значення цілого

числа, яке введе користувач. Дану

команду бажано доповнити

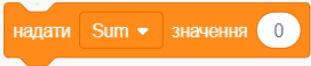
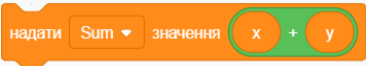
коментарями, щоб користувач

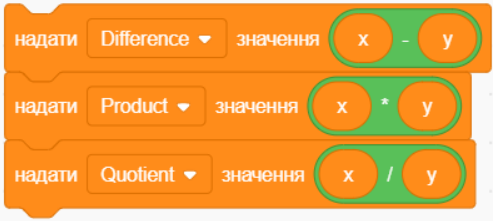
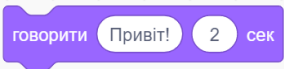
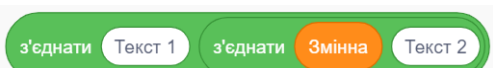
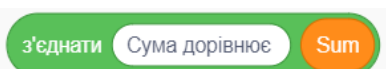
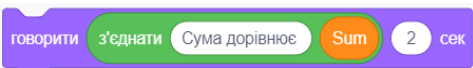
зрозумів, що йому потрібно ввести.

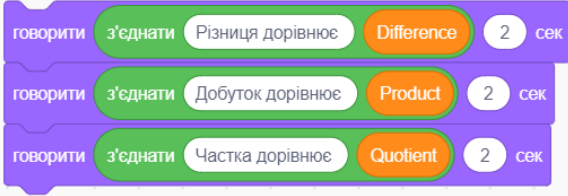
Коментарі в Python вводяться за

допомогою лапок, можна

використовувати одинарні або

	двійні <code>'Коментарій',</code> <code>"Комантарій"</code>. Дані коментарі розміщуються у функції <code>input()</code> У результаті отримаємо команду <code>x=int(input("Введіть x = "))</code> Пропишуємо ту саму команду для змінної <code>y</code> <code>y=int(input("Введіть y = "))</code>
Далі повертаємося до блок-схеми. Потрібно буде зробити обчислення суми, різниці, добутку й частки цих чисел. Узагалі дані обчислення можна виводити відразу на екран без створення змінних, але при використанні цього способу можливі складнощі з обчисленнями. Тому прийнято всі обчислення зберігати в змінних.	
Для збереження результатів створюємо в групі “Змінні” додаткові змінні: Сума, Різниця, Добуток, Частка. Далі в групі “Змінні” знадобиться команда  . Змінним надаємо значення відповідних обчислень нам знадобляться команди з групи “Оператори”, в які потрібно буде помістити два заданих числа  З'єднавши дані команди,  програма буде надавати відповідній змінній обчислення. Аналогічно створюємо команди для обчислення інших змінних.	Для збереження результатів створюємо поступово додаткові змінні. Нагадуємо, що ім'я змінних має відповідати певним вимогам, які розглядалися на попередньому уроці. Першою змінною для обчислення буде <code>Sum</code>. Цій змінній присвоюємо обчислення <code>Sum=x+y</code>. Тепер вона містить результат суми. Аналогічно прописуємо команди для інших обчислень <code>Difference=x-y</code> <code>Product=x*y</code> <code>Quotient=x/y</code>

	
Далі повертаємося до блок-схеми. Тепер потрібно вивести результати обчислення суми, різниці, добутку й частки цих чисел на екран.	
Для того щоб обчислення були виведені на екран, потрібно використати команду з групи “Вигляд”  та оскільки нам необхідно написати коментар до виведення результату, то звернімося до групи “Оператори використати”  Для більш складної конструкції в дану команду поміщається така ж сама команда  У нашому випадку потрібно використати  Поєднання даних команд допоможе вивести на екран коментар з результатом обчислення  Аналогічно створюємо команди для інших обчислень.	Для того щоб обчислення були виведені на екран, потрібно використати функцію <code>print()</code>. У дану функцію потрібно помістити змінну <code>print(Sum)</code>. Таким чином, на екран буде виводитись обчислення суми. Для зручності виводу результатів також використовують коментарі. Коментарі та змінні відокремлюється комою. Кома при виводі на екран замінюється за стандартом на символ пропуск. У результаті отримаємо команду <code>print("Сума =", Sum)</code>. Зверніть увагу на те, що функції мають фіолетовий колір, коментарі - зелений колір, змінні - чорний

	
Повертаючись до блок-схеми потрібно завершити програму.	
За кінець виконання програми використовують команду з групи “Керування” 	У Python у лінійному алгоритмі окрему функцію для завершення програми прописувати не потрібно.
З'єднавши усі блоки команд, які розбирали при створенні, вийде повноцінна програма, в якій користувач вводить два числа, а програма виводить результати обчислень з коментарями.	
	<pre> x=int(input("Введіть x = ")) y=int(input("Введіть y = ")) Sum=x+y Difference=x-y Product=x*y Quotient=x/y print("Сума =",Sum) print("Різниця =",Difference) print("Добуток =",Product) print("Частка =",Quotient) </pre>

Виходить, що більшість команд Scratch спокійно замінюються в Python.

Розглянемо ще один приклад. У Scratch дуже часто створювали програми, які малювали певні фігури або орнаменти.

Задача №2

Намалювати квадрат червоного кольору.

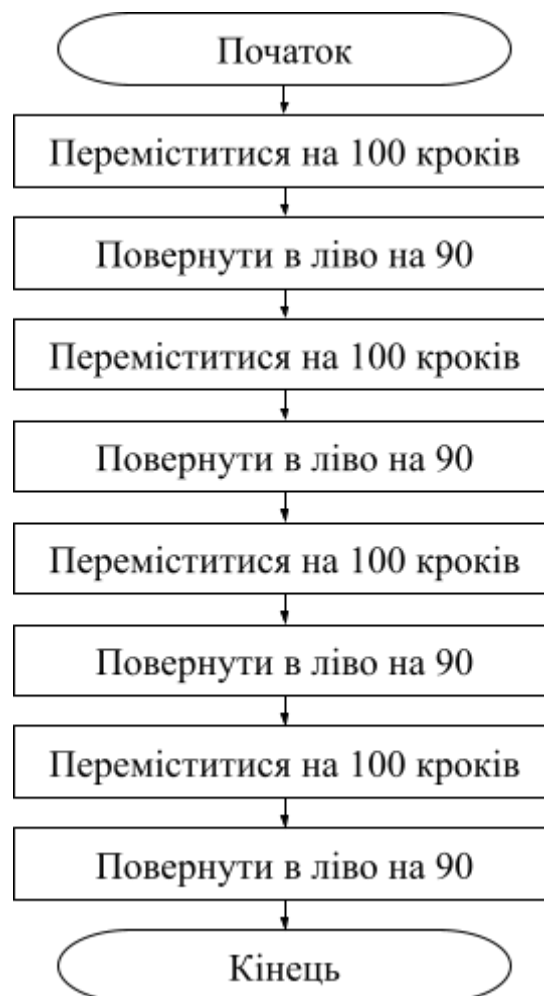
Розбираючи дану задачу на першому етапі, нам потрібно визначити, що дано й що потрібно зробити.

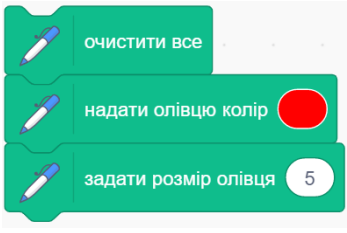
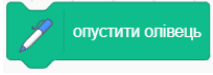
У даній задачі у нас є кут повороту та довжина сторони, значення яких прописано в програмі. Змінні в даній задачі використовувати не будемо. Потрібну довжину та кут повороту будуть вносити відразу в команди.

На другому етапі будемо блок-схему. Дана схема допоможе визначити, які команди нам потрібні та в якому порядку потрібно буде їх виконати.

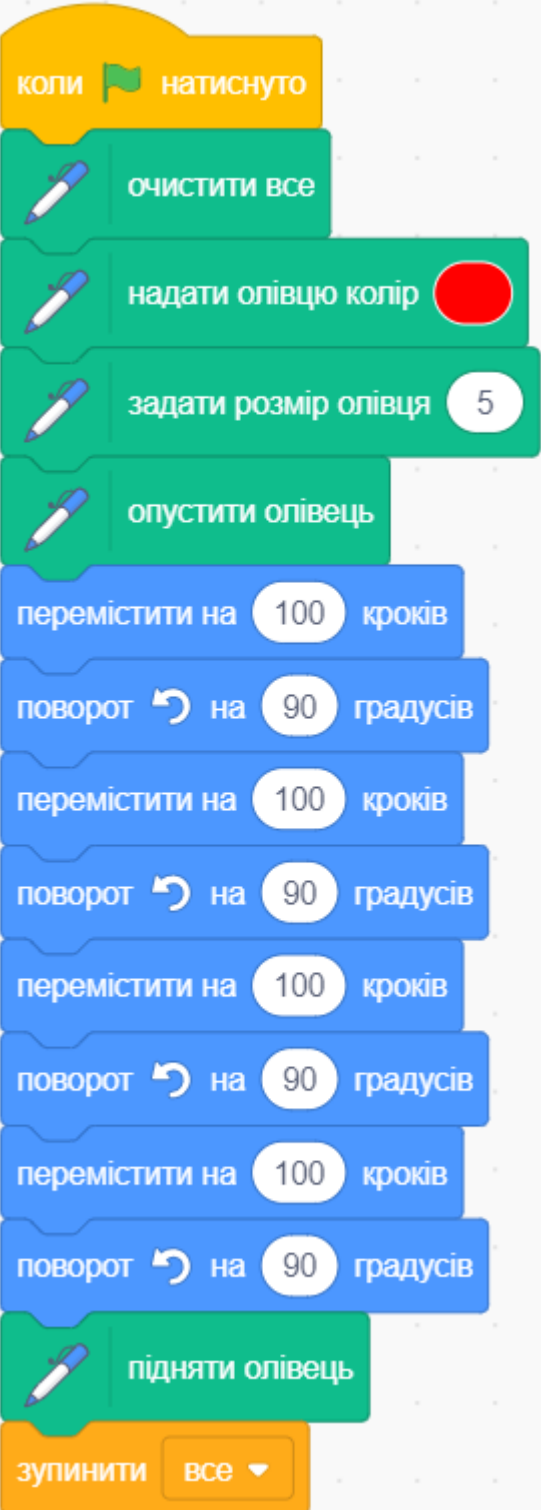
Останній третій етап - створення програми. Для цього нам в Scratch знадобляться команди груп: Олівець, Події, Рух.

Блок-схема



	
Спочатку потрібно додати бібліотеку малювання та прописати першу команду.	
У Scratch потрібно додати розширення “Олівець”  Олівець Малювати за допомогою спрайтів. Першою командою буде 	У Python першою командою в програмі потрібно прописати <pre>import turtle</pre>
Далі потрібно обрати колір олівця, товщину також потрібно очистити поле для малювання.	
	<pre>turtle.clear() turtle.pencolor("red") turtle.width(5)</pre>
Для того щоб почати малювати, потрібно опустити олівець.	
Команда  дозволить опустити олівець, який буде потім малювати.	Команда <code>turtle.down()</code> дозволить опустити олівець, який буде потім малювати.

Після цього починаємо малювати сторони квадрата за допомогою команди переміститися вперед та після поворот в ліву сторону.	
	<pre>turtle.forward(100) turtle.left(90)</pre>
Для побудови квадрата нам потрібно створити чотири сторони. Одна вже є, інші три будуються таким же чином, як і попередня сторона.	
	<pre>turtle.forward(100) turtle.left(90) turtle.forward(100) turtle.left(90) turtle.forward(100) turtle.left(90)</pre>
Після побудови всіх сторін потрібно підняти олівець.	
	<pre>turtle.up()</pre>
У кінці потрібно завершити програму.	
Для завершення виконання програми використовують команду з групи “Керування”	У Python у лінійному алгоритмі окрему функцію для завершення програми прописувати не потрібно.
З'єднавши усі блоки команд, які розбирали при створенні, вийде програма побудови червоного квадрату.	

 Scratch script components: - Yellow block: коли (when) натиснуто (clicked) - Green block: очистити все (clear all) - Green block: надати олівцю колір (set pen color to) red - Green block: задати розмір олівця (set pen size to) 5 - Green block: опустити олівець (put pen down) - Blue block: перемістити на 100 кроків (move 100 steps) - Blue block: поворот на 90 градусів (turn 90 degrees) - Blue block: перемістити на 100 кроків (move 100 steps) - Blue block: поворот на 90 градусів (turn 90 degrees) - Blue block: перемістити на 100 кроків (move 100 steps) - Blue block: поворот на 90 градусів (turn 90 degrees) - Blue block: перемістити на 100 кроків (move 100 steps) - Blue block: поворот на 90 градусів (turn 90 degrees) - Green block: підняти олівець (pick up pen) - Yellow block: зупинити все (stop all)	<pre>import turtle turtle.clear() turtle.pencolor("red") turtle.width(5) turtle.down() turtle.forward(100) turtle.left(90) turtle.forward(100) turtle.left(90) turtle.forward(100) turtle.left(90) turtle.forward(100) turtle.left(90) turtle.up()</pre>
--	--