**lombok学习文档**

# 相关网站

官网：<https://projectlombok.org>

入门：<https://www.cnblogs.com/moonlightL/p/8311664.html>

[eclipse安装lombok](https://www.cnblogs.com/30go/p/8468981.html): <https://www.cnblogs.com/30go/p/8468981.html>

https://www.cnblogs.com/heyonggang/p/8638374.html

# 前言

Lombok 是一个 Java 库，它作为插件安装至编辑器中，其作用是通过简单注解来精简代码，以此达到消除冗长代码的目的。

# 简单介绍

## 优点

1) 通过注解自动生成成员变量的 getter、setter 等方法，使代码简洁

2) 修改类变量名时，无需关注其 getter、setter 等方法

## 缺点

降低源码文件的可读性。

## 原理

从 Java 6 开始，javac 就支持 JSR 269 API 规范，而 Lombok 实现 JSR 269 Pluggable Annation Processing API 规范。

当我们编写代码并保存后，编辑器会自动编译源码文件，在这个过程中，源码先被转化为 AST。

然后，Lombok 插件解析 AST 是否存在 Lombok 的注解。如果存在则修改 AST ，使其生成注解对应的代码。

最终将修改的 AST 解析并生成字节码文件。

# 安装插件

为编辑器安装 Lombok 插件。

## Eclipse 安装

点击 [Lombok.jar](https://projectlombok.org/download)，下载该 jar 包。

将lombok.jar包拷贝到eclipse的安装目录下，并且在eclipse.ini文件中添加如下两行。

-Xbootclasspath/a:lombok.jar --备注：如果没有这一行也是没问题的

-javaagent:lombok.jar

重启eclipse即可

# 使用

使用 Lombok 的方法非常简单，就是在类上或者成员变量上添加注解即可。

为了能使用注解，我们还需要在项目中引入 lombok 的 jar 包。

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.6</version>

</dependency>

## 注解介绍

Lombok 常用的注解如下：

| **注解名** | **作用描述** |
| --- | --- |
| @Getter | 作用在类上或成员变量上，生成对应的 getter 方法 |
| @Setter | 作用在类上或成员变量上，生成对应的 setter 方法 |
| @NoArgsConstructor | 作用在类上，生成对应的无参构造方法 |
| @AllArgsConstructor | 作用在类上，生成对应的有参构造方法 |
| @ToString | 作用在类上，生成对应的 toString 方法 |
| @EqualsAndHashCode | 作用在类上，生成对应的 equals 和 hashCode 方法 |
| @Data | 作用在类上，效果等同于上述 5 个注解，排除 @AllArgsConstructor 功能 |
| @Log4j/@Slf4j | 作用在类上，生成对应的 Logger 对象，变量名为 log |
| @Builder | 作用在类上，对外保持private setter，而对属性的赋值采用Builder的方式 |

## 案例演示

本次测试使用 Ecplise 编辑器。

**import** java.util.Date;

**import** lombok.Data;

@Data

**public** **class** User {

**private** **int** id;

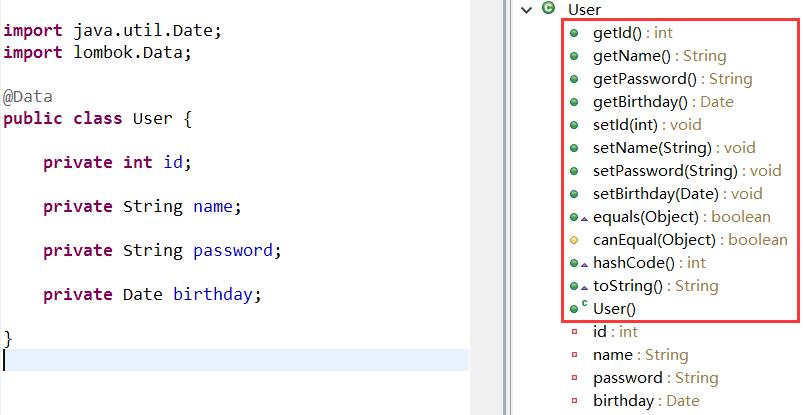
**private** String name;

**private** String password;

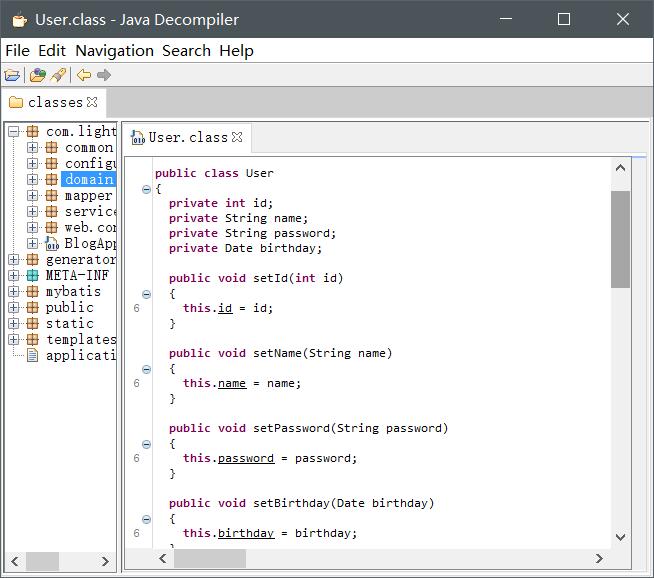
**private** Date birthday;

}

当添加注解保存文件后，Ecplise 编辑器的 Outline 视图结果如下：



我们还可以使用 jd-gui 等反编译工具查看源码，结果如下：



# 其他

## @EqualsAndHashCode注解问题

<https://blog.csdn.net/linlinxie/article/details/79509236>

当使用@Data注解的实体有父类（非Object）时，eclipse会有警告提示

大概意思时自动生成的equals和hashCode方法没有调用父类，如果是故意不掉用父类的话，可以加上@EqualsAndHashCode(callSuper=false)注解。

因为这个是实体，继承父类主要也是需要使用父类的元素，所以生成的equals和hashCode方法应该包含父类在内，因此可以把callSuper设为true。

通过反编译来查看callSuper为true和false的区别。

当callSuper为true时，在equals方法中会多出一个判断，如下

if (!super.equals(o)) {

return false;

}

1

2

3

可以查看官网的解释https://projectlombok.org/features/EqualsAndHashCode,大概意思是你可以通过使用@EqualsAndHashCode注解声明去除哪些父类的域到equals方法中，如果你的父类也有@Data标签，则可以放心地callSuper，但是当你的类没有显式继承其他类的时候，使用该注解会有编译异常。