

Тема: Створення алгоритмів і програм з повторенням. Цикл з лічильником

Мета:

Формування предметних компетентностей:

уміння давати визначення понять “цикл з лічильником”, “параметр циклу”, “тіло циклу”, “поле циклу”;

знання синтаксису й семантики циклу з лічильником;

уміння складати й виконувати алгоритми з повторенням для розв’язування задач.

Розвиток ключових компетентностей:

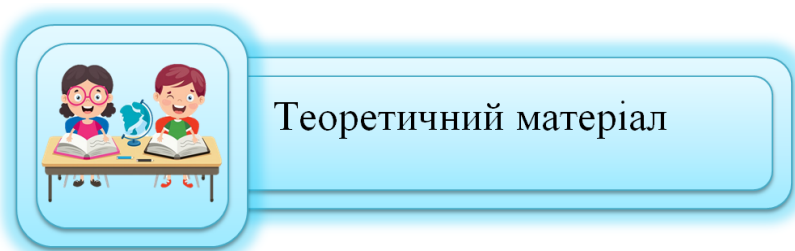
математична компетентність (уміння застосовувати математичні методи для вирішення прикладних завдань у різних сферах діяльності);

основні компетентності у природничих науках і технологіях (уміння спостерігати циклічні процеси, встановлювати причинно-наслідкові зв’язки);

інформаційно-цифрова компетентність (уміння подавати алгоритми в певному форману вигляді й виконувати їх; уміння використовувати в алгоритмах алгоритмічну структуру повторення);

уміння вчитися впродовж життя (розвиток алгоритмічного мислення: уміння визначати послідовність дій, які необхідно виконати для розв’язання певної задачі; уміння обирати оптимальний алгоритм розв’язання задачі);

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





Бувають такі випадки, коли при створенні програми необхідно виконати одну і ту ж саму послідовність дій декілька разів. Для цього використовуються цикли. На цьому уроці ми розглянемо цикл з лічильником.

Цикл - багаторазове повторення певної послідовності.

Цикл з лічильником дає змогу організувати у програмі певної послідовності дій задану кількість разів.



Лічильником є змінна. Вона поступово приймає значення кожного елемента зі списку, поки вони не закінчаться.

Синтаксис конструкція for такий:

	
	<pre>for змінна in список: дія перша дія друга ... дія n</pre> <p>де змінна - лічильник, яка буде змінюватися всередині циклу, список - список значень, які буде приймати змінна. Список записується в квадратні дужки, елементи списку розділяються комами.</p> <p>Для створення списку з цілих чисел використовується функція <code>range(x, y, n)</code> список(з числа <i>x</i>, до числа <i>y</i>, з кроком <i>n</i>). Якщо початок списку не заданий, то список починається зі значення нуль. Якщо крок не заданий, то виставляється за стандартом один.</p> <p>Потім йде блок дій, які виконуються в циклі. Як і у випадку з умовною інструкцією, блок повинен бути виділений відступом.</p>

Послідовність команд, які повторюються під час виконання оператора циклу, називають тілом циклу.

Приклад

	
---	--

	<pre>for Number in ['1', '2', '3', '4', '5']: print (Number) print ('Вийшов зайчик погуляти')</pre> <p>Змінна приймає значення зі списку, в якому п'ять елементів.</p>
--	--

У цьому прикладі Number - це змінна, яка змінюється в циклі. Тіло циклу складається з команд, які будуть виконані п'ять разів, при кожному виконанні змінна number буде приймати наступного значення. Після виконання циклу буде виконана дія для виведення на екран повідомлення “Вийшов зайчик погуляти”.

Задача №5

Дано послідовність чисел від 1 до 10 включно. Потрібно знайти середнє арифметичне цих чисел.

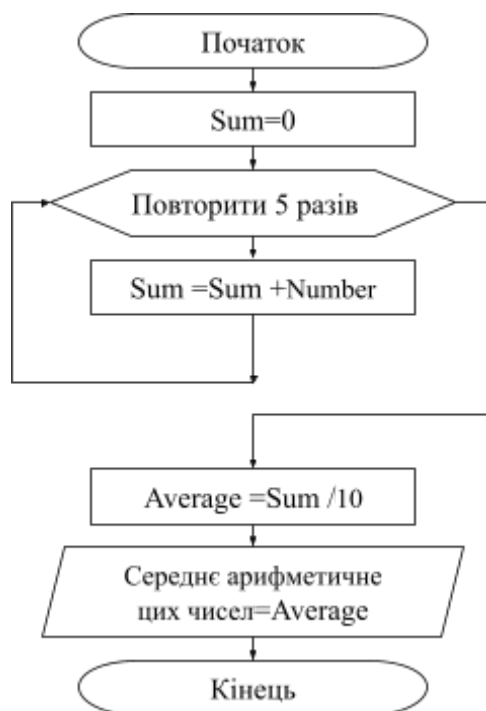
Розбираючи дану задачу на першому етапі, нам потрібно визначити, що дано і що потрібно визначити. А також визначити тип даних, з яким будемо працювати.

У даній задачі у нас є десять чисел, користувачу вводити самостійно нічого не потрібно. Середнім арифметичним декількох чисел називають суму цих чисел, поділену на кількість доданків. Тому нам потрібна буде змінна Sum (тип даних цілі числа), яка буде зберігати результати додавання. На початку програми змінна Sum повинна містити значення нуль. Ще одна змінна Average (тип даних дійсні числа), яка буде зберігати середнє арифметичне. Залишаємо змінну Number в якості лічильника.





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

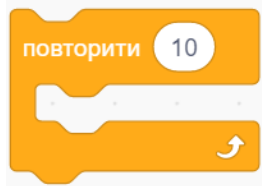
Останній третій етап вже створюємо програму в Scratch та Python.

Блок-схема



Переходимо до створення програми

	
<p>Першою командою, яка буде відповідати блок-схемі “Початок”</p>	
<p>У Scratch з групи “Події” - . Ця команда нам потрібна буде для запуску програми</p>	<p>Дана команда в Python зараз нам не потрібна. Програму написати можна відразу.</p>
<p>Далі за блок-схемою бачимо, що нам потрібно надати змінній Sum значення нулю.</p>	
	<p>Sum=0</p>
<p>Далі використовуємо конструкцію for. Задаємо кількість повторів 10.</p>	



Для використання конструкції `for` використовувати список, в якому будуть всі десять елементів буде незручно.

```
for Number in [1,2,3,4,5,6,7,8,9,10]:
```

Дану конструкцію потрібно спростити за допомогою функції `range(x, y, n)`

У результаті отримуємо

```
for Number in range(1,10+1):
```

Тобто створюємо список, в якому перший елемент списку буде число 1, а останній 10. Зверніть увагу на те, що ми додаємо 1, щоб останній елемент був 10, інакше останній елемент буде 9.

Далі повертаємося до блок-схеми. Потрібно буде обчислити суму цих чисел. Для цього ми можемо використати лічильник, оскільки числа, які нам потрібно підсумувати це числа від 1 до 10 включно.

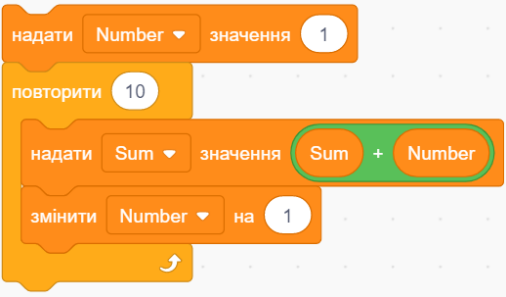
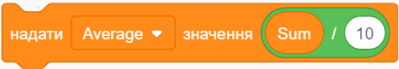

Поміщаємо команди для обчислення суми в середину циклу.

У блок-схемі ніде не вводиться значення змінної `Number`. У даному випадку з Python це лічильник, який автоматично прийме значення зі списку. У даному разі потрібно надати ще задати змінній `Number` значення 1. Тепер нам потрібно змінній `Sum` змінити значення. До попереднього значення додати наступне число, тобто змінну `Number`. І після цього лічильник `Number` збільшувати штучно.

При наповненні командами конструкції `for`

```
for Number in range(1,10+1):  
    Sum=Sum+Number
```

нам потрібно змінній `Sum` змінити значення. До попереднього значення додати наступне число, тобто змінну `Number`. І після цього лічильник `Number` приймає наступне значення зі списку.

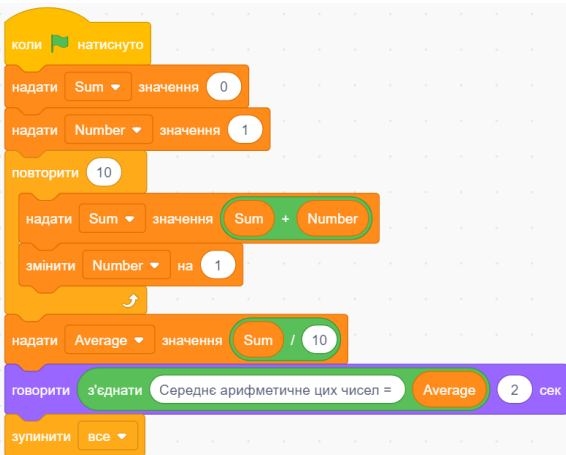
 <p>Якщо змінній Number надамо значення 0, то потрібно буде в циклі зразу збільшити значення на 1. В іншому випадку значення Number</p>	
<p>Після виконання попереднього циклу 10 разів отримуємо суму всіх 10 чисел. Дане значення міститься в змінній Sum. Далі за блок-схемою потрібно обчислити середнє арифметичне цих чисел. Нам потрібно змінній Average надати результат обчислення $Sum/10$. Дана команда виконується після виконання циклу for.</p>	
<p>У Scratch потрібно створити нову змінну Average, а після чого надавати їй значення.</p>  <p>При виконанні операції ділення змінна приймає дійсний тип даних.</p>	<p>Дана команда надасть змінній Average результат обчислення. $Average=Sum/10$</p> <p>При виконанні операції ділення змінна приймає дійсний тип даних.</p>
<p>Далі потрібно вивести результат середнього арифметичного значення на екран. Для цього будемо використовувати команди, які вивчали на попередніх уроках.</p>	
	<pre>print("Середнє арифметичне цих чисел =",Average)</pre>
<p>Повертаючись до блок-схеми, потрібно завершити програму.</p>	

Для завершення виконання програми використовують команду з групи “Керування”



У Python в лінійному алгоритмі окрему функцію для завершення програми прописувати не потрібно.

З'єднавши усі блоки команд, вийде повноцінна програма, в якій користувач вводить два числа, а програма виводить результати обчислень.



```
Sum=0
for Number in range(1,10+1):
    Sum=Sum+Number
Average=Sum/10
print("Середнє арифметичне цих чисел =",Average)
```

Тепер переходимо до іншого прикладу.

Задача №6

Програма запитує у користувача, який рівносторонній багатокутник він хоче намалювати. У відповіді користувач повинен написати ціле додатне число. У задачі потрібно врахувати, якщо користувач введе значення 1, 2. При введенні даних значень потрібно вивести повідомлення “Багатокутника з “відповідь користувача” не існує”.

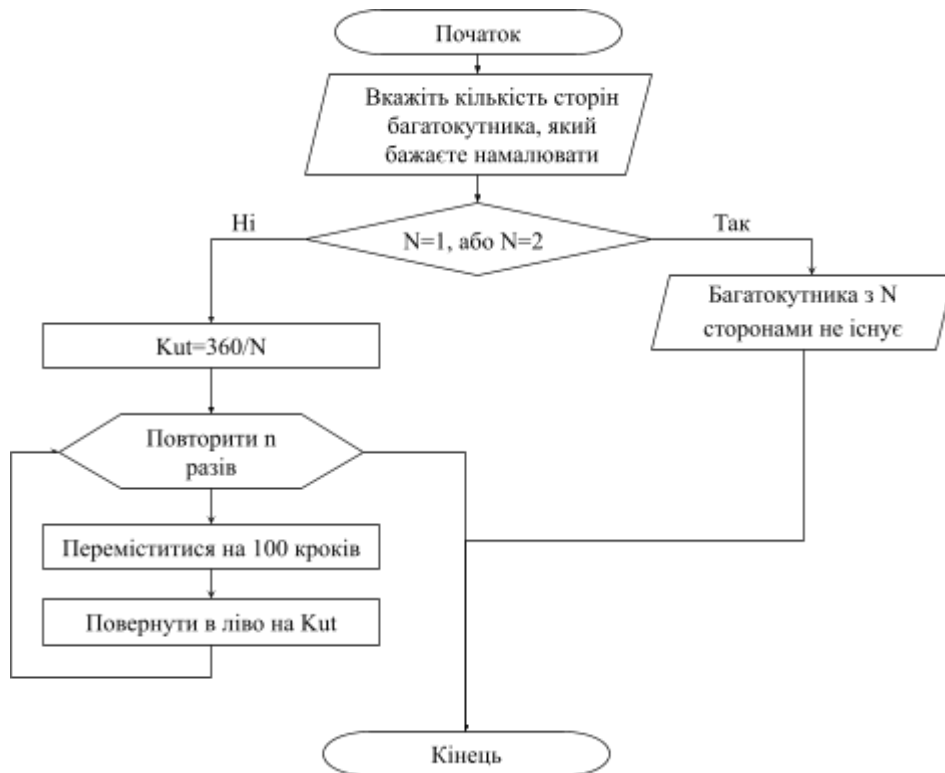
Розбираючи дану задачу на першому етапі, нам потрібно визначити, що дано і що потрібно зробити.



У даній задачі у нас є кут повороту та довжина сторони, значення якого прописано в програмі. Кут повороту залежить від кількості сторін, яку введе користувач. Формула для обчислення куту повороту $360/N$. Тому змінна Kut, якій присвоюємо дане обчислення, матиме дійсний тип даних. Також є змінна, яку вводить користувач N, вона відповідає за кількість сторін в рівностороннього багатокутника. У даної змінної цілий тип даних.

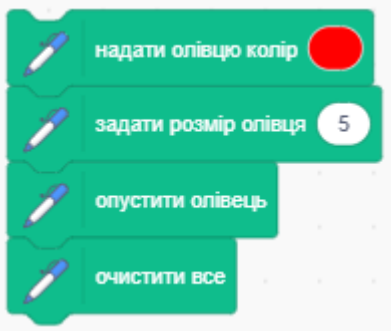
На другому етапі будемо блок-схему. Дана схема допоможе визначити, які команди нам потрібні та в якому порядку потрібно буде їх виконати.

Останній третій етап - створення програми.

Блок-схема

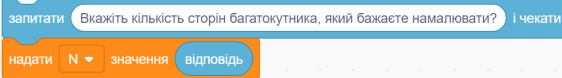


	
<p>Спочатку потрібно додати бібліотеку малювання та прописати першу команду.</p>	
<p>У Scratch потрібно додати розширення “Олівець” Та додати першу команду</p> 	<p>Додаємо першу команду в програму <code>import turtle</code></p>
<p>Далі потрібно обрати колір олівця, товщину, очистити поле та опустити олівець для малювання.</p>	



```
turtle.pencolor("red")
turtle.width(5)
turtle.down()
turtle.clear()
```

Тепер потрібно надати змінній N значення. Для цього потрібно запитати в користувача, кількість сторін багатокутника, який бажаєте намалювати.

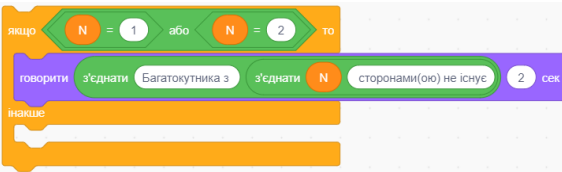


```
N=int(input("Вкажіть кількість сторін багатокутника, який бажаєте намалювати? "))
```

Додатково створюємо змінну Namber, яка буде мати значення 0. Ця змінна виступатиме в ролі лічильника.

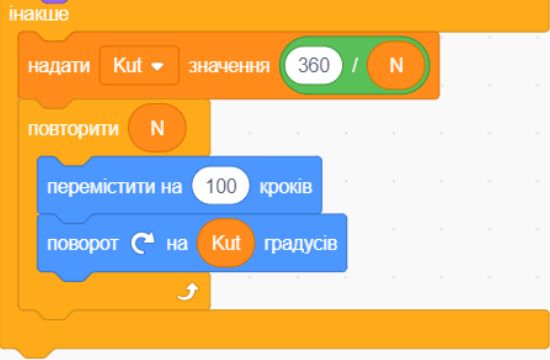
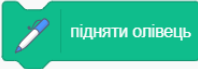



Тепер створюємо умову, при якій буде будуватися квадрат. Якщо користувач ввів кількість сторін 1, або 2, в if-блок вставляємо команду, яка виведе повідомлення "Багатокутника з N сторонами(ою) не існує". Інакше наповнюємо командами else-блок.



```
if N==1 or N==2:
    print("Багатокутника з",N,"сторонами(ою) не існує")
else:
```

Переходимо до наповнення else-блоку. Нам потрібно намалювати рівносторонній багатогранник. Для цього спочатку потрібно обчислити кут за формулою, а далі використати цикл, за кількість повторів відповідає змінна N - кількість сторін, які вводить користувач.

	<pre>else: Kut=360/N for Number in range(N): turtle.forward(100) turtle.left(Kut)</pre>
<p>Після побудови всіх сторін потрібно підняти олівець.</p>	
	<pre>turtle.up()</pre>
<p>У кінці потрібно завершити програму.</p>	
<p>Для завершення виконання програми використовують команду з групи “Керування”</p> 	<p>У Python у лінійному алгоритмі окрему функцію для завершення програми прописувати не потрібно.</p>
<p>З'єднавши усі блоки команд, вийде повноцінна програма, в якій користувач вводить два числа, а програма виводить результати обчислень.</p>	

The Scratch code blocks are as follows:

- коли натиснуто
- надати олівцю колір (червоний)
- задати розмір олівця (5)
- опустити олівець
- очистити все
- запитати "Вкажіть кількість сторін багатокутника, який бажаєте намалювати?" і чекати
- надати N значення відповідь
- якщо N = 1 або N = 2 то
- говорити з'єднати Багатокутника з з'єднати N сторонами(ою) не існує 2 сек
- інакше
- надати Kut значення $360 / N$
- повторити N
- перемістити на 100 кроків
- поворот на Kut градусів
- підняти олівець
- зупинити все

```
import turtle
turtle.pencolor("red")
turtle.width(5)
turtle.down()
turtle.clear()
N=int(input("Вкажіть кількість сторін багатокутника, який бажаєте намалювати? "))
if N==1 or N==2:
    print("Багатокутника з",N,"сторонами(ою) не існує")
else:
    Kut=360/N
    for Number in range(N):
        turtle.forward(100)
        turtle.left(Kut)
    turtle.up()
```