Python. Начала программирования

Циклические алгоритмы

В качестве примера разработки циклических алгоритмов рассмотрим задачу табулирования функции одного переменного, например, $y = x^2$ для пяти узлов.

Цикл с условием продолжения

Представим первоначальное описание алгоритма на естественном языке:

1. Начало
2. Положить k = 0
3. Пока k < 5 повторить
3.1. Ввести значение x
3.2. Вывести значения x и x * x
3.3. Положить k = k + 1
4. Конец

Представим альтернативную версию описания алгоритма на естественном языке:

```
begin
   k = 0
   while k < 5
      input x
      print x, x * x
       k = k + 1
end
```

Представим программу для вещественной арифметики:

```
k = 0
while k < 5 :
  x = float(input('? '))
  print(x, x * x)
  k += 1
```

Результат работы программы:

```
? 1
1.0 1.0
? 2
2.0 4.0
```

```
? 3
3.0 9.0
? 4
4.0 16.0
? 5
5.0 25.0
```

Представим альтернативную версию этой программы:

```
k = 0
while True :
    x = float(input('? '))
    print(x, x * x)
    k += 1
    if k == 5 : break
```

Результат работы программы:

```
? 1
1.0 1.0
? 2
2.0 4.0
? 3
3.0 9.0
? 4
4.0 16.0
? 5
5.0 25.0
```

Как видим, использование в теле цикла инструкции ***break*** позволяет осуществить выход из цикла по достижении заданного количества итераций.

Представим первоначальное описание алгоритма на естественном языке для краткой формы счетного цикла:

1. Начало
2. От k = 1 до k = 5 повторить
2.1. Ввести значение x
2.2. Вывести значения x и x * x
3. Конец

Представим альтернативную версию описания алгоритма на естественном языке:

```
begin
  for k = 1 to k = 5 do
    input x
```

```
    print x, x * x
end
```

Представим программу для вещественной арифметики:

```
for k in range(5) :
    x = float(input('? '))
    print(x, x * x)
```

Результат работы программы:

```
? 1
1.0 1.0
? 2
2.0 4.0
? 3
3.0 9.0
? 4
4.0 16.0
? 5
5.0 25.0
```

Напомним, что циклы ***for*** в языке Python начинаются со строки заголовка, где указывается переменная для присваивания, а также объект последовательности, обход которого будет выполнен. Вслед за заголовком следует блок (обычно с отступами) инструкций, называемый по традиции телом цикла.

Когда интерпретатор выполняет цикл ***for***, он поочередно, один за другим, присваивает элементы объекта последовательности переменной цикла и выполняет тело цикла для каждого из них.

Функция ***range()*** с одним аргументом генерирует список целых чисел в диапазоне от нуля до указанного в аргументе значения, не включая его.

Если функции передать два аргумента, первый будет рассматриваться как нижняя граница диапазона. Необязательный третий аргумент определяет шаг (по умолчанию он равен 1) – в этом случае величина шага будет добавляться при вычислении каждого последующего значения.