Статические переменные

Привет!

Этот материал познакомит вас со статическими переменными. Вы уже изучили классы и объекты и увидели, что у объектов могут быть переменные, каждая из которых может иметь своё значение.

Теперь представим, что нам понадобилось иметь какую-то переменную не у отдельных объектов, а у всего класса. Например, в интернет-магазине нужно посчитать общее количество корзин — объектов класса Basket, — которые были созданы.

Можно учитывать это в коде, в котором корзины создаются. То есть если есть одно место, где написано:

```
Basket basket = new Basket();
```

то можно просто создать переменную, которая в этот момент будет увеличиваться:

```
int basketCount = 0;
```

и после создания корзины писать:

```
basketCount = basketCount + 1;
```

Но если в коде есть другое место, где тоже создаются корзины, то с подсчётом будет всё сложнее. Например, есть специальный класс, объект которого создаётся, когда пользователь логинится (сессия пользователя):

```
public class Session {
      public Session() {
            //some code
      }
}
```

и в нём уже собранная пользователем корзина, например когда-то раньше или на другом компьютере, создаётся из строки и числа, сохранённых в базе данных:

```
Basket basket = new Basket("товары", 3462);
```

то уже не очень понятно, как считать количество созданных корзин. Можно было бы создать отдельный класс с переменной basketCount, создать на его основе объект и передавать его всюду, где могут создаваться корзины.

Но это очень неудобно, особенно если проект большой и корзины могут создаваться в разных его местах. Есть простой способ решить эту проблему — создать статическую переменную у класса Basket:

```
public static int count = 0;
```

В этом случае переменная будет доступна и без создания объектов. И увеличивать значение этой переменной можно в конструкторе:

```
count = count + 1;
```

То, что переменная является статической, означает, что она принадлежит не к объекту класса, а к самому классу. То есть она, по сути, будет общей для всех его объектов.

Обращаться к статическим переменным можно по имени класса, без создания объектов в любом месте нашего кода, в любом классе и в любом методе:

```
System.out.println(Basket.count);
```

К статическим переменным можно обращаться и через переменную, ссылающуюся на объект соответствующего класса:

```
Basket basket = new Basket();
System.out.println(basket.count);
```

Но так лучше не делать. К статическим переменным принято обращаться через имя класса, поскольку это повышает читабельность кода. Если вы видите в коде конструкцию вида:

```
System.out.println(Basket.count);
```

вам сразу понятно, что вы обращаетесь к статической переменной. Если же видите:

```
System.out.println(basket.count);
```

то непонятно, к статической переменной вы обращаетесь или к нестатической.

Кстати, в команде System.out.println *out* — это статическая переменная класса System. Она, как несложно догадаться, ссылается на некий объект, у которого

есть множество методов println, принимающих в качестве параметров самые разнообразные варианты значений.

Итоги

Вы познакомились со статическими переменными, а также узнали, что статическая переменная относится к классу, а не к объекту, и для всех объектов этого класса она имеет одно и то же значение в отличие от нестатической переменной. Методы тоже бывают статические, но об этом мы с вами поговорим в следующем видео.

Глоссарий

статическая переменная