

Переменные-ссылки

Java Syntax
2 уровень, 6 лекция

ОТКРЫТА

— Привет, Амиго, это снова я — Элли. Прости, что постоянно так говорю, так просто принято на Земле в 31 веке. Я хочу рассказать тебе об особенностях переменных-ссылок и передаче в функции(методы) аргументов-ссылок.

— Я готов.

— Отлично, тогда слушай. Переменные-ссылки — это переменные всех типов, кроме примитивных. Такие переменные содержат в себе только адрес объекта (ссылку на объект).

— Переменные примитивных типов хранят в себе значения, а переменные типов-классов хранят ссылку на объекты этого же класса, ну или хранят null. Я все правильно говорю?

— Абсолютно верно.

— Так что же это такое — эта ссылка?

— Объект и ссылка на него связаны, примерно, как человек и его телефонный номер. Телефонный номер не является человеком, но номер можно использовать, чтобы звонить человеку, спрашивать у него какую-то информацию, руководить им или давать команды. Ссылка тоже используется для взаимодействия с объектом. Все объекты взаимодействуют друг с другом при помощи ссылок.

— Это как если бы люди общались друг с другом по телефону?

— Именно. При присваивании примитивного объекта, его значение копируется (дублируется). При присваивании же ссылочной переменной, копируется только адрес объекта (телефонный номер), сам же объект при этом не копируется.

— Ок, понял.

— Ссылка даёт ещё одно преимущество: можно передать ссылку на объект в какой-нибудь метод, и этот метод будет в состоянии модифицировать (изменять) наш объект используя ссылку на него, вызывая его методы и обращаясь к данным внутри объекта.

Пример 1

Тут значение m и n не меняется.

```
1 public class References
2 {
3     public static void main (String[] args)
4     {
5         int m = 5;
6         int n = 6;
7
8         System.out.println("M=" +m+ " N=" + n);
9         swap( m, n );
10        System.out.println("M=" +m+ " N=" + n);
11    }
12
13    private static void swap( int a, int b )
14    {
15        int c = a;
16        a = b;
17        b = c;
```

И вот почему.

Данный код аналогичен коду слева

```
1 public class References
2 {
3     public static void main (String[] args)
4     {
5         int m = 5;
6         int n = 6;
7
8         System.out.println("M=" +m+ " N=" + n);
9         int a = m, b = n;
10
11        int c = a;
12        a = b;
13        b = c;
14
15        System.out.println("M=" +m+ " N=" + n);
16    }
17 }
```

```
18 }
19 }
```

— Переменным **a** и **b** присваиваются только значения 5(m) и 6(n) соответственно, **a** и **b** ничего не знают про (никак не влияют на) **m** и **n**.

— Честно говоря, я понял, что я ничего не понял. Можно еще пару примеров?

— В случае со ссылочным типом, можно было сделать так:

Пример 2

Тут данные объектов меняются

```
1 public class Primitives
2 {
3     public static void main(String[] args)
4     {
5         Student olga = new Student();
6         olga.name = "Olga";
7         olga.age = 21;
8
9         Student vera = new Student();
10        vera.name = "Veronika";
11        vera.age = 15;
12
13        System.out.println("Olga is " + olga.age);
14        System.out.println("Vera is " + vera.age);
15
16        ageSwap(olga, vera);
17
18        System.out.println("Olga is " + olga.age);
19        System.out.println("Vera is " + vera.age);
20    }
21
22    private static void ageSwap(Student a,
23                                Student b)
24    {
25        int c = a.age;
26        a.age = b.age;
27        b.age = c;
28    }
29
30    static class Student
31    {
32        String name;
33        int age;
34    }
35 }
```

И вот почему.

Данный код аналогичен коду слева

```
1 public class Primitives
2 {
3     public static void main(String[] args)
4     {
5         Student olga = new Student();
6         olga.name = "Olga";
7         olga.age = 21;
8
9         Student vera = new Student();
10        vera.name = "Veronika";
11        vera.age = 15;
12
13        System.out.println("Olga is " + olga.age);
14        System.out.println("Vera is " + vera.age);
15
16        Student a = olga, b = vera;
17
18        int c = a.age;
19        a.age = b.age;
20        b.age = c;
21
22        System.out.println("Olga is " + olga.age);
23        System.out.println("Vera is " + vera.age);
24    }
25
26
27
28
29
30    static class Student
31    {
32        String name;
33        int age;
34    }
35 }
```

— Переменным **a** и **b** присваиваются ссылки на **olga** и **vera** соответственно, **a** и **b** меняют значения внутри объектов **olga** и **vera**.

— Так классы внутри других классов ещё можно объявлять? Круто!

— Но в остальном не очень-то и понятно.

— Не все сразу.

< Предыдущая

×2