

## Программирование циклических алгоритмов

*Ключевые слова:*

- цикл `while`
- цикл `for`

### 1.5.1. Программирование циклов с известным условием продолжения работы

В языке Python цикл с заданным условием продолжения работы записывается с помощью оператора `while` (в переводе с английского языка — «до тех пор, пока»). Общий вид оператора:

```
while <условие>:
    <операторы>
```

Здесь:

<условие> — логическое выражение, условие продолжения работы цикла; пока оно истинно, выполняется тело цикла; если условие ложно с самого начала, тело цикла не выполнится ни разу;

<операторы> — один или несколько операторов, с помощью которых записано тело цикла.

Запишем на языке Python рассмотренный в п. 2.4.3 (пример 14) учебника для 8 класса алгоритм получения частного  $q$  и остатка  $r$  от деления натурального числа  $x$  на натуральное число  $y$  без использования операции деления.

```
# Программа 14
print('Частное и остаток')
x = int(input('Введите делимое x>>'))
y = int(input('Введите делитель y>>'))
r = x
q = 0
while r >= y:
    r = r - y
    q += 1
print('Частное q =', q)
print('Остаток r =', r)
```

Выполните программу при  $x = 25$  и  $y = 4$ . Каким будет результат выполнения программы при  $x = -10$  и  $y = 3$ ? Как вы можете объяснить этот результат?

## 1.5.2. Программирование циклов с известным условием окончания работы

В языке Python нет специального оператора для записи циклов с заданным условием окончания работы, но его можно организовать с помощью цикла **while**:

```
while True:
    <операторы>
    if <условие>: break
```

Цикл

```
while True:
    <операторы>
```

будет выполняться бесконечно, потому что условие **True** всегда истинно. Выйти из такого цикла можно только с помощью специального оператора **break** (в переводе с англ. — «прервать»).

Запишем на языке Python рассмотренный в п. 2.4.3 (пример 17) учебника для 8 класса алгоритм решения задачи о графике тренировок спортсмена.

```
# Программа 15
print('График тренировок')
i = 1; x = 10
while True:
    i += 1
    x = x + 0.1 * x
    if x >= 25: break
print ('Начиная с ', i, '-го дня спортсмен будет
пробежать 25 км' sep='')
```

### 1.5.3. Программирование циклов с известным числом повторений

Цикл с известным числом повторений в языке Python записывается с помощью оператора **for** (в переводе с английского — «для»). Его общий вид:

```
for <параметр> in range(k, n, m):  
    <операторы>
```

Здесь:

<параметр> — переменная целого типа;

range() — функция, описывающая необходимое количество повторов тела цикла; в скобках может быть указано от одного до трёх чисел:

- одно число ( $n$ ) указывает на то, что нужно последовательно перебрать все целые числа от 0 до  $n - 1$ ;
- два числа ( $k, n$ ) говорят о том, что нужно последовательно перебрать все целые числа, находящиеся в диапазоне от  $k$  (начальное значение) до  $n - 1$  (конечное значение);
- три числа ( $k, n, m$ ) указывают на то, что параметр должен изменяться от  $k$  до  $n - 1$  с шагом, равным  $m$ ;

<операторы> — один или несколько операторов, составляющих тело цикла.

При выполнении цикла **for** с тремя параметрами после каждого выполнения тела цикла происходит увеличение параметра

цикла на шаг  $m$ . Если  $k = n$  или  $k > n$ , цикл не выполнится ни разу.

Запишем на языке Python рассмотренный в п. 2.4.3 (пример 19) учебника для 8 класса алгоритм вычисления степени с натуральным показателем  $n$  для любого вещественного числа  $a$ .

```
# Программа 16  
print('Возведение в степень')  
a = float(input('Введите основание a>>'))  
n = int(input('Введите показатель n>>'))  
y = 1  
for i in range(n):  
    y = y * a  
print('y=', y)
```

Здесь параметр цикла  $i$  будет изменяться от 0 до  $n - 1$  включительно, следовательно, тело цикла выполнится ровно  $n$  раз.

#### 1.5.4. Различные варианты программирования циклического алгоритма

Для решения одной и той же задачи могут быть предложены разные алгоритмы, на основе которых могут быть разработаны разные программы. Вы могли убедиться в этом, программируя ветвления. Рассмотрим пример, показывающий, что реализация цикла может быть запрограммирована разными способами.

**Пример.** Напишем программу, в которой осуществляется ввод целых чисел до тех пор, пока не будет введён ноль, и подсчёт количества введённых положительных и отрицательных чисел.

Так как здесь в явном виде задано условие окончания работы, то воспользуемся конструкцией **while True**:

```
# Программа 17
k1 = k2 = 0
while True:
    n = int(input('Введите целое число>>'))
    if n > 0:
        k1 += 1
    if n < 0:
        k2 += 1
    if n == 0: break

print ('Введено:')
print ('положительных чисел -', k1)
print ('отрицательных чисел -', k2)
```

Имеющееся условие окончания работы можно достаточно просто преобразовать в условие продолжения работы — работа продолжается, пока *n* не равно 0, значит, мы можем воспользоваться оператором **while** с таким условием:

```
# Программа 18
k1 = k2 = 0
n = int(input('Введите целое число>>'))
while n != 0:
    if n > 0:
        k1 += 1
    if n < 0:
        k2 += 1
    n = int(input('Введите целое число>>'))
print('Введено:')
print('положительных - ', k1);
print('отрицательных - ', k2)
```

В рассмотренном примере число повторений тела цикла заранее неизвестно, и поэтому оператор **for** здесь применить нельзя. Если число повторений тела цикла известно, то лучше воспользоваться оператором **for**. Вместе с тем любая задача, в которой число повторений тела цикла определено заранее, может быть запрограммирована с помощью любого из двух рассмотренных выше циклов.

## САМОЕ ГЛАВНОЕ

В языке Python имеются два вида операторов цикла: **while** (цикл с условием) и **for** (цикл с параметром).

Цикл с условием выполняется до тех пор, пока некоторое условие (условие продолжения работы цикла) не станет ложным. Если условие в заголовке цикла ложно с самого начала, тело цикла не выполнится ни разу.

Если условие в заголовке цикла всегда остаётся истинным, цикл работает бесконечно. Для досрочного выхода из цикла используют оператор **break**.

Цикл с параметром применяют тогда, когда количество повторений цикла известно заранее или может быть вычислено до

начала цикла. Переменная, изменение которой определяет работу цикла, называется параметром (переменной) цикла. При вызове функции `range()` указывают от одного до трёх параметров: начальное значение, значение-ограничитель (не входящее в диапазон) и шаг изменения переменной цикла; обязательный параметр — значение-ограничитель.

Если число повторений тела цикла известно, то лучше воспользоваться оператором **for**; в остальных случаях используется оператор **while**.

## Вопросы и задания

1. Проанализируйте работу программы:

```
x = 1
y = 1
while x < 5:
    y *= 2
    x += 1
```

Ответьте на вопросы.

- Сколько раз выполнится тело цикла?
- Какое значение примет  $x$  после завершения программы?
- Какое значение примет  $y$  после завершения программы?
- Сколько раз выполнится тело цикла, если заменить условие на  $x \leq 5$ ?
- Сколько раз выполнится тело цикла, если заменить условие на  $x \geq 5$ ?
- Сколько раз выполнится тело цикла, если заменить условие на  $x > 0$ ?
- Что произойдёт, если из тела цикла убрать команду  $x += 1$ ?
- Сколько раз выполнится тело цикла, если заменить команду  $x += 1$  на  $x += 2$ ?
- Сколько раз выполнится тело цикла, если заменить команду  $x += 1$  на  $x -= 1$ ?

2. Дана последовательность операторов:

```
a = 1; b = 2
while a + b < 8:
    a += 1
    b += 2
    s = a + b
```

3. Определите значения переменных  $s$  и  $i$  после выполнения фрагмента программы.

- |    |  |    |  |    |  |
|----|--|----|--|----|--|
| a) | <pre>s = 0 i = 0 while i &lt; 5:     i += 1     s += i</pre> | б) | <pre>s = 0 i = 0 while i &lt; 5:     i += 1     s += i</pre> | в) | <pre>s = 0 i = 2 while i &gt; 1:     s = s + 1 / i     i = i - 1</pre> |
|----|--|----|--|----|--|

4. Определите значение переменной  $s$  после выполнения фрагмента программы при указанных значениях  $a$ . Составьте таблицы значений переменных.

```
p = a
s = 0
while p > 0:
    s = s + p % 10
    p = p // 10
```

- а)  $a = 23$ ;  
б)  $a = 32$ ;  
в)  $a = 109$ ;

5. От программы, записанной на алгоритмическом языке, перейдите к записи программы на языке Python. Определите, что будет выведено в результате работы программы. Составьте таблицу значений переменных.

```
алг
  цел s, k
нач
  s := 0
  k := 0
  нц пока k < 6
    s := s + 2
    k := k + 1
  кц
  вывод s
кон
```

6. Запишите на языке Python фрагмент программы, соответствующий блок-схеме. Определите значения переменных  $k$  и  $t$  после её выполнения. Составьте таблицу значений переменных.

