

# Code generation

Профессия Java-разработчик на Hexlet

Преподаватель: Яковлев Егор

# Вопросы к лекции:

1. Как бороться с бойлерплейт-кодом в Java?
2. Что такое Lombok?
3. Что умеет делать Lombok?

# План

1. Бойлерплейт-код
2. Lombok
3. Основные аннотации Lombok
4. Недостатки Lombok

# Бойлерплейт-код в Java

```
public class User {  
    private Integer id;  
    private String name;  
  
    // boilerplate getters  
    public String getName() {  
        return this.name;  
    }  
  
    // boilerplate setters  
    public void setName(String name) {  
        this.name = name;  
    }  
}
```

# Lombok

```
@Data
public class User {
    private Integer id;
    private String name;

    // getters, setters, equals, hashCode, toString
}
```

демо

# Добавление в проект

```
<dependencies>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.16.18</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

# It is a kind of magic

- Lombok генерирует код на этапе компиляции (см. .class)
- При использовании Lombok мы пишем не валидный код на Java – требуется плагин для IDE

[Шпаргалка по Lombok](#)

# Основные аннотации Lombok

Аннотация	Функционал
@Data	toString, equals, hashCode, getters, setters, no args constructor
@Getter	getter для свойства
@Setter	setter для свойства
@Getter/@Setter над классом	getter / setter для всех свойств



# Основные аннотации Lombok

Аннотация	Функционал
@NoArgsConstructor/@NoArgsConstructor	Конструктор без аргументов / Конструктор со всеми аргументами
@Builder	Реализация паттерна builder
@NotNull	Аргумент не null

# Недостатки Lombok

- "Невидимый" код
- Невалидный Java-код
- Лишняя зависимость