

## Immutable-классы

Привет!

Ранее мы говорили о геттерах и сеттерах. А что, если вы не хотите менять значения переменных в объекте после его создания, то есть хотите гарантировать неизменность состояния объекта?

Такое свойство классов называется **иммутабельностью**, то есть (в переводе с английского) неизменностью. О ней мы и поговорим.

По сути, если вы хотите сделать класс иммутабельным, вам нужно удалить сеттеры. И чтобы никогда никто не мог изменить переменные этого объекта, задаваемые лишь однажды через конструктор, вам нужно пометить эти переменные ключевым словом **final**.

```
public class Product {  
    private final String name;  
    private final int price;  
  
    public Product(String name, int price) {  
        this.name = name;  
        this.price = price;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getPrice() {  
        return price;  
    }  
}
```

```
}
```

Теперь любой программист, который столкнётся с вашим кодом, увидит, что эти переменные поменять не получится. Даже если он допишет сеттер, среда разработки подчеркнёт такое изменение красным, выдаст ошибку *Cannot assign a value to final variable 'name'* (Невозможно присвоить значение финальной переменной 'name') и не даст запустить программу.



```
1 public class Product {
2     private final String name;
3     private final int price;
4
5     public Product(String name, int price) {
6         this.name = name;
7         this.price = price;
8     }
9
10    public String getName() {
11        return name;
12    }
13
14    public void setName(String name) {
15        this.name = name;
16    }
17
18    public int
19    return
20    }
21 }
22
```

Обратите внимание на те моменты, которые важны для создания immutable-классов:

- все переменные таких классов должны быть помечены как `private`, чтобы их нельзя было изменить извне;
- они все должны быть помечены как `final`, чтобы их можно было установить лишь однажды в конструкторе при создании объекта;

- в таком классе не должно быть сеттеров — да и зачем они, если ничего нельзя изменять?

## Итоги

Итак, мы изучили понятие иммутабельности. Это свойство, которое позволяет делать объекты неизменяемыми.

## Глоссарий

**Иммутабельность** — это свойство класса, гарантирующее неизменность состояния объекта после его создания.