Что такое Lombok?

Сегодня мы познакомимся с интересной библиотекой, которая помогает сократить объёмы необходимого, но тривиального кода.

Вы уже встречались с теми моментами, когда нужно описать модель данных состоящую только из полей, методов get(), set(), equals(), hashcode() и toString(). В итоге класс, имеющий всего три поля разрастается до нескольких десятков строк.

Возьмём, для примера, класс, описывающий птицу. Класс имеет три поля цвет птицы, её возраст и размах крыла:

public class Bird {

String color;

int age;

int wingspan;

public String getColor() {

return color;

}

public void setColor(String color) {

this.color = color;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public int getWingspan() {

return wingspan;

}

public void setWingspan(int wingspan) {

this.wingspan = wingspan;

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

Bird bird = (Bird) o;

return age == bird.age && wingspan == bird.wingspan && Objects.equals(color, bird.color);

}

@Override

public int hashCode() {

return Objects.hash(color, age, wingspan);

}

@Override

public String toString() {

return "Bird{" +

"color='" + color + '\'' +

", age=" + age +

", wingspan=" + wingspan +

'}'; }

Сократить код, сделать его более компактным, читаемым нам поможет библиотека Lombok.

Этот инструмент содержит в себе набор аннотаций, которые заменяют собой крупные куски типового кода. Вот так, например, будет выглядеть модель данных, описанная выше, с применением Lombok:

import lombok.Data;

@Data

public class Bird\_lombok {

String color;

int age;

int wingspan;

Давайте подключим Lombok к проекту.

Откроем pom.xml и добавим туда зависимость:

<dependencies>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.22</version>

<scope>provided</scope>

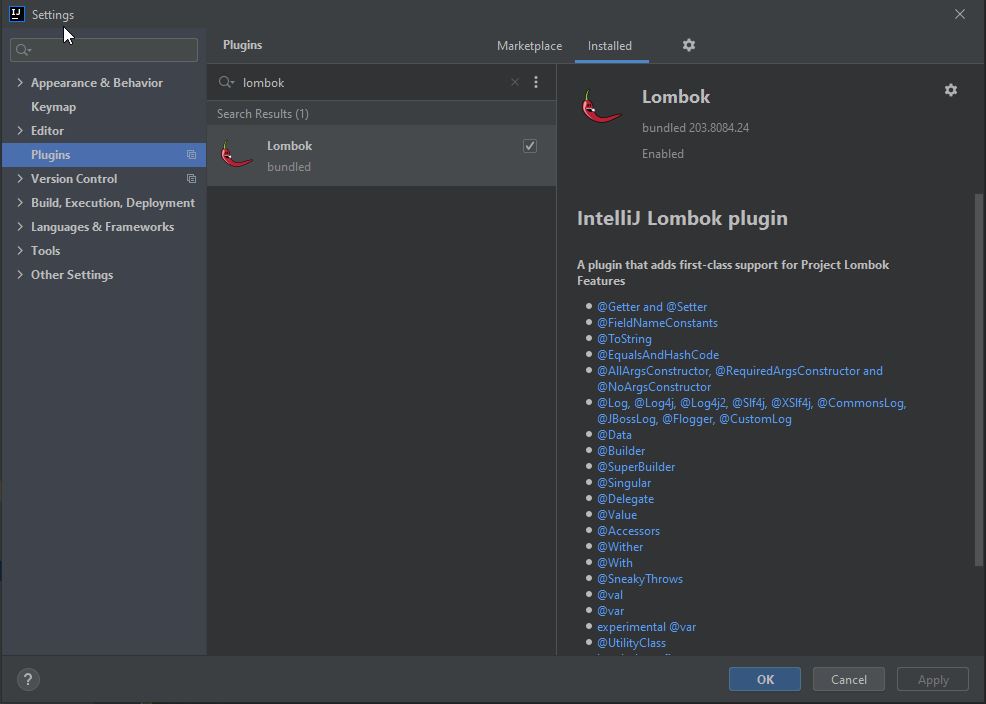
</dependency>

</dependencies>

Далее требуется подключить плагин Lombok к проекту.

File -> Settings -> plugins

В поле поиска вводим Lombok и подключаем его к проекту. В моём случае он уже подключен.



Теперь можно пользоваться аннотациями.

Виды аннотаций.

Давайте рассмотрим, какие аннотации есть в Lombok и что они делают.

@Getter/@Setter – эти две аннотации создают геттер и сеттер соответственно. Применяются к конкретному полю.

@Getter

@Setter

int age;

@ToString – генерирует метод toString()

@EqualsAndHashCode – создаёт методы equals() и @hashCode().

Пользоваться аннотациями @ToString и @EqualsAndHashCode следует с осторожностью. Причинам будет посвящён отдельный урок. Пока скажем, что у этих аннотаций есть возможность добавить параметры.

Например в классе Bird мы не хотим, чтобы методы equals() и hashCode() основывались на всех полях класса. С помощью параметра мы можем убрать проверку из метода equals() по какому то полю. И убрать это поле из участия в генерации хэш-кода в методе hashCode():

@EqualsAndHashCode(exclude = {"color", "age"})

Теперь сравнение объектов методом equals() будет производиться основываясь только на поле wingspan.

@Date - создаёт геттеры, сеттеры, конструктор, методы equals, hashCode, toString. А также делает все поля private.

@Builder – реализует шаблон Builder. С этой аннотацией мы можем создавать цепочки методов для создания экземпляра.

Bird\_lombok bl =Bird\_lombok.builder()

.age(10)

.color("red")

.wingspan(50)

.build();

Это краткое описание самых простых аннотаций. Подробнее о каждой из них Вы можете прочитать на официальном сайте разработчика:

<https://projectlombok.org/features/all>

Задание:

1. Подключить Lombok к проекту.
2. Создать модель данных House. Модель должна иметь 4 поля: высота здания, площадь, количество комнат и наличие гаража.
3. С помощью Lombok сгенерировать геттеры, сеттеры, методы equals(), hashCode() и toString().
4. Организовать сравнение экземпляров House только по полям: площадь, количество комнат, наличие гаража. Поле «высота здания» на результат вызова метода equalse() влиять не должно.
5. Напишите тесты.
6. Загрузите код в репозиторий. оставьте ссылку на коммит.
7. Переведите ответственного на Петра Арсентьева.