

Циклы с условиями

На прошлом занятии мы разобрали условные конструкции. А что если мы хотим повторять какие-то операции до тех пор, пока истинно какое-то условие?

Для этого в Java есть циклы `while`.

Общий синтаксис цикла `while`:

```
while(условие){  
    делать_пока_правда  
}
```

Попробуем решить следующую задачу:

На вход подаются целые числа, последнее из них отрицательное.

Требуется вывести сумму всех чисел.

```
import java.util.Scanner;  
  
public class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        int a = sc.nextInt();  
        int sum = 0;  
  
        while(a>0){  
            sum+=a;  
            a = sc.nextInt();  
        }  
  
        sum+=a;  
        System.out.println(sum);  
    }  
}
```

Очевидно, что условие цикла — это `a>0`, т. к. после чтения отрицательного числа, нам читать числа больше не надо.

Чтобы посчитать сумму, надо ввести какую-то переменную. Мы назвали её `sum`.

Внутри цикла `while` мы прибавляем к переменной `sum` каждое прочитанное значение.

После завершения у нас к сумме не прибавлено последнее число. Поэтому мы прибавляем его уже после тела цикла.

Повторяющийся код — это, безусловно, плохо. Т.к. гарантированно на вход подадут хотя бы одно число, то хотелось бы сначала числа считывать, прибавлять их в сумму, а когда придёт отрицательное, просто закончить выполнение цикла.

Такой цикл есть. Он называется цикл с постусловием `do...while`

```
do{  
    тело_цикла;  
}while(условие);
```

Циклы с постусловием работают по следующему принципу: тело цикла гарантированно выполняется хотя бы один раз. Если условие верно, то тело цикла повторяется. Обратите внимание, что после while(условие) ставится «;»

Т.о. цикл с постусловием отличается от простого цикла с условием тем, что его тело гарантированно выполнится хотя бы один раз.

Перепишем нашу задачу, используя цикл с постусловием.

```
import java.util.Scanner;  
  
public class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int a = sc.nextInt();  
        int sum = 0;  
        do{  
            a = sc.nextInt();  
            sum+=a;  
        }while(a>0);  
        System.out.println(sum);  
    }  
}
```

Как видите, такой код не содержит повторяющихся команд. Поэтому для каждой отдельной ситуации надо думать, какой цикл больше подойдёт.

Вывод вещественных чисел.

Раньше, работая с вещественными числами, мы вызывали команду:

```
System.out.println(d);
```

В консоль при этом выводилось число с непонятным числом знаков после запятой. Очень часто нам требуется вывести число с точным количеством знаков после запятой.

В следующем примере будет организован вывод вещественного числа с двумя и четырьмя знаками после запятой.

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        double d = 1.0/3;

        System.out.printf("%.2f",d);

        System.out.printf("%.4f",d);

    }

}
```

В консоль выведется:

>>0.33

>>0.3333

Т.к. printf выводит число без переноса на новую строку, то нам необходимо вызвать System.out.println() для переноса на новую строку.

System.out.printf() вызывается следующим образом: сначала пишется строка «%.2f» (число после точки ставится число, соответствующее количеству знаков после запятой в выводе числа на консоль), а после запятой имя переменной, значение которой мы хотим вывести.