циклы с условием



Понятие цикла



Цикл – это повторяющаяся последовательность действий.

Циклический алгоритм – это алгоритм, который содержит повторяющуюся последовательность действий.





Итерация от лат. iteratio повторение

Цикл с условием (while)



Синтаксис цикла while в простейшем случае выглядит так:

while условие:

блок инструкций

Цикл с условием while позволяет выполнить одну и ту же последовательность действий, пока проверяемое условие истинно.

При выполнении цикла while сначала проверяется условие. Если оно ложно, то выполнение цикла прекращается, и управление передаётся на следующую инструкцию после тела цикла while.

Если условие истинно, то выполняются все инструкции из блока, после чего условие проверяется снова, и снова выполняется блок инструкций. Так продолжается до тех пор, пока условие будет истинно. Как только условие становится ложным, работа цикла завершается, и управление передаётся следующей инструкции после цикла.

Как правило, цикл while используется, когда невозможно определить точное количество проходов исполнения цикла.

Рассмотрим пример



```
Сколько всего знаков * будет выведено после исполнения фрагмента программы:
```

```
i = 0
while i < 5:
    print('*')
    if i % 2 == 0:
        print('**')
    if i > 2:
        print('***')
    i = i + 1
```

```
i = 0
     while i < 5:
         print('*')
         if i % 2 == 0:
             print('**')
         if i > 2:
             print('***')
         i = i + 1
Оболочка
```

Изменения в работе цикла



break – оператор завершения цикла

```
i=0
while i<5:
    a=int(input())
    b=int(input())
    if (a==0) and (b==0):
        break #досрочно завершаем цикл
    print (a*b)
    i+=1</pre>
```

Если вместо **break** вписать **quit()**, то будет завершена вся программа.

continue – оператор перехода к следующей итерации цикла (если она есть)

```
i=0
while i<5:
   a=int(input())
   b=int(input())
   if (a==0) and (b==0):
     break #досрочно завершаем
ПИКЛ
   if (a==0) or (b==0):
       continue #переходим к
следующей итерации
   print (a*b)
   i+=1
```

Обработка потока данных



Напишите программу, которая определяет сумму последовательности целых чисел. Программа получает на вход целые числа, количество введённых чисел не известно, последовательность чисел заканчивается числом о (о – признак окончания ввода, не входит в последовательность). Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

В данной задаче не нужно сохранять все данные в памяти, а можно добавлять их к сумме по одному.

Пусть х – последнее введённое число, s – сумма.

Бесконечные циклы



while True:

блок инструкций

Поскольку условие **True** всегда истинно, такой цикл никогда не завершится. Но завершить его всё-таки можно, если применить специальный оператор **break**.

Рассмотрим на примере нахождения суммы чисел потока данных.

В некоторых задачах используются циклы, условие в которых всегда истинно.

s=0
x=int(input('x= '))
while x!=0:
 s=s+x
 x=int(input('x= '))
print("s= ", s)

s=0
while True:

x=int(input('x= '))

if x==0: break

s+=х условие выхода

print('s=', s)

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

Наименьшее число в последовательности



```
x=int(input('x= '))
if x!=0:
  min=x
while x!=0:
  x=int(input('x= '))
  if x<min and x!=0:
    min=x
print('min=', min)
```

Напишите программу, которая в последовательности натуральных чисел определяет минимальное число. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом о (о – признак окончания ввода, не входит в последовательность). Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30000.