

Классы и объекты

Привет!

В этом материале поговорим о ключевых понятиях объектно-ориентированного программирования — классах и объектах, о том, что они из себя представляют, для чего нужны и как их правильно использовать.

Давайте снова обратимся к коду корзины, который мы писали в предыдущих видео.

В коде мы описали действия, которые можно осуществлять с корзиной: добавлять в неё товары, очищать её, печатать её содержимое и проверять, есть ли в ней тот или иной товар.

Но есть одна проблема: по сути, корзина всего одна. А в настоящем магазине корзин может быть много — как минимум у каждого пользователя своя. В нашей же программе мы пока не можем создавать разные корзины.

Сделаем так, чтобы эта возможность появилась. Для этого придумали так называемые классы и объекты. Они нужны, чтобы отражать в программном коде сущности, которые есть в предметной области, например те же корзины.

Класс

01:10–04:34

Но давайте разбираться по порядку. Класс — это некая абстракция, которая описывает, какими могут быть объекты: их свойства и способы взаимодействия с ними.

Например, у нас есть корзина. Корзины разных пользователей должны быть разные: в них должны храниться разные списки товаров, разные цены, — но все они обладают общими свойствами: у каждой из них есть свойство «список товаров», даже если он пустой, и есть общая цена.

Кроме того, у всех корзин всех пользователей должно быть одинаковое поведение: у них всех должна быть возможность добавления в них товаров, очистки, распечатки списка товаров и так далее — то есть, по сути, одни и те же методы.

Обратите внимание, что у нас здесь уже есть класс `Basket`. Но мы им пользуемся из метода `main`. Чтобы создавать объекты на основе этого класса, то есть разные корзины, мы должны прежде всего отделить метод `main`. Создадим для него отдельный класс и туда его положим.

И второе, что нужно сделать — у всех методов и переменных убрать слово `static`. Оно было нужно для вызова их из метода `main`, который всегда `static`. Но это ключевое слово разберём подробно в следующих материалах.

Пока просто запомните, что для того чтобы переменную или метод можно было использовать в классе, на основе которого создаются разные объекты — в нашем случае разные корзины, — эти переменные и методы не должны быть обозначены как `static`.

Класс — это шаблон, по которому можно создавать реальные объекты. Класс — это, по сути, набор методов и переменных.

Они находятся в фигурных скобках, перед которыми написаны слово `class` и имя класса. Оно, так же как и имя метода, может быть произвольным и должно отражать суть того, что делает класс и какие объекты на его основе можно создавать.

Мы решили создать класс, отражающий состояние корзины пользователя в интернет-магазине, и так его и назвали — `Basket`. Имена классов в языке Java принято писать с большой буквы, и если имя класса состоит из нескольких слов, то каждое последующее слово пишется слитно и тоже с большой буквы — аналогично тому, как это делается в именах методов.

Обратите внимание, что раньше мы говорили только о переменных, передаваемых в методы и используемых в методах. Но переменные, расположенные на уровне класса, вне методов, тоже могут использоваться в методах этого же класса.

Итак, мы создали полноценный класс корзины, по структуре которого понятно, что из себя представляет корзина и какие действия с ней можно совершать. Теперь на его основе создадим объекты — то есть корзины конкретных пользователей. Для этого в методе `main` напишем:

```
Basket vasyaBasket = new Basket();
```

Что означает эта строчка кода? Во-первых, для создания объекта какого-то класса необходимо написать имя этого класса.

Затем пишется имя переменной. Оно, опять же, может быть произвольным, как имя любой переменной в коде, которую мы сами придумали, но крайне

желательно называть переменную таким образом, чтобы название однозначно соответствовало тому, что в этой переменной содержится. В данном случае это `vasyaBasket`, поскольку здесь рассматривается корзина Васи.

Объект класса

04:34–08:18

Далее — ключевое слово `new`. Оно означает создание объекта класса.

После снова пишется имя класса со скобками, говоря тем самым, что создаётся объект именно этого класса.

```
vasyaBasket.add("Молоко", 80);  
vasyaBasket.add("Хлеб", 40);
```

Теперь создадим вторую корзину `petyaBasket` и тоже с ней поработаем.

```
Basket petyaBasket = new Basket();  
petyaBasket.add("Лопата", 280);  
petyaBasket.add("Бочка", 2900);
```

Обратите внимание, что хоть эти две корзины созданы на основе одного класса, их содержимое совершенно разное. То есть каждая из них представляет собой объект со своими собственными значениями всех переменных.

Таким образом, можно создавать любое количество объектов одного класса. И это очень удобно.

Важно помнить и понимать, что классы, методы и объекты — это то, что отражает предметную область, с которой мы работаем. То есть неверно просто брать код и делить его на неосмысленные фрагменты, используя методы.

Нужно подумать над тем, чтобы эти фрагменты отражали предметную область и реальные действия, которые можно произвести с объектами или которые сами объекты могут произвести.

Методы и классы должны быть созданы таким образом, чтобы их можно было в коде использовать и вызывать многократно, — то есть они должны быть переиспользуемыми.

Итоги

08:18 — до конца видео

Вы познакомились с понятиями класса и объекта.

Вы узнали, что классы — это фрагменты кода, обозначенные ключевым словом «класс» и именем класса.

Классы состоят из методов и переменных. Они все находятся в фигурных скобках. И на основе классов можно создавать объекты — то есть переменные этого класса, внутри каждой из которых будет свой набор переменных конкретного объекта.

Глоссарий

Класс, ключевое слово `new`, ключевое слово `static`, объект