



Фоксфорд

Кружки

Кружок по программированию на Python

Занятие №6

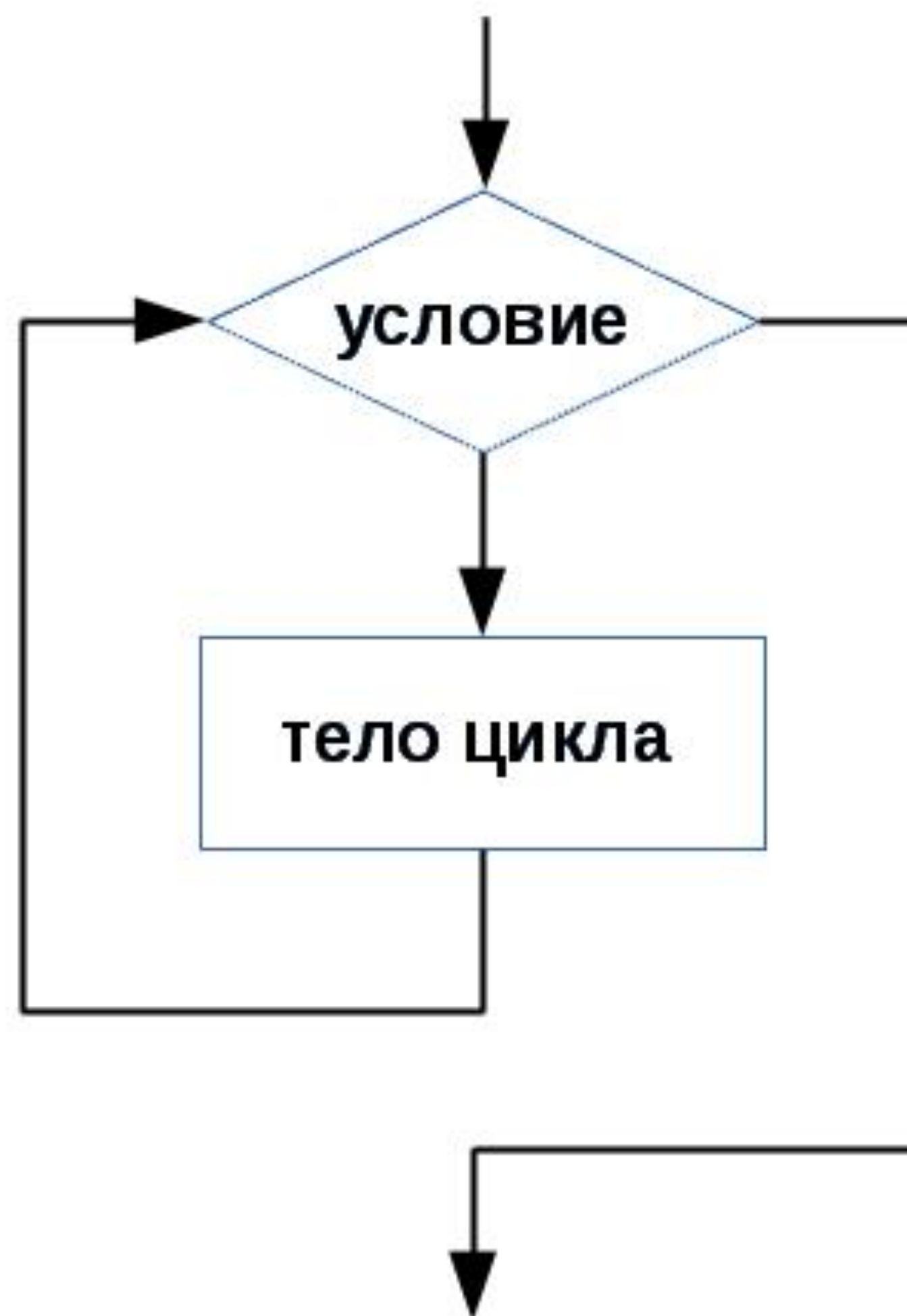


Кружок по программированию на Python

Цикл while

- Отличия от цикла for
- Применимость в каждом конкретном случае
- Инструкции управления циклом

Цикл





Цикл `while`

`while` *Условие1*:

 Блок инструкций 1

 Отступ!

Отступы обозначают
ветвление в программе.



Цикл `while`

`while` **Условие1**:

 Блок инструкций 1

 Отступ!

• Например:

```
x = 5
```

```
while x>0:
```

```
    print(x)
```

```
    x -= 1
```

5
4
3
2
1

Отступы обозначают
ветвление в программе.



Цикл `while`

`while` **Условие1**:

■ Блок инструкций 1

↪ Отступ!

- Однако!

```
x = 5
```

```
while x>0:
```

```
    x -= 1
```

```
    print(x)
```

4
3
2
1
0

Отступы обозначают ветвление в программе.



Знаки отношений

$<$	Меньше
$>$	Больше
$<=$	Меньше или равно
$>=$	Больше или равно
$=$	Равно
\neq	Не равно



Логические высказывания

Конъюнкция	И, \wedge	and
Дизъюнкция	Или, \vee	or
Отрицание	Не, \neg	not



Приоритет операций

- Приоритет :
 1. отношения (<, >, <=, >=, ==, !=)
 2. not (отрицание, «не»)
 3. and (конъюнкция, «и»)
 4. or (дизъюнкция, «или»)



Цикл `while`

`while` **Условие1**:

■ Блок инструкций 1

↪ Отступ!

Отступы обозначают ветвление в программе.

- Однако в цикле `while` может стоять и числовое выражение тоже:

```
x=3           3
while x-1:    2
    print(x)
    x-=1
```

Цикл будет выполняться, пока условие цикла не станет равным нулю.



Цикл `while`

`while` **Условие1**:

 Блок инструкций 1

`else`:

■ Блок инструкций 2

↑ Отступ!

Отступы обозначают ветвление в программе.

В случае завершения цикла выполнится блок инструкций `else` (даже без захода в цикл!)



Инструкции управления циклом

while **Условие1**:

■ Блок инструкций 1

↪ Отступ!

```
while i <= n ** 0.5:
```

```
    if n % i == 0:
```

```
        k = False
```

```
        break
```

```
    i+=1
```

Отступы обозначают ветвление в программе.

Инструкция `break` позволяет осуществить полный выход из цикла



Инструкции управления циклом

while **Условие1**:

■ Блок инструкций 1

↑ Отступ!

x = -10

while x <= 10:

if x == -1:

 x+=1

 continue

 print("{:+7.4f}".format(1/(x+1)))

 i+=1

$$y = \frac{1}{x + 1}$$

Отступы обозначают ветвление в программе.

Инструкция continue позволяет осуществить выход из текущей итерации



Задача 1

- По данному натуральному числу N распечатайте все квадраты натуральных чисел, не превосходящие N , в порядке возрастания.
- Вводится только натуральное число N .



Задача 1

- По данному натуральному числу N распечатайте все квадраты натуральных чисел, не превосходящие N , в порядке возрастания.

```
n=int(input())
i=1
while i <n**0.5:
    print(i**2)
    i+=1
```



Задача 1

- По данному натуральному числу N распечатайте все квадраты натуральных чисел, не превосходящие N , в порядке возрастания.

```
n=int(input())  
for i in range(1, int(n**0.5)+1):  
    print(i**2)
```




Задача 2

- Вклад в банке составляет x рублей. Ежегодно он увеличивается на p процентов, после чего дробная часть копеек отбрасывается. Определите, через сколько лет вклад составит не менее y рублей.
- Программа получает на вход три натуральных числа: x , p , y .
- Программа должна вывести одно целое число



Задача 2

```
x = int(input())*100
p = int(input())
y = int(input())*100
k = 0
while x < y:
    x += p*x//100
    k += 1
print(k)
```



Задача 3

- Дано натуральное число N . Выведите слово YES, если число N является точной степенью двойки, или слово NO в противном случае.

Операцией возведения в степень пользоваться нельзя!



Задача 3

```
n = int(input())
i = 1
while i<=n:
    if i==n:
        print("YES")
        break
    i*=2
else:
    print("NO")
```



Задача 4

- Последовательность состоит из натуральных чисел и завершается числом 0. Определите, какое количество элементов этой последовательности, равны ее наибольшему элементу.
- Вводится последовательность целых чисел, оканчивающаяся числом 0 (само число 0 в последовательность не входит, а служит как признак ее окончания).



Задача 4

```
n = int(input())
Max = n
i=0
while n:
    if Max<n:
        Max = n
        i = 0
    if Max == n:
        i = i+1
    n = int(input())
print(i)
```



Задача 4.1

- Последовательность состоит из натуральных чисел и завершается числом 0. Определите среднее значение этой последовательности.
- Вводится последовательность целых чисел, оканчивающаяся числом 0 (само число 0 в последовательность не входит, а служит как признак ее окончания).



Задача 4.1

```
s = int(input())
count = 1
while(True):
    i = int(input())
    if i == 0:
        break
    s += i
    count += 1

print(s/count)
```




Задача 5

- Последовательность Фибоначчи определяется так:

$$\varphi_0 = 0, \varphi_1 = 1, \dots, \varphi_n = \varphi_{n-1} + \varphi_{n-2}$$

По данному натуральному числу n определите n -е число Фибоначчи φ_n .



Задача 5

```
n=int(input())
```

```
a1=0
```

```
a=0
```

```
b=1
```

```
while n!=1:
```

```
    n-=1
```

```
    a1=a
```

```
    a=b
```

```
    b=a1+b
```

```
print(b)
```



Задача 5.1

- Последовательность Фибоначчи определяется так:

$$\varphi_0 = 0, \varphi_1 = 1, \dots, \varphi_n = \varphi_{n-1} + \varphi_{n-2}$$

Дано натуральное число A . Определите, каким по счету числом Фибоначчи оно является, то есть выведите такое число n , что

$$\varphi_n = A$$

Если A не является числом Фибоначчи, выведите число -1 .



Задача 5.1

```
a=0
```

```
i=1
```

```
b=1
```

```
while n!=b:
```

```
    a += b
```

```
    a, b = b, a
```

```
    i += 1
```

```
    if n<b:
```

```
        i -= 1
```

```
        break
```

```
print(i)
```



Фоксфорд
Кружки

Спасибо за внимание!