Программирование на языке Python

§ 57. Циклические алгоритмы

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».

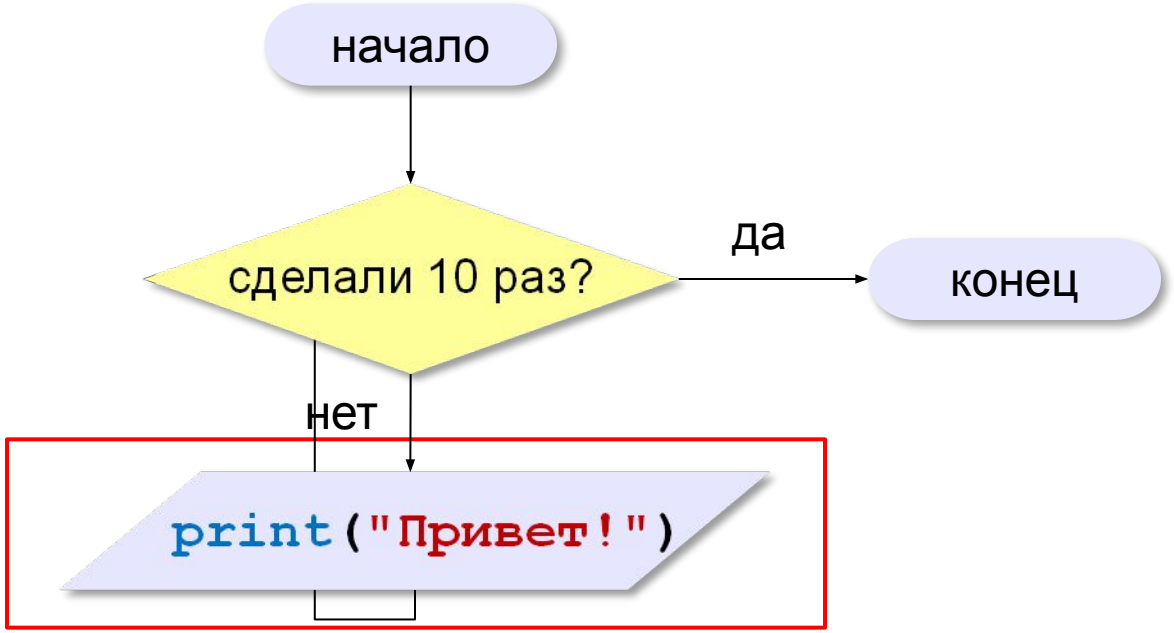
Можно ли решить известными методами?

Повторения в программе

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

Что плохо?

Блок-схема цикла



тело цикла

Как организовать цикл?

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

результат операции автоматически сравнивается с нулём!

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

? 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
```

n	счётчик
1234	0

? Как отсечь последнюю цифру?

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

? Как увеличить счётчик на 1?

```
счётчик = счётчик + 1
```

```
счётчик += 1
```

Считаем цифры

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!

Максимальная цифра числа

Задача. Определить **максимальную цифру** в десятичной записи целого положительного числа, записанного в переменную **n**.

```
n = int(input())
M = -1
while n > 0:
    d = n % 10
    if d > M:
        M = d
    n = n // 10
print( M )
```

последняя цифра

пока остались цифры

поиск максимума

? Что плохо!

отсечь последнюю цифру

Цикл с условием

При известном количестве шагов:

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

Зацикливание:

```
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

Алгоритм Евклида

Алгоритм Евклида. Чтобы найти НОД двух натуральных чисел, нужно вычитать из большего числа меньшее до тех пор, пока они не станут равны. Это число и есть НОД исходных чисел.

НОД(14,21) = НОД(14,7) = НОД(7, 7) = 7

```
пока a != b:
  если a > b:
    a -= b  # a = a - b
  иначе:
    b -= a  # b = b - a
```

```
while a != b:
  if a > b:
    a -= b
  else:
    b -= a
```

НОД(1998,2) = НОД(1996,2) = … = НОД(2, 2) = 2

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**