

Программирование на языке Python

Лекц_5 Циклические алгоритмы

Д/з 4 вида заданий

Что такое цикл?

Цикл – это многократное выполнение одинаковых действий.

Два вида циклов:

- цикл с **известным** числом шагов (сделать 10 раз)
- цикл с **неизвестным** числом шагов (делать, пока не надоест)

Задача. Вывести на экран 10 раз слово «Привет».



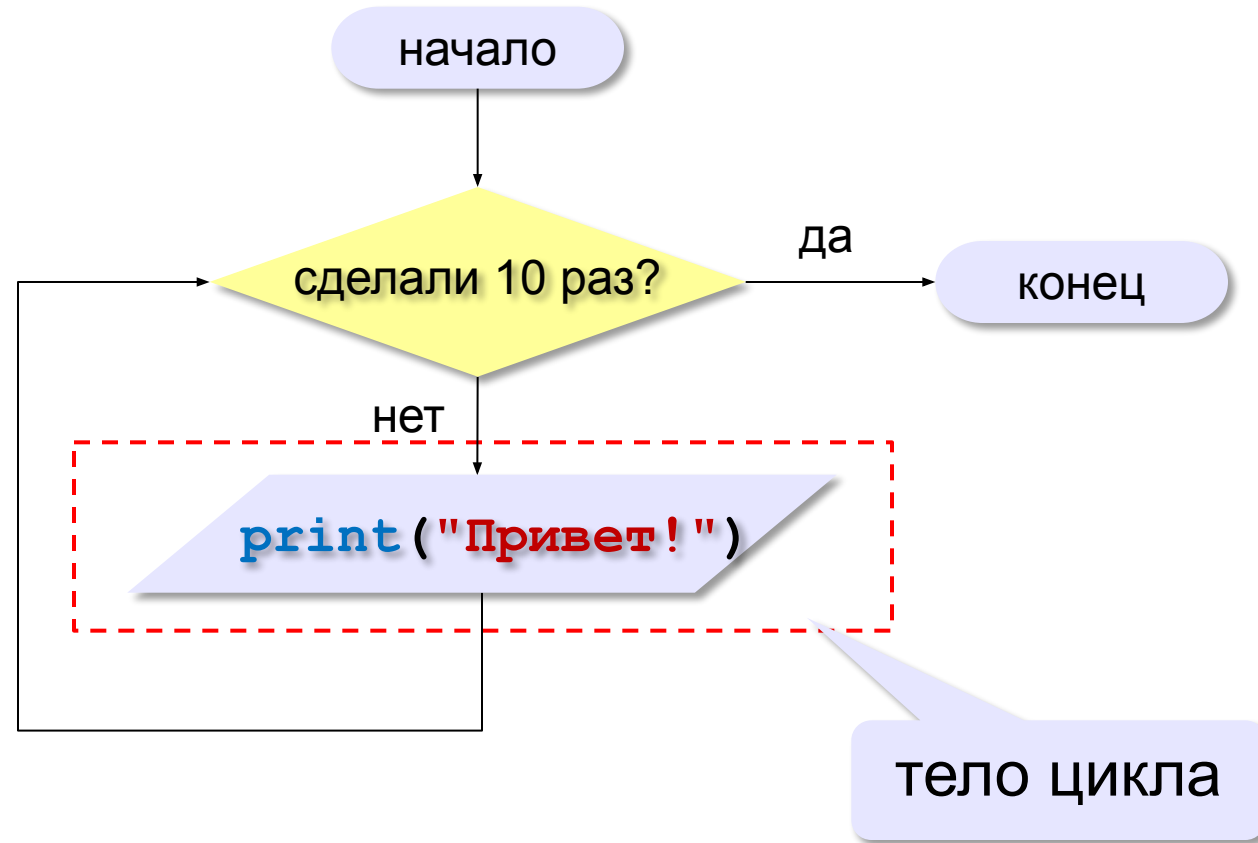
Можно ли решить известными методами?

Повторения в программе

```
print ("Привет")  
print ("Привет")  
...  
print ("Привет")
```



Блок-схема цикла



Как организовать цикл?

```
счётчик = 0
пока счётчик < 10:
    print("Привет")
    увеличить счётчик на 1
```

```
счётчик = 10
пока счётчик > 0:
    print("Привет")
    уменьшить счётчик на 1
```

результат операции
автоматически
сравнивается с нулём!



Какой способ удобнее для процессора?

Цикл с условием

Задача. Определить **количество цифр** в десятичной записи целого положительного числа, записанного в переменную n .

```
счётчик = 0
пока n > 0:
    отсечь последнюю цифру n
    увеличить счётчик на 1
```

? Как отсечь последнюю цифру?

```
n = n // 10
```

? Как увеличить счётчик на 1?

```
счётчик = счётчик + 1
```

n	счётчик
1234	0

```
счётчик += 1
```

Цикл с условием

начальное значение
счётчика

условие
продолжения

заголовок
цикла

```
count = 0;  
while n > 0 :  
    n = n // 10  
    count += 1
```

тело цикла



Цикл с предусловием – проверка на входе в цикл!

Цикл с условием

При известном количестве шагов:

```
k = 0
while k < 10:
    print ( "привет" )
    k += 1
```

Защивление:

```
k = 0
while k < 10:
    print ( "привет" )
```


Сколько раз выполняется цикл?

```
a = 4; b = 6  
while a < b: a += 1
```

2 раза
a = 6

```
a = 4; b = 6  
while a < b: a += b
```

1 раз
a = 10

```
a = 4; b = 6  
while a > b: a += 1
```

0 раз
a = 4

```
a = 4; b = 6  
while a < b: b = a - b
```

1 раз
b = -2

```
a = 4; b = 6  
while a < b: a -= 1
```

зацикливание

Цикл с постусловием

Задача. Обеспечить ввод **положительного** числа в переменную `n`.

бесконечный
цикл

```
while True:  
    print ( "Введите положительное число:" )  
    n = int ( input ( ) )  
    if n > 0: break
```

тело цикла

условие
выхода

прервать
цикл

- при входе в цикл условие **не проверяется**
- цикл всегда выполняется **хотя бы один раз**

Д/з Задачи -1

«А»: Напишите программу, которая получает два целых числа A и B ($0 < A < B$) и выводит квадраты всех натуральных чисел в интервале от A до B .

Пример:

Введите два целых числа:

10 12

10*10=100

11*11=121

12*12=144

«В»: Напишите программу, которая получает два целых числа и находит их произведение, не используя операцию умножения. Учтите, что числа могут быть отрицательными.

Пример:

Введите два числа:

10 -15

10 * (-15) = -150

Д/з Задачи -1

«С»: Ввести натуральное число N и вычислить сумму всех чисел Фибоначчи, меньших N . Предусмотрите защиту от ввода отрицательного числа N .

Пример:

Введите число N :

10000

Сумма 17709

Д/з Задачи -2

«А»: Ввести натуральное число и найти сумму его цифр.

Пример:

Введите натуральное число :

12345

Сумма цифр 15.

«В»: Ввести натуральное число и определить, верно ли, что в его записи есть две одинаковые цифры, стоящие рядом.

Пример:

Введите натуральное число :

12342

Нет.

Пример:

Введите натуральное число :

12245

Да.

Д/з Задачи -2

«С»: Ввести натуральное число и определить, верно ли, что в его записи есть две одинаковые цифры (не обязательно стоящие рядом).

Пример:

Введите натуральное число :

12342

Да .

Пример:

Введите натуральное число :

12345

Нет .

Цикл с переменной

Задача. Вывести 10 раз слово «Привет!».

? Можно ли сделать с циклом «пока»?

```
i = 0;
while i < 10 :
    print ("Привет!")
    i += 1
```

Цикл с переменной:

```
for i in range(10) :
    print ("Привет!")
```

в диапазоне
[0, 10)

! Не включая 10!

`range(10)` → 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Цикл с переменной

Задача. Вывести все степени двойки от 2^1 до 2^{10} .

 Как сделать с циклом «пока»?

```
k = 0;
while k < 10 :
    print ( 2**k )
    k += 1
```

Цикл с переменной:

```
for k in range (1, 11) :
    print ( 2**k )
```

в диапазоне
[1, 11)

 Не включая 11!

`range (1, 11)` → 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Цикл с переменной: другой шаг

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

шаг

```
for k in range(10, 0, -1):  
    print(k**2)
```



Что получится?

1, 3, 5, 7, 9

```
for k in range(1, 11, 2):  
    print(k**2)
```

100

81

64

49

36

25

16

9

4

1

1

9

25

49

81

Сколько раз выполняется цикл?

```
a = 1  
for i in range(3): a += 1
```

a = 4

```
a = 1  
for i in range(3, 1): a += 1
```

a = 1

```
a = 1  
for i in range(1, 3, -1): a += 1
```

a = 1

```
a = 1  
for i in range(3, 1, -1): a += 1
```

a = 3

Д/з Задачи -3

«А»: Найдите все пятизначные числа, которые при делении на 133 дают в остатке 125, а при делении на 134 дают в остатке 111.

«В»: Натуральное число называется **числом Армстронга**, если сумма цифр числа, возведенных в N-ную степень (где N – количество цифр в числе) равна самому числу. Например, $153 = 1^3 + 5^3 + 3^3$.
Найдите все трёхзначные Армстронга.

Д/з Задачи -3

«С»: Натуральное число называется автоморфным, если оно равно последним цифрам своего квадрата. Например, $25^2 = 625$. Напишите программу, которая получает натуральное число N и выводит на экран все автоморфные числа, не превосходящие N.

Пример:

Введите N:

1000

1*1=1

5*5=25

6*6=36

25*25=625

76*76=5776

Вложенные циклы

Задача. Вывести все простые числа в диапазоне от 2 до 1000.

```
сделать для n от 2 до 1000
    если число n простое то
        вывод n
```

нет делителей [2.. n-1]:
проверка в цикле!

? Что значит «простое число»?

```
for n in range(2, 10001):
    if число n простое:
        print( n )
```

Вложенные циклы

```
for n in range(2, 10001):  
    count = 0;  
    for k in range(2, n):  
        if n % k == 0:  
            count += 1  
    if count == 0:  
        print( n )
```

ВЛОЖЕННЫЙ ЦИКЛ

Вложенные циклы

```
for i in range(1, 45):  
    for k in range(1, i+1):  
        print( i, k )
```

? Как меняются переменные?

! Переменная внутреннего цикла изменяется быстрее!

```
1 1  
2 1  
2 2  
3 1  
3 2  
3 3  
4 1  
4 2  
4 3  
4 4
```

Поиск простых чисел – как улучшить?

$$n = k \cdot m, \quad k \leq m \Rightarrow k^2 \leq n \Rightarrow k \leq \sqrt{n}$$

```
while k <= math.sqrt(n):
    ...
```

? Что плохо?

```
count = 0
k = 2
while k*k <= n:
    if n % k == 0:
        count += 1
    k += 1
```

? Как ещё улучшить?

ВЫЙТИ ИЗ ЦИКЛА

```
while k*k <= n:
    if n % k == 0: break
    k += 1
if k*k > n:
    print ( n )
```

ЕСЛИ ВЫШЛИ
ПО УСЛОВИЮ

Д/з Задачи -4

«А»: Напишите программу, которая получает натуральные числа A и B ($A < B$) и выводит все простые числа в интервале от A до B .

Пример:

Введите границы диапазона:

10 20

11 13 17 19

«В»: В магазине продается мастика в ящиках по 15 кг, 17 кг, 21 кг. Как купить ровно 185 кг мастики, не вскрывая ящики? Сколькими способами можно это сделать?

Д/з Задачи -4

«С»: Ввести натуральное число N и вывести все натуральные числа, не превосходящие N и делящиеся на каждую из своих цифр.

Пример:

Введите N:

15

1 2 3 4 5 6 7 8 9 11 12 15