  Lomboz是一个基于LGPL的开源J2EE综合开发环境的Eclipse[插件](http://baike.baidu.com/view/18979.htm)，对编码，发布，测试，以及debug等各个软件开发的生命周期提供支持，支持JSP,EJB等。Lomboz是Eclipse的一个J2EE的[插件](http://baike.baidu.com/view/18979.htm)，它将很多Java[应用服务器](http://baike.baidu.com/view/995472.htm)、J2EE组件和Web应用开发集成到Eclipse中，可以帮助Java开发者使用Eclipse建立、测试、部署J2EE应用。

        在<http://www.blogjava.net/fancydeepin/archive/2012/07/12/lombok.html>发现讲解Lomboz的安装入门文章，观看原文的可以点击上面的链接，也可以在此处继续阅读，内容一致。

前言：  
    逛开源社区的时候无意发现的，用了一段时间，觉得还可以，特此推荐一下。  
**lombok** 提供了简单的注解的形式来帮助我们简化消除一些必须有但显得很臃肿的 java 代码。特别是相对于 POJO，光说不做不是我的风格，先来看看吧。  
  
lombok 的官方网址：<http://projectlombok.org/>    
  
lombok 其实到这里我就介绍完了，开个玩笑，其实官网上有 lombok 三分四十九秒的视频讲解，里面讲的也很清楚了，而且还有文档可以参考。  
在这里我就不扯太多，先来看一下 **lombok 的安装**，其实这个官网视频上也有讲到啦  
  
lombok 安装  
    使用 lombok 是需要安装的，如果不安装，IDE 则无法解析 lombok 注解。先在官网下载最新版本的 JAR 包，现在是 0.11.2 版本，我用的是 0.11.0  
    第一次使用的时候我下载的是最新版本的，也就是我现在用的 0.11.0，到现在已经更新了两个版本，更新的好快啊 ... ...  
  
1. 双击下载下来的 JAR 包安装 lombok  
    我选择这种方式安装的时候提示没有发现任何 IDE，所以我没安装成功，我是手动安装的。如果你想以这种方式安装，请参考官网的视频。  
  
2.eclipse / myeclipse 手动安装 lombok  
    1. 将 lombok.jar 复制到 myeclipse.ini / eclipse.ini 所在的文件夹目录下  
    2. 打开 eclipse.ini / myeclipse.ini，在最后面插入以下两行并保存：  
        -Xbootclasspath/a:lombok.jar  
        -javaagent:lombok.jar  
    3.重启 eclipse / myeclipse  
  
lombok 注解：  
    lombok 提供的注解不多，可以参考官方视频的讲解和官方文档。  
    Lombok 注解在线帮助文档：[http://projectlombok.org/features/index.](http://projectlombok.org/features/index.html)    下面介绍几个我常用的 lombok 注解：  
        @Data   ：注解在类上；提供类所有属性的 getting 和 setting 方法，此外还提供了equals、canEqual、hashCode、toString 方法  
        @Setter：注解在属性上；为属性提供 setting 方法  
        @Getter：注解在属性上；为属性提供 getting 方法  
        @Log4j ：注解在类上；为类提供一个 属性名为log 的 log4j 日志对象  
        @NoArgsConstructor：注解在类上；为类提供一个无参的构造方法  
        @AllArgsConstructor：注解在类上；为类提供一个全参的构造方法  
  
下面是简单示例  
    1.不使用 lombok 的方案

1http://www.blogjava.net/Images/OutliningIndicators/None.gif  
 2http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.blogjava.net/Images/OutliningIndicators/ContractedBlock.gifpublic class Person {  
 3http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif  
 4http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif    private String id;  
 5http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif    private String name;  
 6http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif    private String identity;  
 7http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif    private Logger log = Logger.getLogger(Person.class);  
 8http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif      
 9http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.blogjava.net/Images/OutliningIndicators/ContractedSubBlock.gif    public Person() {  
10http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif          
11http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
12http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif      
13http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.blogjava.net/Images/OutliningIndicators/ContractedSubBlock.gif    public Person(String id, String name, String identity) {  
14http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        this.id              = id;  
15http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        this.name       = name;  
16http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        this.identity  = identity;  
17http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
18http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif      
19http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.blogjava.net/Images/OutliningIndicators/ContractedSubBlock.gif    public String getId() {  
20http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        return id;  
21http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
22http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif      
23http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.blogjava.net/Images/OutliningIndicators/ContractedSubBlock.gif    public String getName() {  
24http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        return name;  
25http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
26http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif      
27http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.blogjava.net/Images/OutliningIndicators/ContractedSubBlock.gif    public String getIdentity() {  
28http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        return identity;  
29http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
30http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif      
31http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.blogjava.net/Images/OutliningIndicators/ContractedSubBlock.gif    public void setId(String id) {  
32http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        this.id = id;  
33http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
34http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif      
35http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.blogjava.net/Images/OutliningIndicators/ContractedSubBlock.gif    public void setName(String name) {  
36http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        this.name = name;  
37http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
38http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif      
39http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.blogjava.net/Images/OutliningIndicators/ContractedSubBlock.gif    public void setIdentity(String identity) {  
40http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif        this.identity = identity;  
41http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
42http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
43http://www.blogjava.net/Images/OutliningIndicators/None.gif

2.使用 lombok 的方案  
1http://www.blogjava.net/Images/OutliningIndicators/None.gif  
 2http://www.blogjava.net/Images/OutliningIndicators/None.gif@Data  
 3http://www.blogjava.net/Images/OutliningIndicators/None.gif@Log4j  
 4http://www.blogjava.net/Images/OutliningIndicators/None.gif@NoArgsConstructor  
 5http://www.blogjava.net/Images/OutliningIndicators/None.gif@AllArgsConstructor  
 6http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.blogjava.net/Images/OutliningIndicators/ContractedBlock.gifpublic class Person {  
 7http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif  
 8http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif    private String id;  
 9http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif    private String name;  
10http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif  private String identity;  
11http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif      
12http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
13http://www.blogjava.net/Images/OutliningIndicators/None.gif

        上面的两个 java 类，从作用上来看，它们的效果是一样的，相比较之下，很明显，使用 lombok 要简洁许多，特别是在类的属性较多的情况下，  
同时也避免了修改字段名字时候忘记修改方法名所犯的低级错误。最后需要注意的是，在使用 lombok 注解的时候记得要导入 lombok.jar 包到工程