
Лабораторная работа №3

Тема: Вложенные условия и циклы. Управление циклом (break, continue). Расширенные задачи на условия и циклы.

Цели работы

1. Освоить вложенные условия (if внутри if).
 2. Научиться использовать вложенные циклы.
 3. Освоить операторы управления циклом (break, continue).
 4. Закрепить навыки работы с циклами на более сложных задачах (суммы, таблицы, простые числа, генерация случайных чисел).
-

1. Вложенные условия

Пример. Проверка числа на положительность и чётность:

```
x = 12
```

```
if x > 0:
```

```
    if x % 2 == 0:
```

```
        print("x положительное и чётное")
```

```
    else:
```

```
        print("x положительное и нечётное")
```

```
else:
```

```
    print("x отрицательное или равно 0")
```

2. Вложенные циклы

Пример. Таблица умножения 5×5:

```
for i in range(1, 6):
```

```
    for j in range(1, 6):
```

```
        print(i * j, end="\t")
```

```
    print()
```

3. Управление циклом: break и continue

break (прерывание цикла)

```
for i in range(1, 10):
```

```
if i == 5:  
    break  
print(i)
```

Вывод:

```
1  
2  
3  
4
```

continue (пропуск итерации)

```
for i in range(1, 10):
```

```
    if i % 2 == 0:  
        continue  
    print(i)
```

Вывод:

```
1  
3  
5  
7  
9
```

4. Генерация случайных чисел

Пример. Проверка случайных чисел на чётность:

```
import random
```

```
for i in range(10):
```

```
    x = random.randint(1, 100)  
    if x % 2 == 0:  
        print(x, "чётное")  
    else:  
        print(x, "нечётное")
```

5. Расширенные задачи с циклами

◆ Лесенки

Прямая лесенка:

```
for i in range(1, 6):
```

```
    print("*" * i)
```

```
*
```

```
**
```

```
***
```

```
****
```

```
*****
```

Обратная лесенка:

```
for i in range(5, 0, -1):
```

```
    print("*" * i)
```

```
*****
```

```
****
```

```
***
```

```
**
```

```
*
```

◆ Суммы и произведения

Сумма квадратов 20 чисел:

```
s = 0
```

```
for i in range(1, 21):
```

```
    s += i ** 2
```

```
print("Сумма квадратов =", s)
```

Среднее арифметическое:

```
nums = [3, 7, 9, 12, 5]
```

```
avg = sum(nums) / len(nums)
```

```
print("Среднее =", avg)
```

◆ Поиск максимума и минимума

```
nums = [5, 7, 2, 9, 3]
```

```
print("Максимум =", max(nums))
```

```
print("Минимум =", min(nums))
```

◆ Простые числа

Простые числа от 2 до 50:

```
for num in range(2, 51):
```

```
    prime = True
```

```
    for i in range(2, num):
```

```
        if num % i == 0:
```

```
            prime = False
```

```
            break
```

```
    if prime:
```

```
        print(num, end=" ")
```

Результат:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

◆ Полная таблица умножения 10×10 с заголовками

```
print(" ", end="")
```

```
for j in range(1, 11):
```

```
    print(f"{j:4}", end="")
```

```
print()
```

```
print("-" * 50)
```

```
for i in range(1, 11):
```

```
    print(f"{i:2} |", end="") # заголовок строки
```

```
    for j in range(1, 11):
```

```
        print(f"{i*j:4}", end="")
```

```
    print()
```

✦ Результат:

```
1  2  3  4  5  6  7  8  9 10
```

```
1 | 1  2  3  4  5  6  7  8  9 10
```

2| 2 4 6 8 10 12 14 16 18 20
3| 3 6 9 12 15 18 21 24 27 30
4| 4 8 12 16 20 24 28 32 36 40
5| 5 10 15 20 25 30 35 40 45 50
6| 6 12 18 24 30 36 42 48 54 60
7| 7 14 21 28 35 42 49 56 63 70
8| 8 16 24 32 40 48 56 64 72 80
9| 9 18 27 36 45 54 63 72 81 90
10| 10 20 30 40 50 60 70 80 90 100

6. Задания для самостоятельной работы

1. Проверить, является ли введённое число положительным, отрицательным или нулём.
2. Составить программу, которая выводит таблицу умножения 12×12 .
3. Вывести все нечётные числа от 1 до 100, используя оператор `continue`.
4. Вычислить сумму кубов первых 15 чисел.
5. Сгенерировать 20 случайных чисел от 1 до 100 и подсчитать, сколько из них чётные.
6. Найти максимальное и минимальное число в списке из 15 случайных чисел.
7. Вывести обратную лесенку из символа `#` высотой 7 строк.
8. Вывести все простые числа от 1 до 100.
9. Найти произведение всех нечётных чисел от 1 до 15.
10. Вывести "числовую пирамиду":

1
12
123
1234
12345
