POJO-классы, геттеры и сеттеры

POJO-классы

00:00–01:41

Привет! Ранее мы в основном рассматривали методы классов, в которых содержится какая-то логика: проверка условий и изменение значений переменных, но у нас также встречались методы, которые просто возвращают какое-то значение.

Например, в классе GearBox есть метод getCurrentGear, который просто возвращает значение переменной gear.

```
public int getCurrentGear() {
    return gear;
}
```

Существуют так называемые **POJO-классы**, объекты которых выступают просто хранилищами каких-то данных. POJO расшифровывается как plain old java object, в дословном переводе с английского — «простой старый объект Java».

Под простым здесь имеется в виду отсутствие в классе сложной логики. В некоторых языках программирования такие классы называют data class — то есть классами с данными. Давайте создадим такой класс.

Представьте, что мы добавляем в корзину интернет-магазина товары, у которых есть ряд свойств — для простоты пока два свойства: название и цена. Логично для товара создать отдельный класс Product и задать для него эти два свойства: name и price.

```
public class Product {
    public String name;
```

```
    public int price;
}
```

У этого товара не будет никаких других свойств. А эти свойства нам нужно как-то устанавливать и получать. Мы, конечно, можем делать это напрямую, обращаясь к именам этих свойств через точку. Но так мы нарушим принцип инкапсуляции, и это в итоге может привести к снижению поддерживаемости нашего кода.

Геттеры и сеттеры

01:41–05:12

Если вы когда-нибудь захотите добавить дополнительную логику в этот класс, вы создадите методы, но весь остальной код, который использует этот класс, будет обращаться к переменным напрямую, и этот код вам тоже придётся переписать на использование этих методов.

Если же вы сразу создадите методы, через которые будете работать с этими свойствами, то переписывать весь остальной код, использующий ваш класс, не придётся. Поэтому принято создавать специальные методы, устанавливающие и возвращающие значения свойств класса, которые даже имеют специальные названия — **геттеры** и **сеттеры**, от слов **get** (получить) и **set** (установить).

Давайте пропишем эти методы.

```
public void setName(String name) {
    this.name = name;
}

public String getName() {
    return name;
}

public void setPrice(int price) {
```

```
    this.price = price;
}

public int getPrice() {
    return price;
}
```

И сделаем переменные private, чтобы их нельзя было изменять или получать извне, чтобы это можно было делать только через методы.

```
private String name;
private int price;
```

Можем добавить в конструктор оба параметра, чтобы при создании объекта сразу их устанавливать.

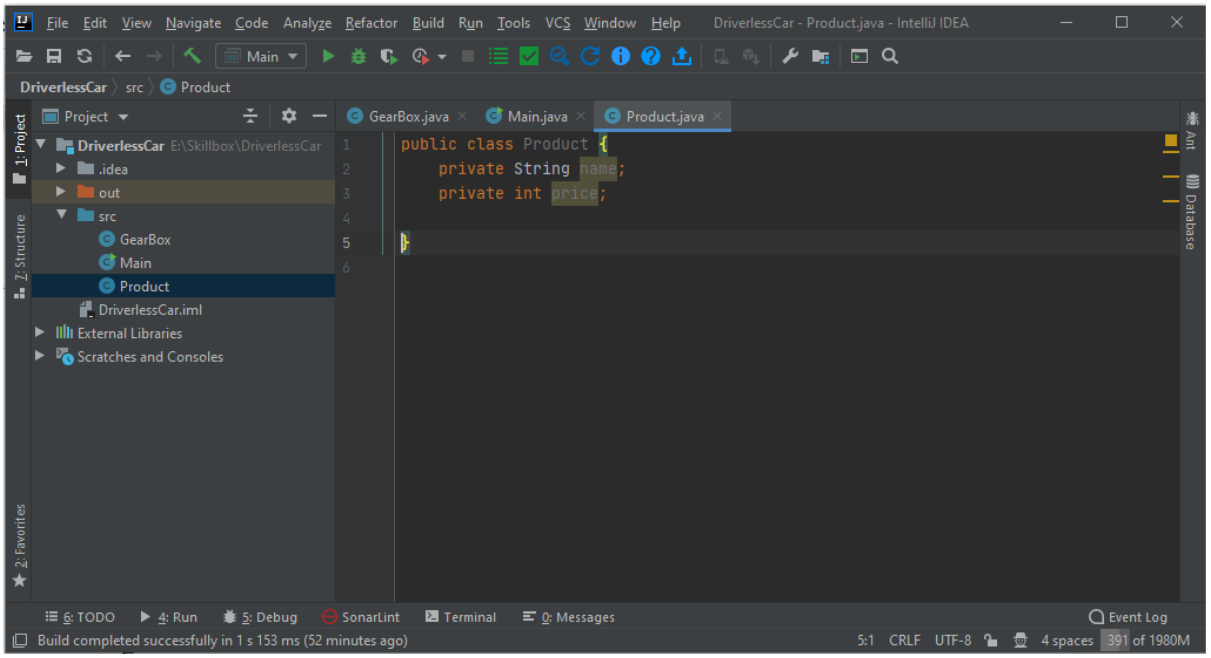
```
public Product(String name, int price) {
    this.name = name;
    this.price = price;
}
```

Итак, мы создали классический POJO-класс с двумя переменными, двумя геттерами и двумя сеттерами — для каждой из этих переменных, а также с конструктором с двумя параметрами. Кстати, в среде разработки IntelliJ Idea есть способ создавать конструкторы, геттеры и сеттеры для POJO-классов автоматически с помощью сочетания клавиш.

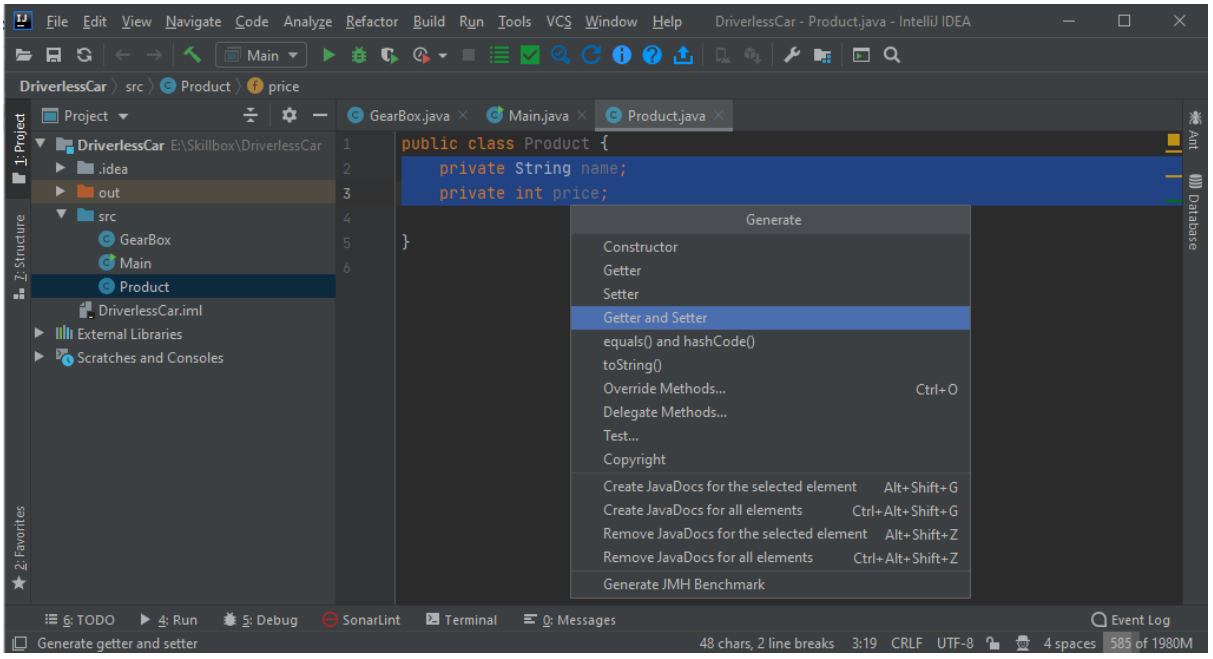
Windows/Linux: ***Alt*** **+** ***Insert***

Mac: ***Cmd*** **+** ***N*** ***(⌘N)***

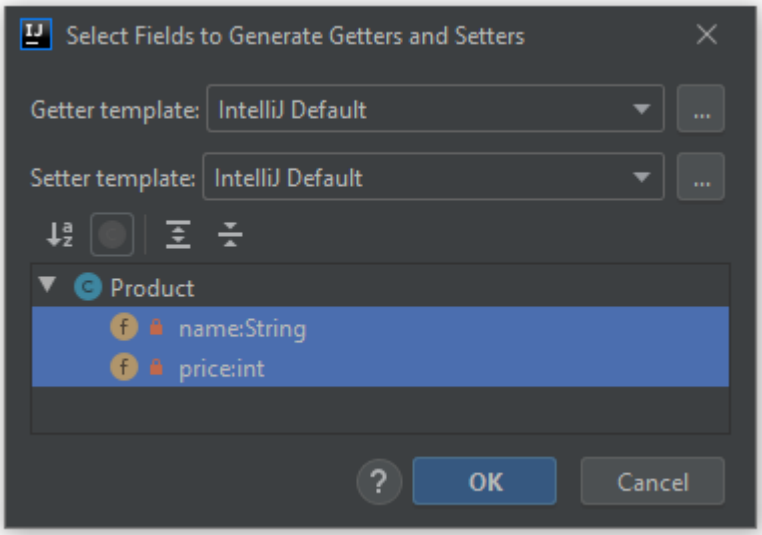
Стираем методы и конструктор.



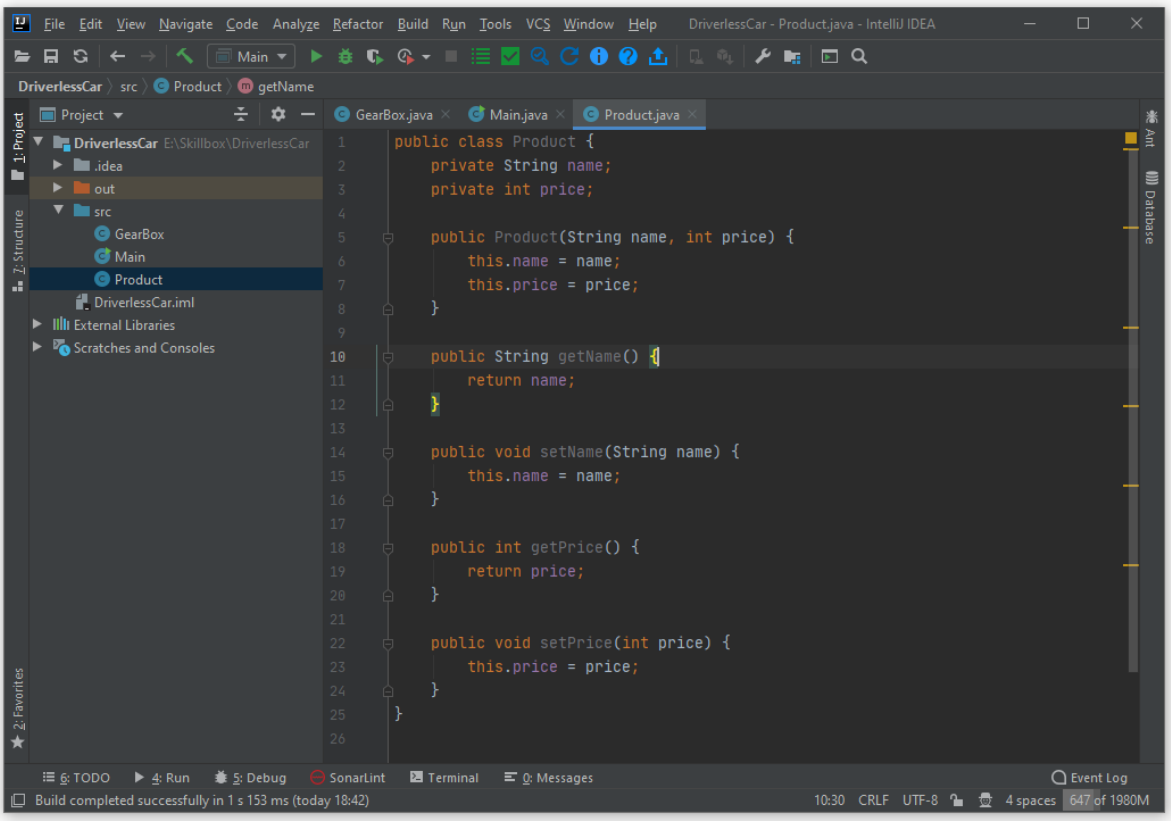
Выделяем переменные, нажимаем **Ctrl** + **N** (**Cmd** + **N**), выбираем Getter and Setter.



Выбираем обе переменные.



Нажимаем ОК — готово, появились геттеры и сеттеры для обеих переменных. Для конструктора можно сделать то же самое: **Ctrl** + **N** (**Cmd** + **N**), выбираем Getter and Setter, выбираем обе переменные, нажимаем ОК и получаем конструктор с двумя переменными.



Обратите внимание, что классы со сложной логикой тоже могут содержать либо просто сеттеры, либо просто геттеры, либо и сеттеры, и геттеры для своих переменных.

Геттеры и сеттеры принято именовать со слов get и set соответственно. Вы, конечно же, по-прежнему можете придумывать методам любые имена, но лучше придерживаться общепринятых правил написания понятного кода.

Итоги

05:12 — до конца

Итак, мы изучили понятия:

- POJO-класса — классического простого Java-объекта;
- геттера и сеттера — методов, которые возвращают или устанавливают значения переменных.

Далее мы поговорим о так называемой иммутабельности и immutable-классах.

Глоссарий

POJO-класс — класс, объекты которого выступают просто хранилищами каких-то данных.

Геттер — специальный метод, возвращающий значение свойства класса.

Сеттер — специальный метод, устанавливающий значение свойства класса.