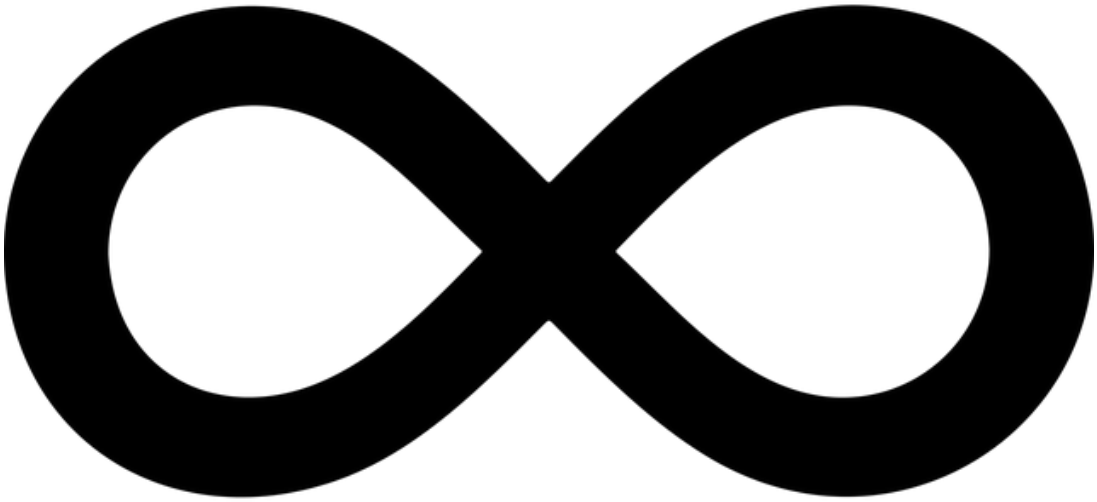
Цикл FOR — введение в Java 008



Каждый день я встаю в шесть утра и иду на завод к восьми, в семь вечера я уже снова дома, смотрю кино по телевизору, курю с соседом на лестничной клетке и в одиннадцать уже иду спать. Утром вторника повторяется всё то же самое, и в среду, и в четверг, и в пятницу.

Всё это можно описать, например, с помощью блока if else. Ничего сложного: если понедельник, то подъём; а если вдруг вторник, то тоже подъём; если среда, то подъём и т.д. и т.п. Но кусок кода повторяющийся, и для таких случаев придумали циклы, один из них — цикл FOR. Условно, с понедельника по пятницу выполняется следующий алгоритм действий: подъём, работа, телевизор, спать. Пример:

```
public class FourWeek {
    public static void main(String[] args) {
        int weekDays = 7;
        for (int i = 1; i < weekDays ; i++) {
            System.out.println("------------");
            System.out.println("Новый день");
            System.out.println("я встаю в шесть утра");
            System.out.println("иду на завод к восьми");
            System.out.println("я уже дома");
            System.out.println("смотрю кино по телевизору");
            System.out.println("курю с соседом на лестничной клетке");
            System.out.println("иду спать");
            System.out.println("День подошёл к концу");
            System.out.println("-------------");
        }
    }
}
```

Область видимости

Мы можем использовать несколько циклов подряд:

```
public class Main {
    public static void main(String[] args) {

        int a = 0;
        for (int i = 0; i < 10; i++) {
            a++;
```

```
        }

        for (int i = 0; i < 20; i++) {
            a++;
        }
        System.out.println(a);
        System.out.println(i);
    }
}
```

При этом операции над переменной **a** будут проходить в каждом цикле, потому что **a** принадлежит к *области видимости* всего метода. А вот попытка вывести на консоль **i** приведёт к ошибке. Переменная **i**, в данном случае, объявлялась только в циклах, и за пределами самого цикла не видна. Именно поэтому мы и смогли декларировать эту переменную дважды. Потому что в самом методе её как бы и нет.

Мы можем экранировать переменные просто скобками:

```
public class Main {
    public static void main(String[] args) {
        int a = 0;
        {
            int b = 7;
            a = a + b;
        }
        System.out.println(a);
        System.out.println(b);
    }
}
```

В данном примере попытка "распечатать" **b** приведёт к ошибке. Декларация переменной произошла в экранированном участке кода. Но операции с **a** программа запомнит, потому что программа декларировала **a**, и закрытый участок кода для **a** — просто часть программы. Сверху вниз смотреть можно, а снизу вверх смотреть нельзя.

И снова к самому циклу for

Это могут быть и арифметические операции:

```
public class FourTest {
    public static void main(String[] args) {
        int a = 20;
        int b = 11;
        int c;
        for (int i = 0; i < a; i++) {
            b = b + i;
            c = i + 1;
            System.out.println("c " + c);
        }
        System.out.println("b " + b);
    }
}
```

В определённом или в заданном интервале мы изменяем переменную по определённой формуле.

```
for (начало; условие; шаг) {
  // ... тело цикла ... в котором можно выполнять одну, две, три или даже очень много операций.
}
```

```
for (initialization condition; testing condition; increment/decrement){
    statement(s)
}
```

For — это цикл, в котором "тело" выполняется заданное количество раз.

Начало, условие, шаг

Начало, условие, шаг надо запомнить не как "Отче наш". А "Отче наш" заменяется теперь на "начало, условие, шаг". Ещё один пример, где мы работаем только с одной переменной — отсчёт обратного времени. Именно этот код используется всеми космодромами при запуске ракет в космос:

```
public class  TickTackStart {
    public static void main(String[] args) {
        for (int i = 10; i > 0; i--)
            System.out.println("i= " + i);
    }
}
```

Начало может быть любым. Другой переменной int i = b или любым (обычно int) числом. Например, int i = 7.

Условие задаёт рабочий интервал между началом и концом программы.

Шаг может быть любым. Минус два, плюс два, минус 100 или плюс 5000. Шаг мы задаём сами. В шаге мы можем складывать, вычитать, умножать и даже делить: i = 2*i.

```
        for (int i = 0; i < 20; i = 2*i){
            System.out.println(i);
        }
```

И это уже домашнее задание

1. Определите на глаз, что будет выдано на консоли после запуска последнего куска кода?
2. Даны целые числа A и B, где B > 0. Вывести B раз число A.
3. Вывести в порядке возрастания все числа между A и B, где a = 5, а B = 17.
4. Сколько раз будет выведено сообщение на экран?

   ```
   for (int i = 3; i < 10; i ++){
       System.out.println(i);
   }
   ```

5. "Нарисуйте" (выведите на консоль) треугольник из звёздочек, используя только одну звёздочку и то, что только что изучили:

```
*
**
***
****
*****
******
```

```
*******
********
*********
**********
```

Дополнительные материалы

https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html