

Lombok

О библиотеке Lombok

Java — это язык с большим потенциалом, но для него может потребоваться значительный объём кода, особенно для простых задач.

Кроме того, в Java много повторяющегося кода, например, на каждое поле в классе необходимо писать геттеры и сеттеры, что может привести к значительному количеству избыточного и перегруженного кода.

Для борьбы с этим и повышения производительности и была создана такая библиотека, как **Lombok**.

Функции Lombok

Lombok — это библиотека Java, которая включает аннотации, помогающие избавиться от повторяющегося кода и шаблонов. Lombok работает на этапе сборки — с помощью аннотаций, которые пишет разработчик, он автоматически генерирует различные кусочки шаблонного кода, например, геттеры, сеттеры, конструкторы, методы equals и hashCode.

При создании любого класса эти методы неизбежно нужно написать, однако, как правило, они не содержат чего-то уникального. Lombok экономит время и позволяет разработчикам сосредоточиться на бизнес-логике своего проекта.

Установка Lombok

Чтобы установить Lombok, нужно добавить зависимость в pom.xml в своём проекте:

```
<dependency>  
    <groupId>org.projectlombok</groupId>  
    <artifactId>lombok</artifactId>  
    <version>1.18.20</version>  
    <scope>provided</scope>  
</dependency>
```

После этого необходимо перезагрузить проект.
Также желательно установить плагин для Lombok в IDEA.

Аннотации Lombok. **@Getter, @Setter**

Если поле аннотировано **@Getter** и/или **@Setter**, Lombok автоматически сгенерирует заданный по геттер и/или сеттер соответственно. Если аннотации поставлены над полем, то геттер и/или сеттер будет сгенерирован для этого поля. Сгенерированные методы будут публичными (public), если не указан конкретный уровень доступа (AccessLevel). Допустимые значения.

AccessLevel — это PUBLIC, PROTECTED, PACKAGE и PRIVATE. Обратите внимание: можно аннотировать и целый класс. В таком случае данная логика будет применяться к каждому из его полей.

Аннотации Lombok.

@NoArgsConstructor,
@RequiredArgsConstructor,
@AllArgsConstructor

При наличии аннотации **@NoArgsConstructor**, Lombok сгенерирует конструктор без параметров.

Аналогично, когда класс аннотирован **@AllArgsConstructor**, будет сгенерирован конструктор с одним параметром для каждого поля вашего класса.

@RequiredArgsConstructor включает в конструктор все поля, помеченные `final`, и все поля, помеченные **@NonNull**.

Также нужно обратить внимание, что все эти аннотации игнорируют статические поля.

Аннотации Lombok.

@ToString

Если класс помечен аннотацией **@ToString**, Lombok сгенерирует нужную реализацию метода `toString()`. По умолчанию будет возвращаться строка, в которой содержится имя класса, за которой следуют значения всех полей, разделенные запятыми.

Если установить параметр **includeFieldNames** в значение `true`, то имя каждого поля будет ставиться перед его значением. По умолчанию при генерировании метода `toString()` будут учитываться все нестатические поля.

Если вы хотите, чтобы Lombok проигнорировал поле, нужно аннотировать его **@ToString.Exclude**. Также можно явно указать, какие поля будут включены в метод `toString`. Для этого установите **@ToString(onlyExplicitlyIncluded = true)**. Затем при помощи **@ToString.Include** пометьте каждое поле, которое хотите включить.

Аннотации Lombok. **@EqualsAndHashCode**

Если аннотировать класс **@EqualsAndHashCode**, то Lombok автоматически реализует за вас методы `equals()` и `hashCode()`. По умолчанию будут учитываться все нестатические поля. Также можно указать, какие поля будут использоваться, поставив над ними аннотацию `@EqualsAndHashCode.Include` или `@EqualsAndHashCode.Exclude`.

Обратите внимание: Lombok сгенерирует методы `equals()` и `hashCode()`, не нарушая контракта между ними.

Аннотации Lombok. **@NonNull**

Аннотация **@NonNull** применяется для параметров метода или конструктора, В таком случае Lombok сгенерирует код для проверки на null:

```
if (value == null) {  
    throw new NullPointerException("value is marked non-null but is null");  
}
```

Аннотации Lombok.

@Data

@Data — это сокращенная аннотация, сочетающая возможности `@ToString`, `@EqualsAndHashCode`, `@Getter`, `@Setter` и `@RequiredArgsConstructor`.

`@Data` генерирует весь шаблонный код, вовлечённый в работу с объектами POJO.

Другими словами, будут сгенерированы геттеры для всех полей, сеттеры для всех нефинальных полей, правильные реализации `toString`, `equals` и `hashCode`, включающие все поля класса, а также конструктор для всех финальных полей.

Аннотации Lombok. **@Value**

@Value — это неизменяемый вариант **@Data**. С этой аннотацией Lombok по умолчанию делает все поля `private` и `final`. Кроме того, сеттеры не генерируются, а класс как таковой помечается `final`.

Таким образом, от этого класса нельзя наследоваться. Так же, как и в случае с **@Data**, создаются реализации `toString()`, `equals()` и `hashCode()`.

Выводы

В этом материале вы познакомились с библиотекой Lombok и разобрали основные аннотации, которые в ней применяются.

С помощью Lombok вы можете избавить себя от написания рутинного кода и освободить время для решения интересных задач.