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

Мета:

Формування предметних компетентностей:

уміння давати визначення понять “цикл з передумовою”, “цикл з післяумовою”, “умова виконання циклу”, “тіло циклу”;

знання синтаксису й семантики циклу з передумовою;

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

Розвиток ключових компетентностей:

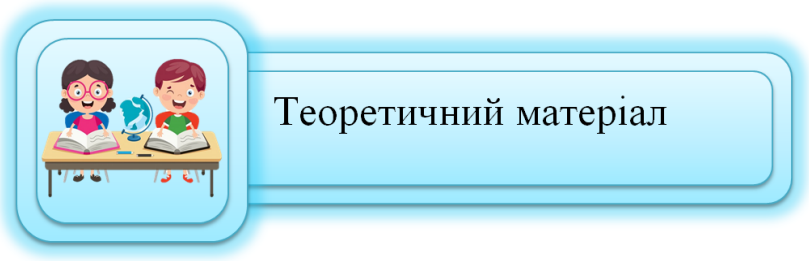
математична компетентність (уміння застосовувати математичні методи для вирішення прикладних завдань у різних сферах діяльності);

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

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

уміння вчитися впродовж життя (розвиток алгоритмічного мислення: вміння 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).

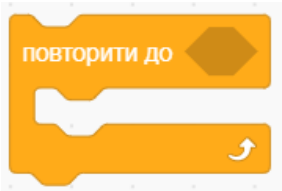


У циклі з умовою цикл виконується, поки істинна його умова. Тому цей цикл також іноді називають циклом “поки”. Він використовується, коли неможливо заздалегідь передбачити, скільки разів необхідно виконати тіло циклу. У повсякденному житті можна зустріти алгоритми, в яких його використовують, наприклад, “Поки йде дощ, йди під парасолькою”.

Цикл - багаторазове повторення певної послідовності.

Цикл з умовою дає змогу організувати у програмі певну послідовності дій за виконання необхідних умов.

Синтаксис конструкції while такий:

Scratch	Python
	`while умова:` `    дія перша` `    дія друга` `    ...` `    дія n`
Дана конструкція розшифровується так: “Повторювати дії поки не виконається умова”	Тіло циклу, що складається з однієї або декількох інструкцій, записаних з відступом однакового розміру. Дана конструкція розшифровується так: “Повторювати дії, поки виконається умова”.

У кожній інструкції while повинні бути присутніми:

Умова, що визначає, чи буде виконуватись тіло циклу. Ця умова може бути довільним арифметичним виразом, в якому повинен бути хоча б один з операторів порівняння, також можуть використовуватися логічні оператори and, or, not.

У циклі з умовою потрібно додавати команду, яка буде змінювати значення змінної, яке перевіряється в умові. Якби цієї команди не було, то значення змінної й не змінювалося б, і умова, яка перевіряється, завжди була б істинною. У результаті утворився б нескінченний цикл.

Цикл while завжди можна використовувати замість циклу for. Однак іноді цикл for зручніше, або цикл while.

Приклад:

Scratch	Python

	``` Namber=0 while Namber<=5: print(Namber) Namber=Namber+1 print("Вийшов зайчик погуляти") ```

У цьому прикладі Number - це змінна, яка змінюється в циклі. Тіло циклу складається з команд, які будуть виконані. Після завершення циклу буде реалізована дія для виведення на екран повідомлення “Вийшов зайчик погуляти”.

Задача №7

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 й що потрібно визначити. А також визначити тип даних, з яким будемо працювати.

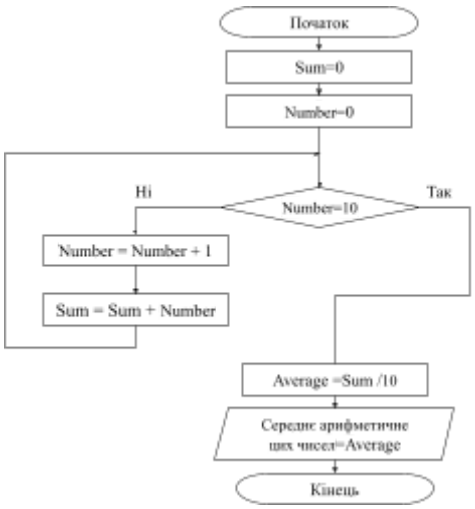
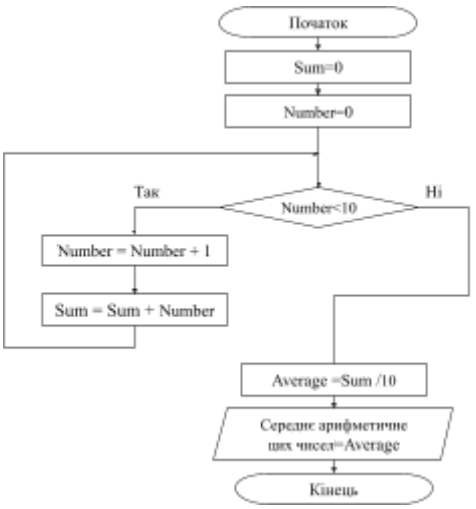
У даній задачі в нас є десять чисел, користувачу вводити самостійно нічого не потрібно. Нам потрібна буде змінна Sum (тип даних цілі числа), яка буде зберігати результати додавання. На початку програми змінна Sum повинна містити значення нуль. Ще одна змінна Average (тип даних дійсні числа), яка буде зберігати середнє арифметичне. Залишаємо змінну Number в якості лічильника.

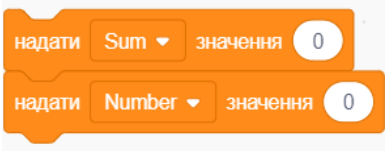
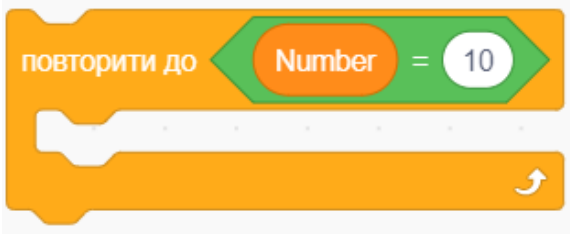
На другому етапі будуємо блок-схему. Дана схема допоможе визначити, які команди нам потрібні та в якому порядку потрібно буде їх виконати.

Останній третій етап вже створюємо програму в Scratch та Python.

Переходимо до створення програми.

	
Блок-схема	

<table>
<tr><td> 
</td><td> 
</td></tr>
<tr><td colspan="2">Першою командою, яка буде відповідати блок-схемі “Початок”.</td></tr>
<tr><td>У Scratch з групи “Події” - 

. Ця команда нам потрібна буде для запуску програми.</td><td>Дана команда в Python зараз нам не потрібна. Програму написати можна відразу.</td></tr>
<tr><td colspan="2">Далі за блок-схемою бачимо, що нам потрібно надати змінній Sum, Number значення нулю.</td></tr>
<tr><td> 
</td><td><pre>Sum=0
Namber=0</pre></td></tr>
<tr><td colspan="2">Далі використовуємо конструкцію while. Умова буде різною, оскільки даний цикл виконується по-різному.</td></tr>
<tr><td> 

Дана умова буде виконуватися до тих пір, поки вона не стане дійсною.</td><td><pre>while Namber <10:</pre>
Дана умова буде виконуватися до тих пір, поки вона дійсна.</td></tr>
</table>

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.

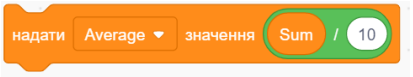
Відразу у змінній Number прописуємо команду, яка буде збільшуватися на одиницю кожного разу.
Тепер нам потрібно змінній Sum змінити значення. До попереднього значення додати наступне число, тобто змінну Number.

Відразу змінній Number прописуємо команду, яка буде збільшуватися на одиницю кожного разу.
Тепер нам потрібно змінній Sum змінити значення. До попереднього значення додати наступне число, тобто змінну Number.

```
while Namber <10:
    Namber=Namber+1
    Sum=Sum+Namber
```

Після виконання 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 while.

У Scratch потрібно створити нову змінну Average, а після цього надавати їй значення.



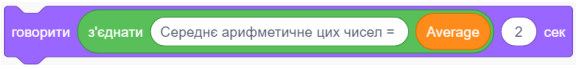
При виконанні операції ділення змінна приймає дійсний тип даних.

Дана команда надасть змінній Average результат обчислення.

```
Average=Sum/10
```

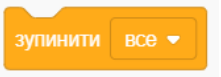
При виконанні операції ділення змінна приймає дійсний тип даних.

Далі потрібно вивести результат середнього арифметичного значення на екран. Для цього будемо використовувати команди, які вивчали на попередніх уроках.




```
print("Середнє арифметичне цих чисел =",Average)
```

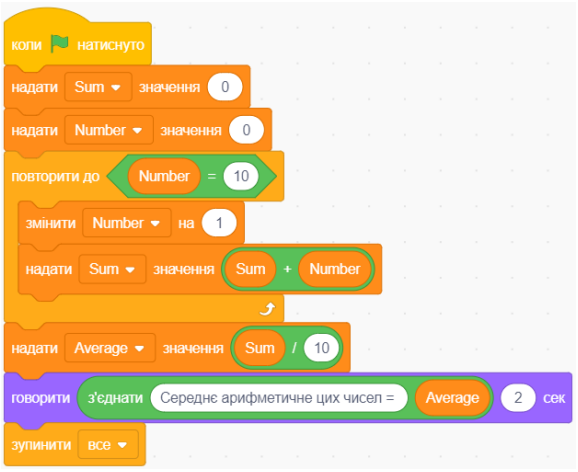
Повертаючись до блок-схеми, потрібно завершити програму.

Для завершення програми використовують команду з групи “Керування” 

.

У Python у 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
Namber=0
while Namber <10:
    Namber=Namber+1
    Sum=Sum+Namber
Average=Sum/10
print("Середнє арифметичне цих чисел =",Average)
```

Тепер перейдімо до іншого прикладу.

Задача №8

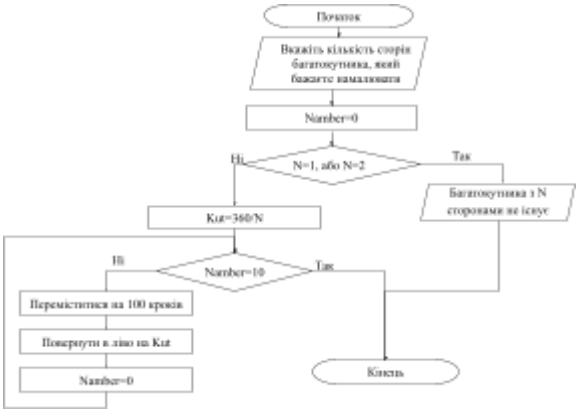
Програма запитує в користувача, який рівносторонній багатокутник він хоче намалювати. У відповіді користувач повинен ввести ціле додатне число. У задачі потрібно врахувати, якщо користувач введе значення 1, 2. При введенні даних значень потрібно вивести повідомлення “Багатокутника з “відповідь користувача”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

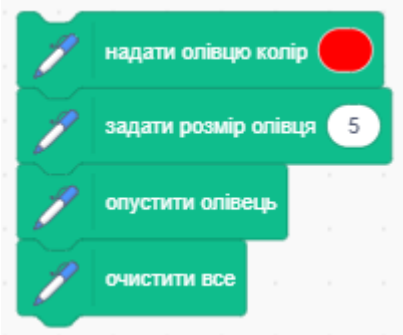
Розбираючи дану задачу на першому етапі, нам потрібно визначити, що дано і що потрібно зробити.

У даній задачі в нас є кут повороту та довжина сторони, значення, які прописано в програмі. Кут повороту залежить від кількості сторін, яку введе користувач. Формула для обчислення куту провороту 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 рівносторонньому багатокутнику. У даної змінної цілий тип даних.

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

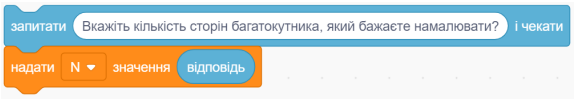
Останній третій етап - створення програми.

Scratch	python
Блок-схема	
	
Спочатку потрібно додати бібліотеку малювання та прописати першу команду.	
У Scratch потрібно додати розширення “Олівець”, першу команду 	Додаємо першу команду в програму `import turtle`
Далі потрібно обрати колір олівця, товщину, очисти поле та опустити олівець для малювання.	



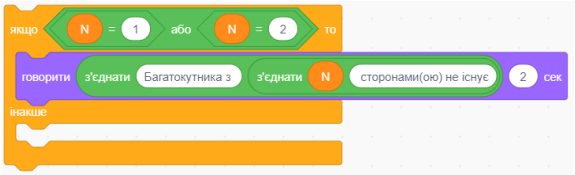

```
turtle.pencolor("red")
turtle.width(5)
turtle.down()
turtle.clear()
```

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



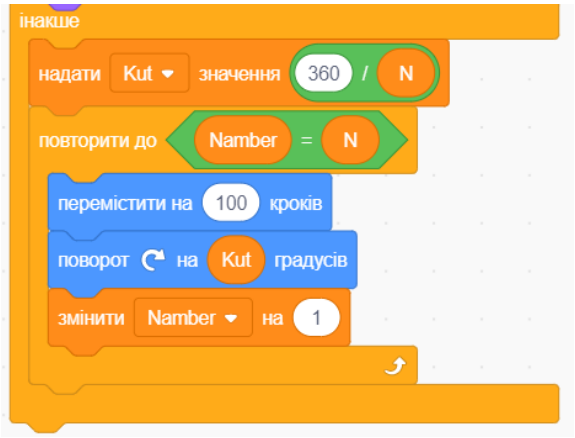

```
N=int(input("Вкажіть кількість сторін багатокутника, який бажаєте намалювати? "))
```

Тепер створюємо умову за якої 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.

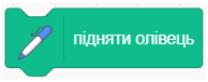
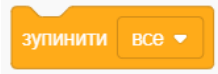


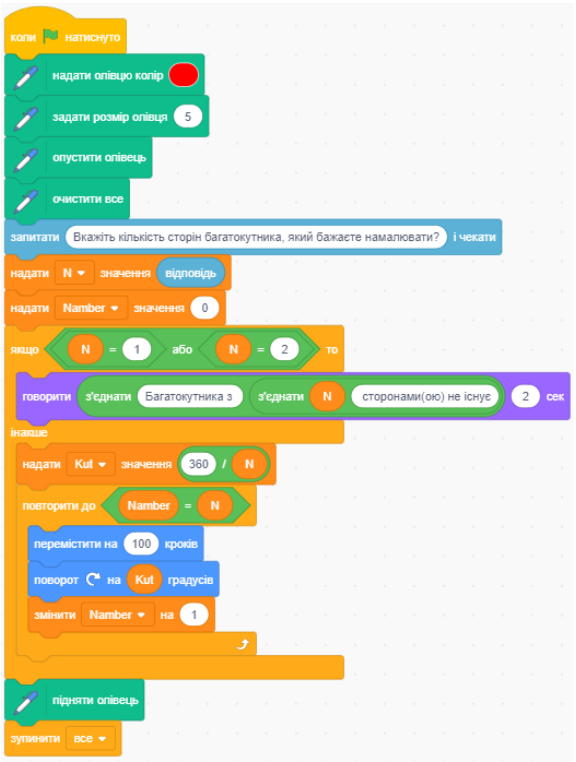

```
if N==1 or N==2:
    print("Багатокутника з",N,"сторонами(ою) не існує")
else:
```

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




```
else:
    Kut=360/N
    while Namber<N:
        Namber=Namber+1
        turtle.forward(100)
        turtle.left(Kut)
```

<table>
<tr><td colspan="2">Після побудови всіх сторін потрібно підняти олівець.</td></tr>
<tr><td> 
</td><td><pre>turtle.up()</pre></td></tr>
<tr><td colspan="2">У кінці потрібно завершити програму.</td></tr>
<tr><td>Для завершення виконання програми використовують команду з групи “Керування” 

.</td><td>У Python у лінійному алгоритмі окрему функцію для завершення програми прописувати не потрібно.</td></tr>
<tr><td colspan="2">З'єднавши усі блоки команд, вийде повноцінна програма, в якій користувач вводить два числа, а програма виводить результати обчислень.</td></tr>
<tr><td> 
</td><td><pre>import turtle
turtle.pencolor("red")
turtle.width(5)
turtle.down()
turtle.clear()
N=int(input("Вкажіть кількість сторін багатокутника, який бажаєте намалювати? "))
Namber=0
if N==1 or N==2:
    print("Багатокутника з",N,"сторонами(ою) не існує")
else:
    Kut=360/N
    while Namber<N:
        Namber=Namber+1
        turtle.forward(100)
        turtle.left(Kut)
turtle.up()</pre></td></tr>
</table>