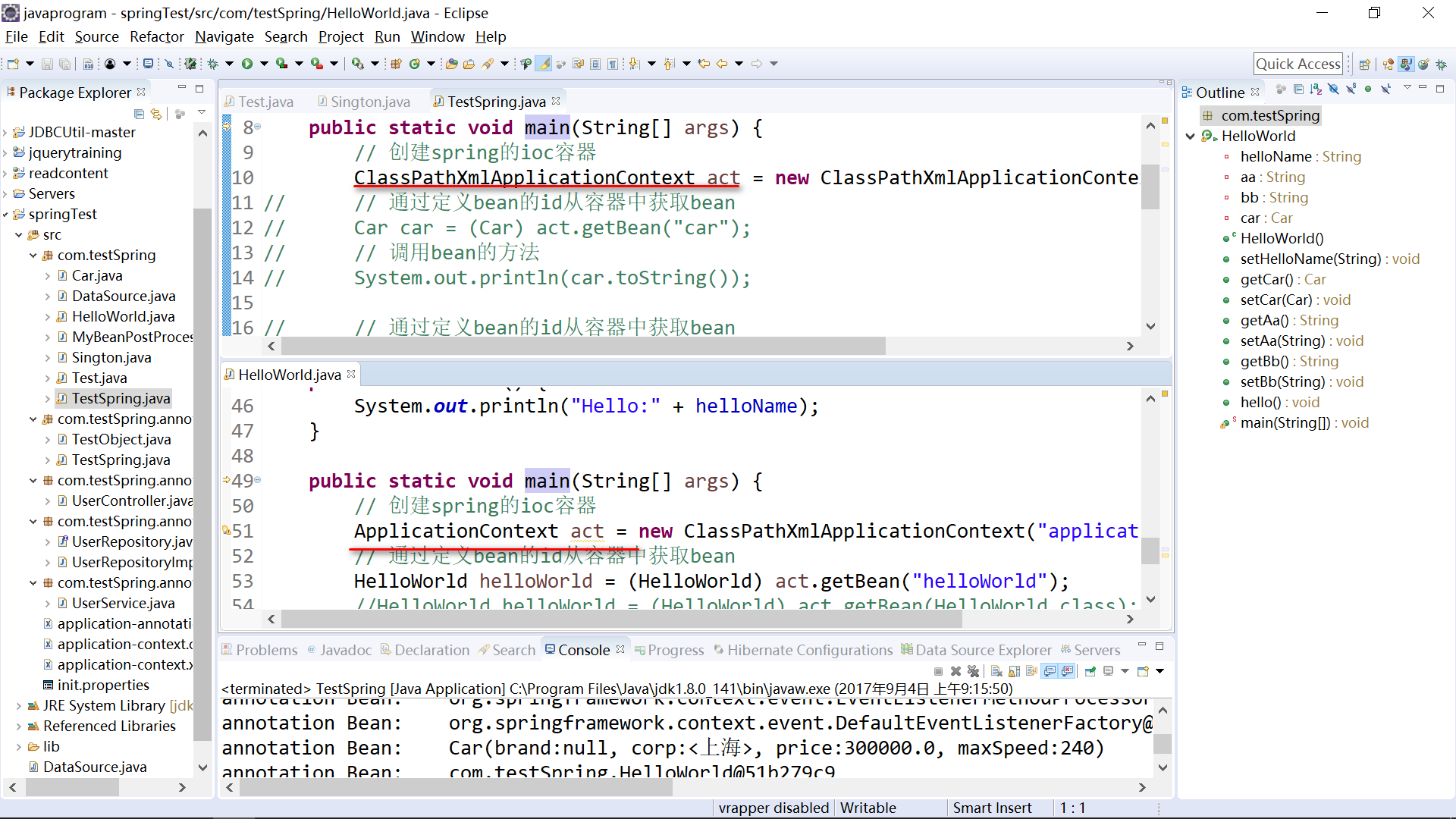
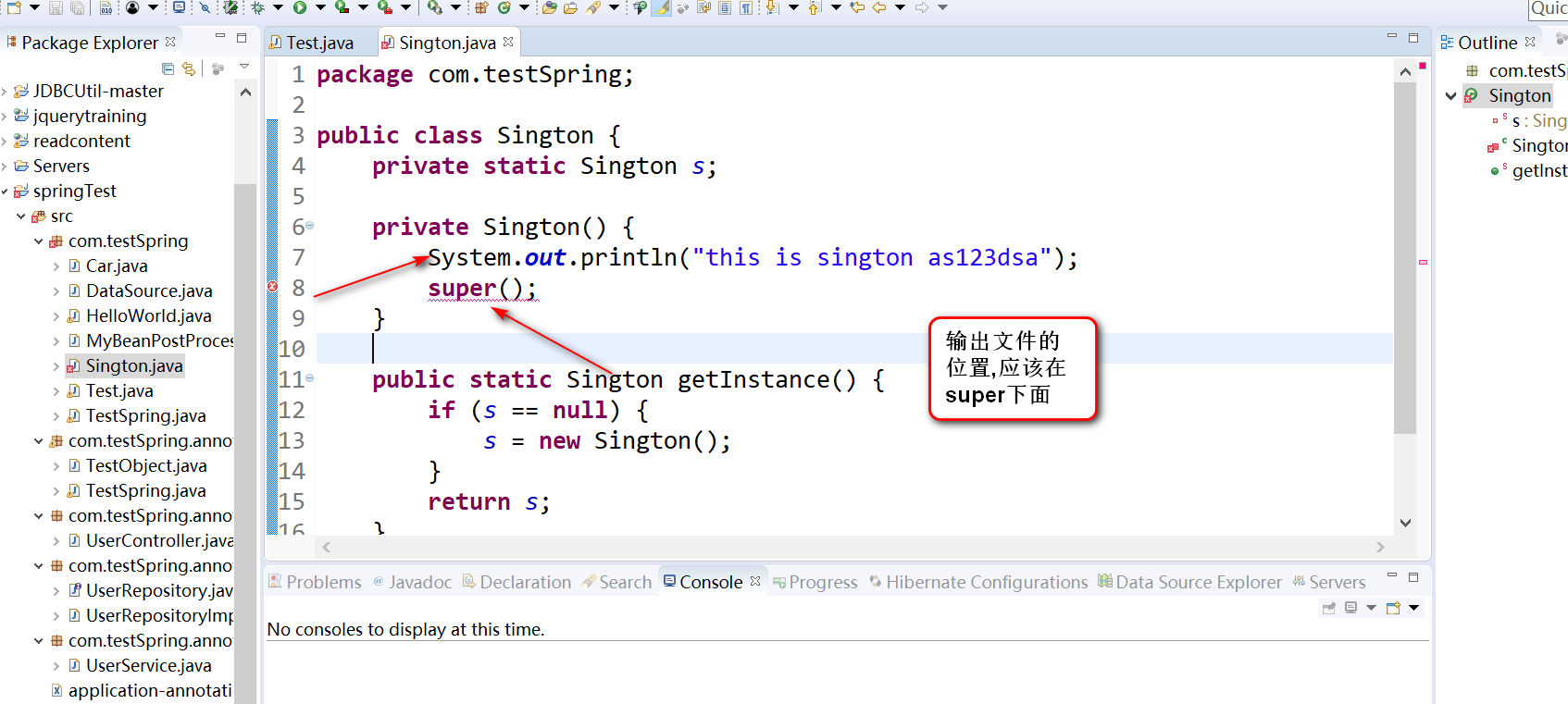
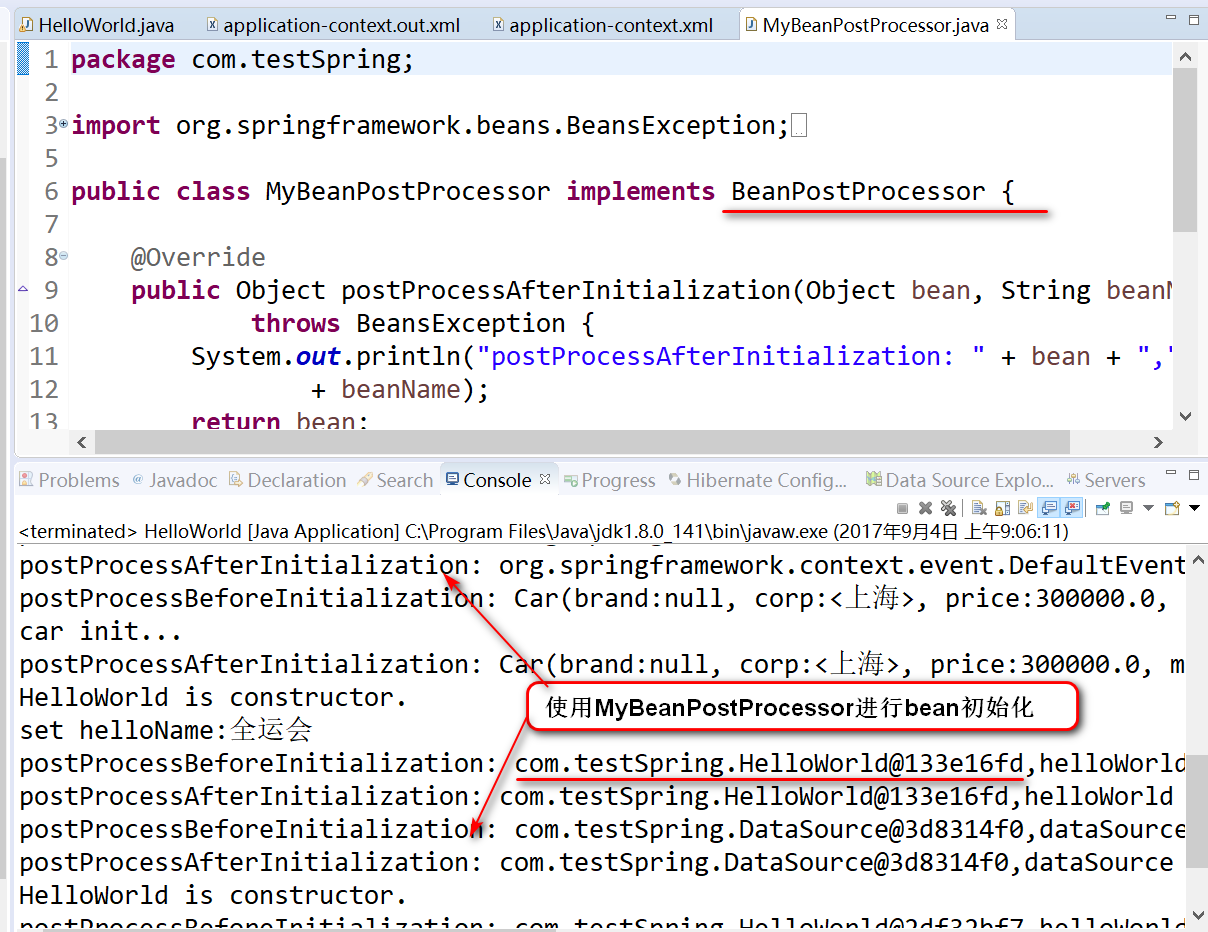
## 初始化 ClassPathXmlApplicationContext对象

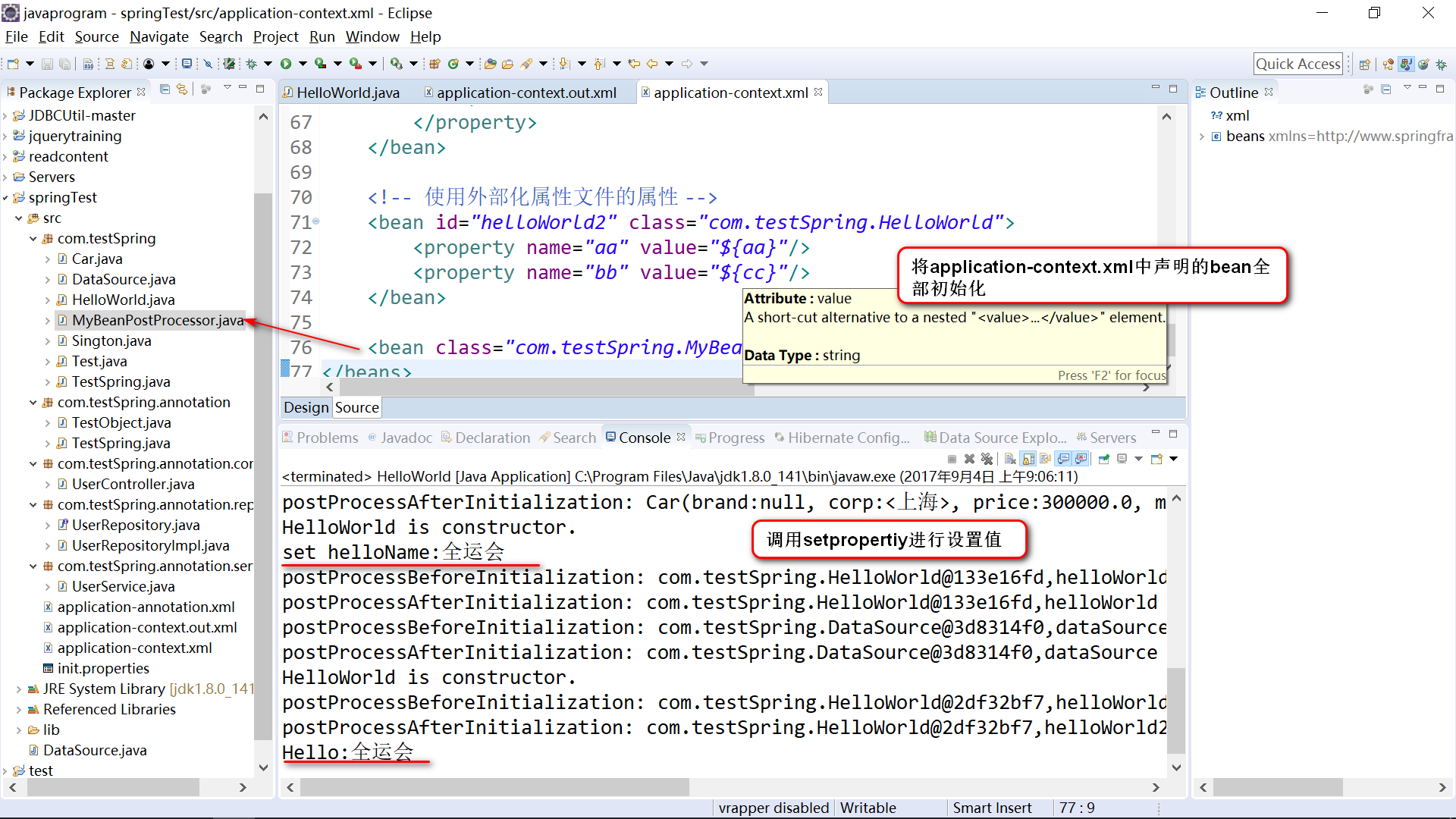




注意 super方面前面不能有输出语句,输出语句应该放在super下方.

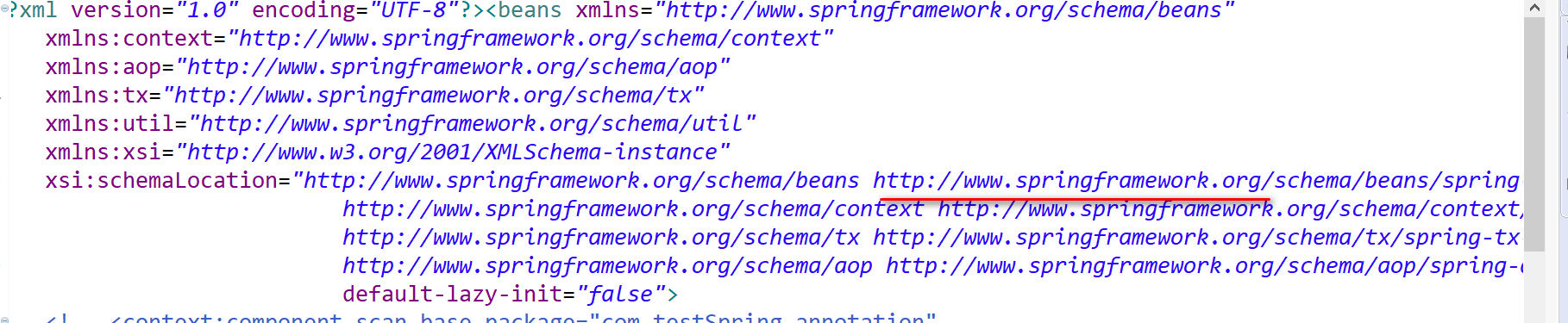
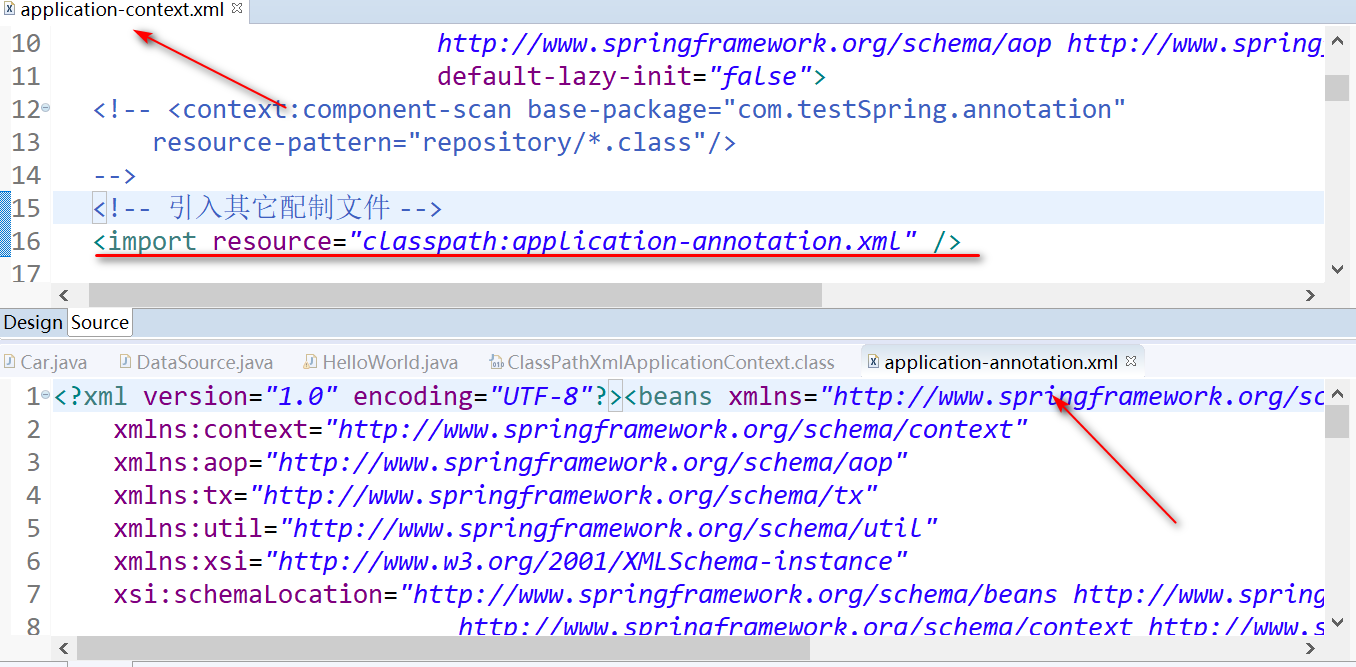
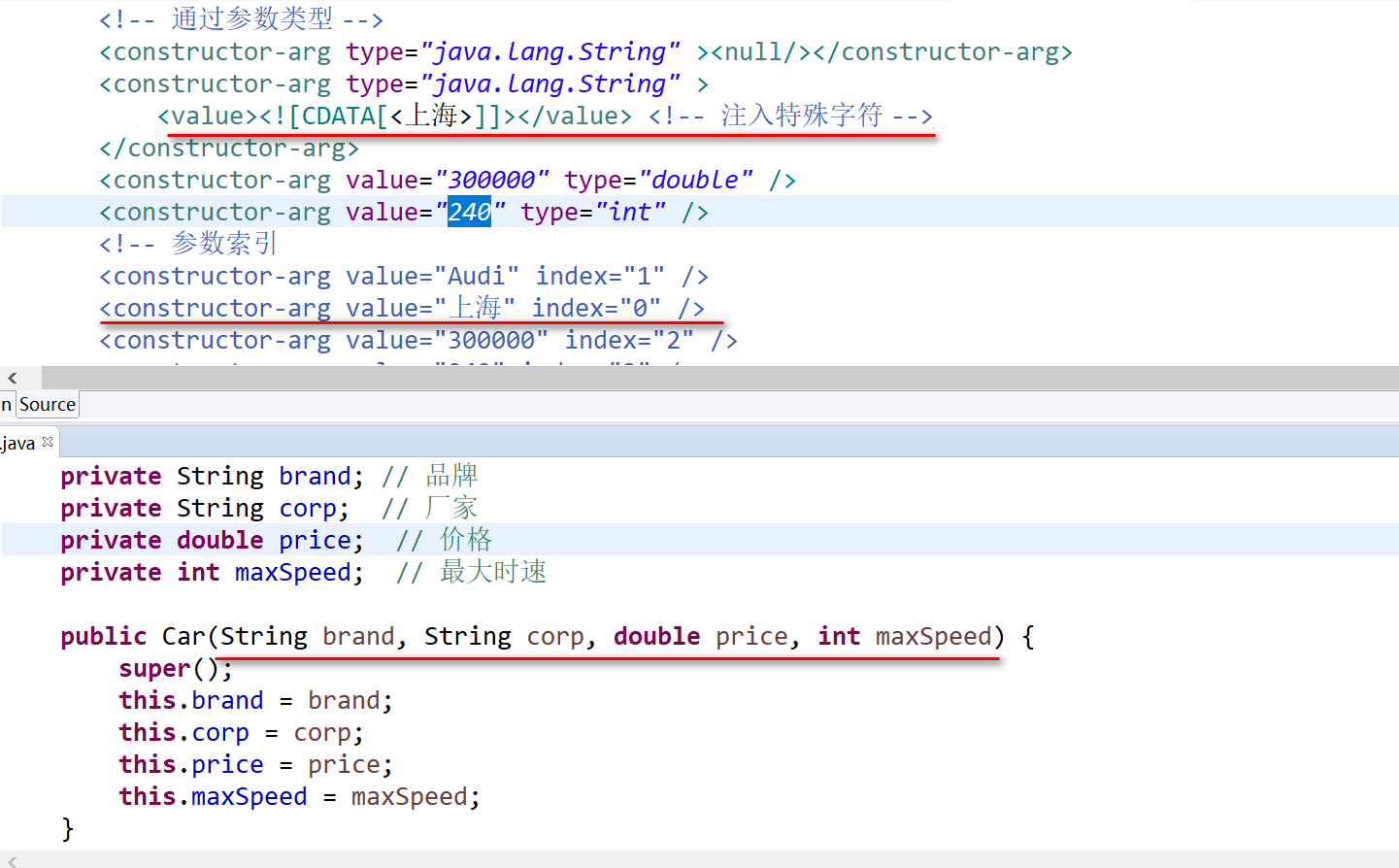
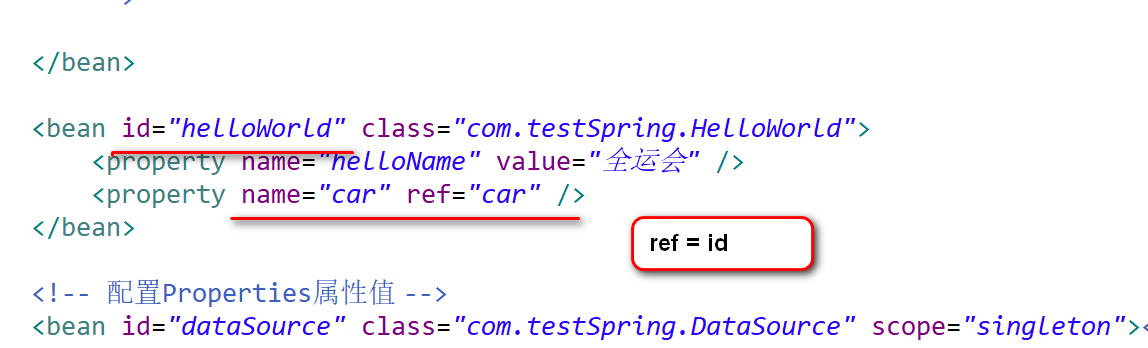
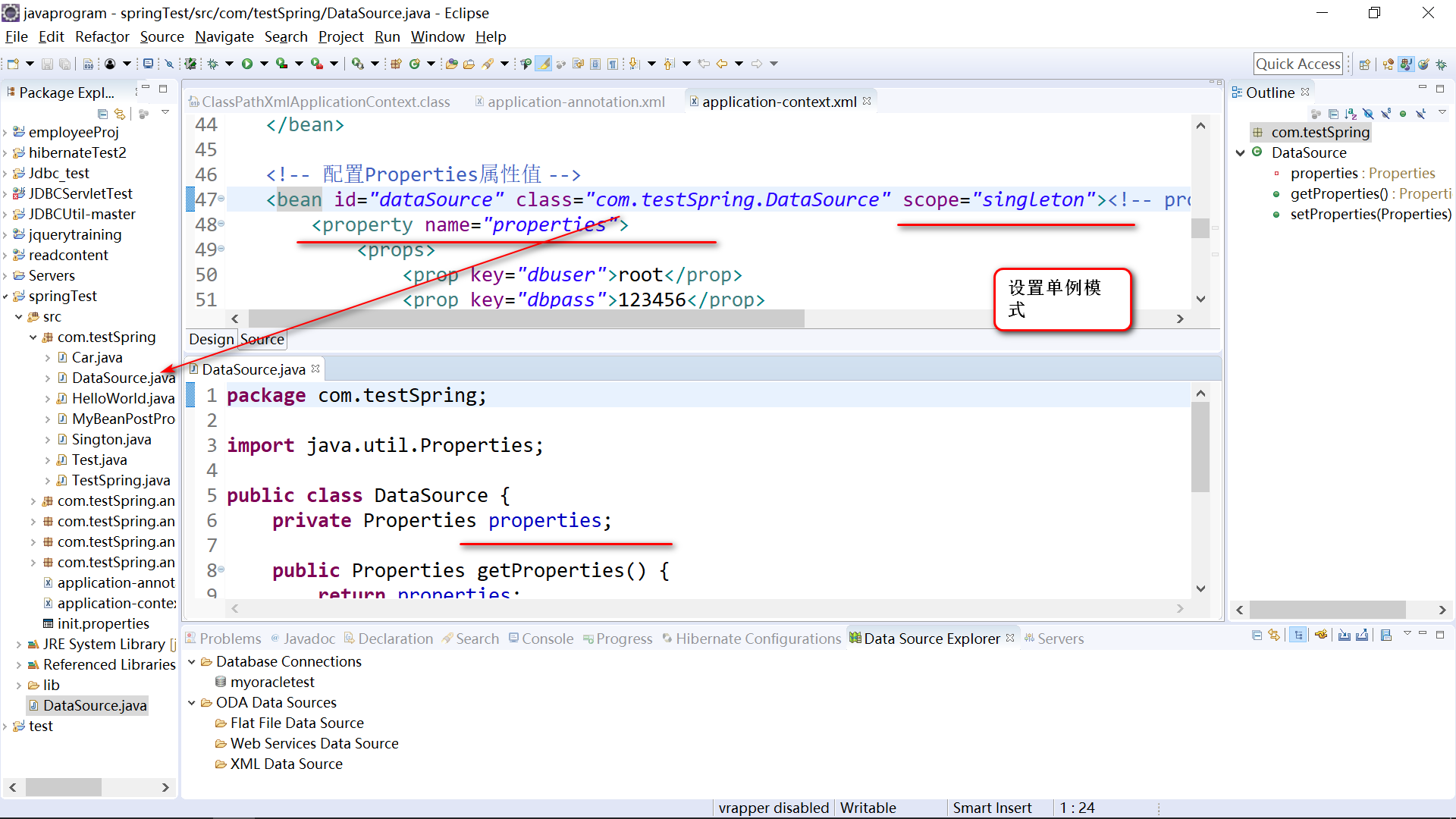
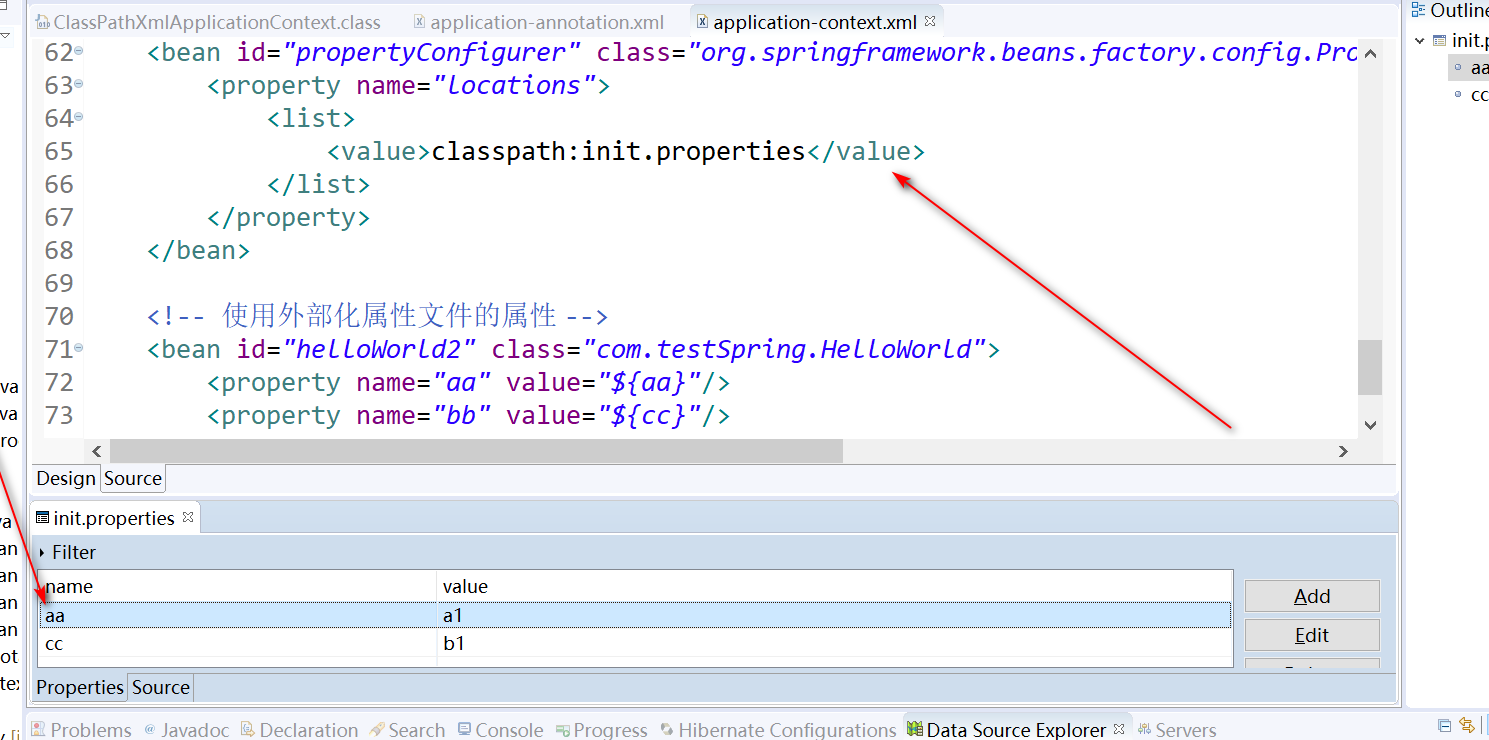
## Processor对象ClassPathXmlApplicationContext





说明

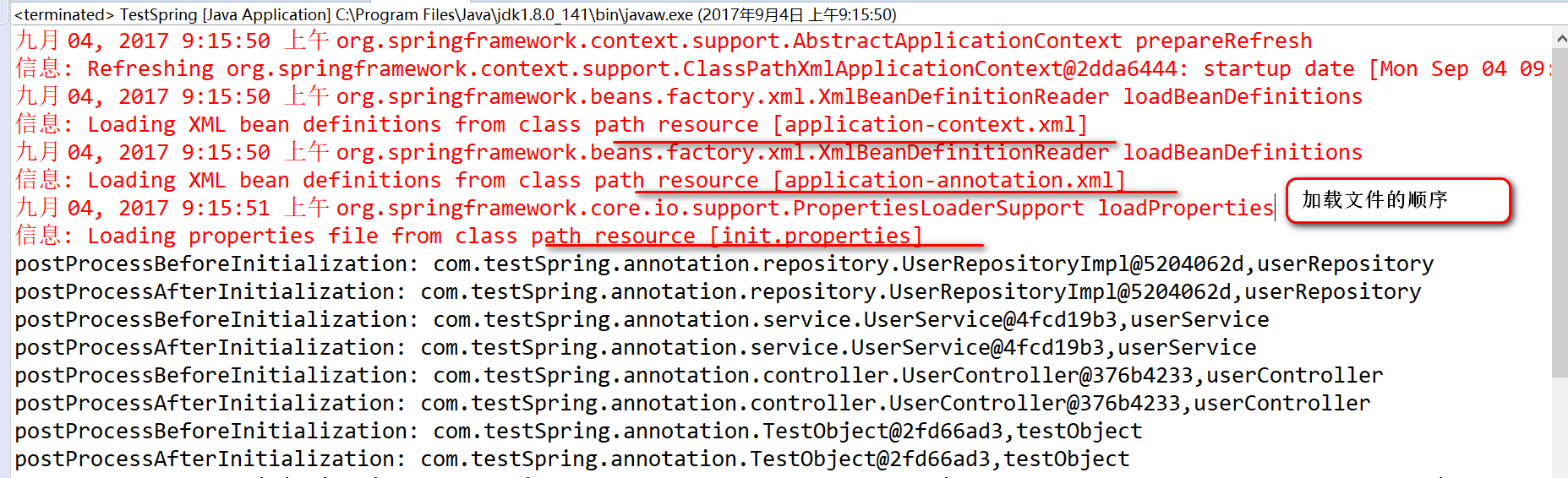
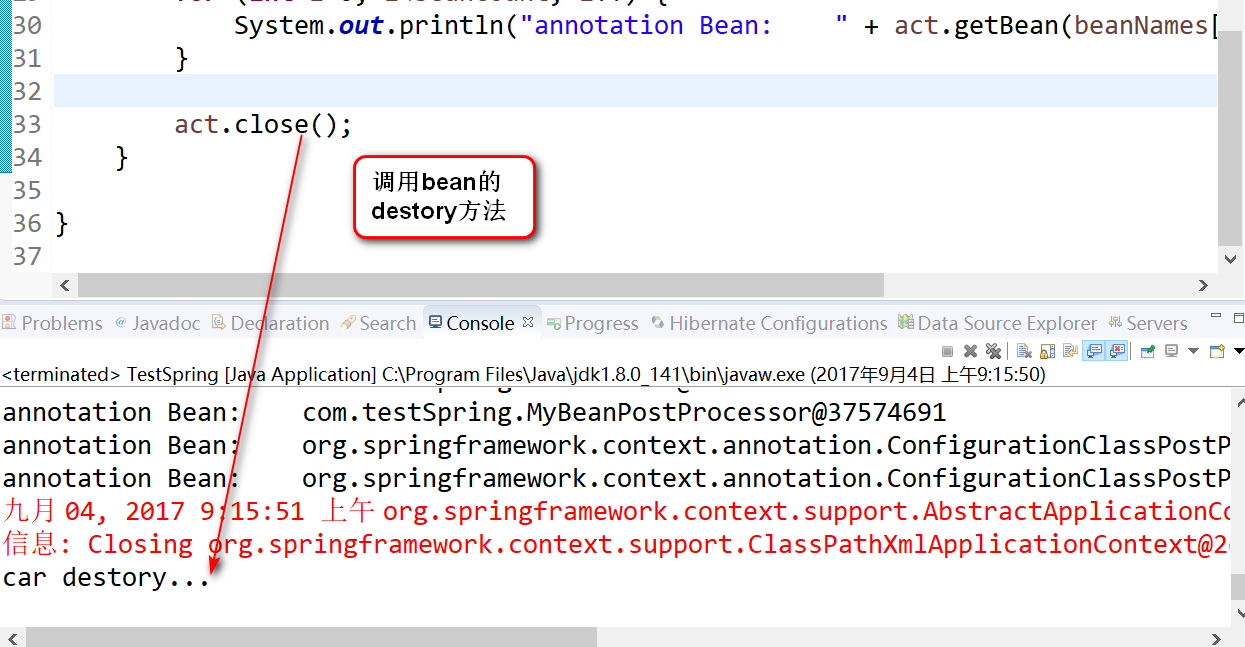
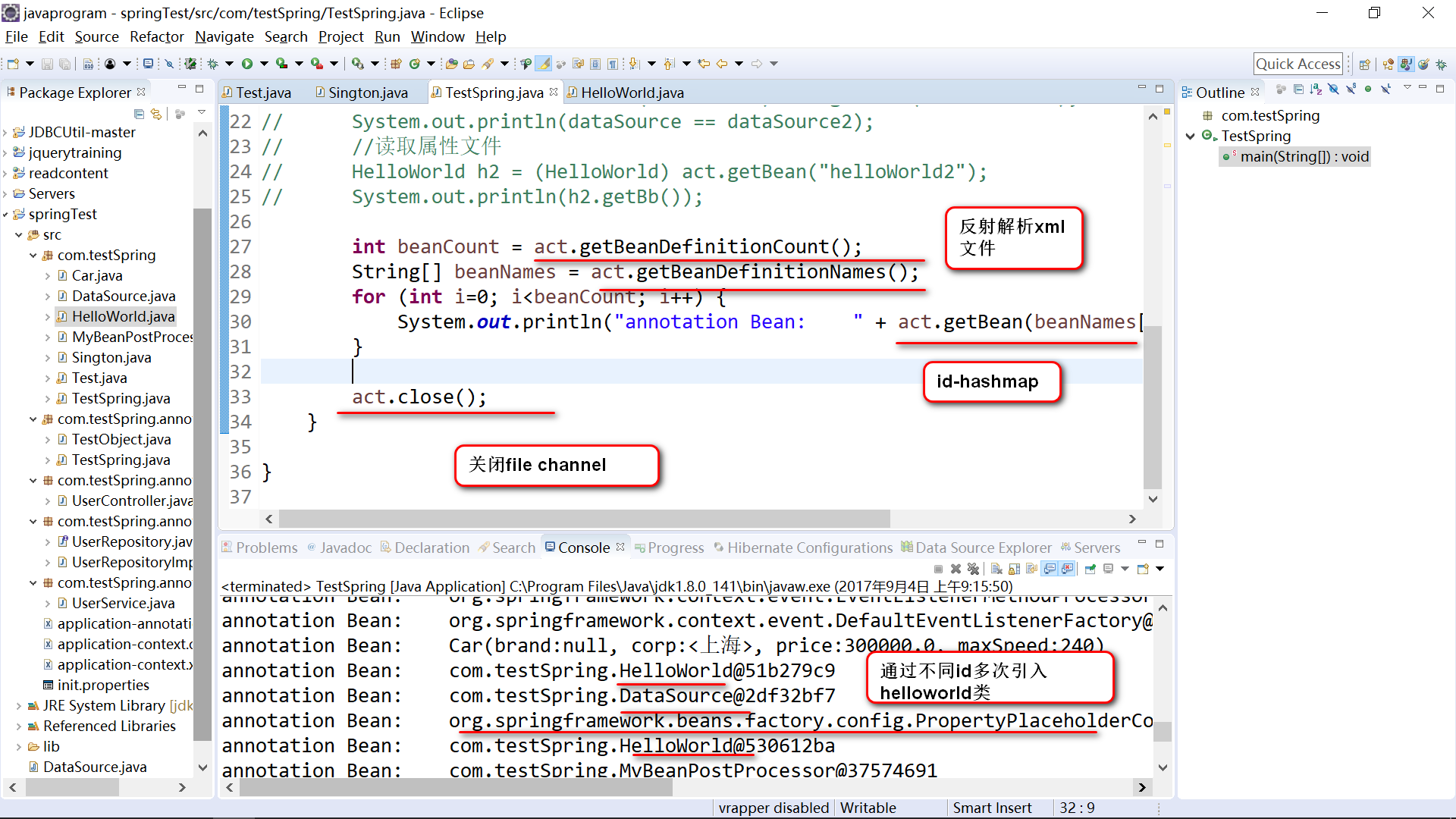
初始化的时候先读取xml文件,寻找相应的processor对象,使用这个对象初始化,其他的bean文件,赋值的时候调用set方法



说明

1. bean中使用 ref 设置引用对象的值,使用id作为key,如果没有设置key,那么使用完整的包名.
2. 使用索引或者类型的方式进行构造方法注入,但是注意的是使用类型应该与构造方法的顺序相同,
3. 构造方法注入和属性注入的关键字差别为 构造方法 constructor,属性注入property
4. Spring配置文件中可以引入 propertyconfiguer进行二次松耦合

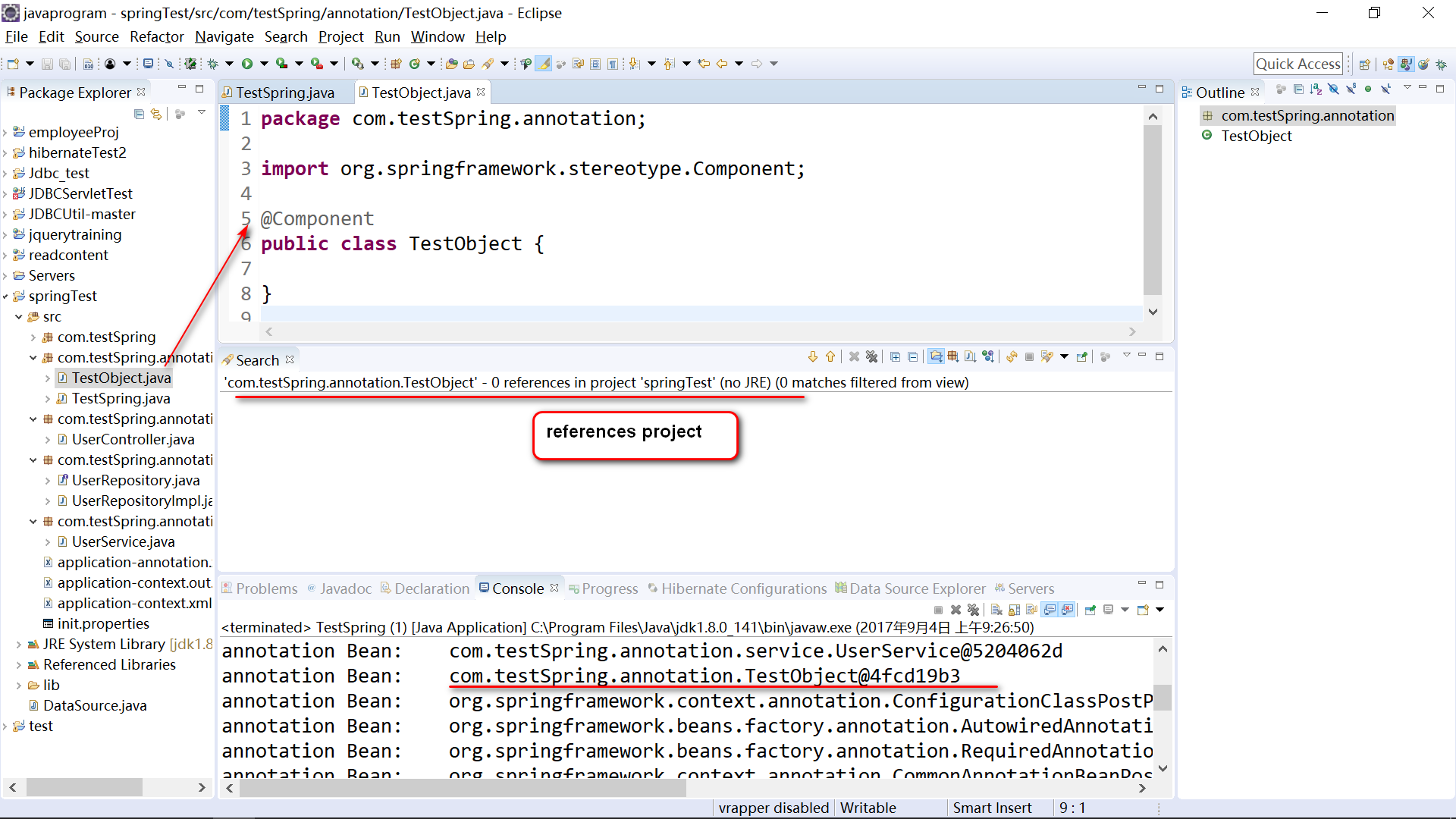
## 使用反射获取ApplicationContext的元数据

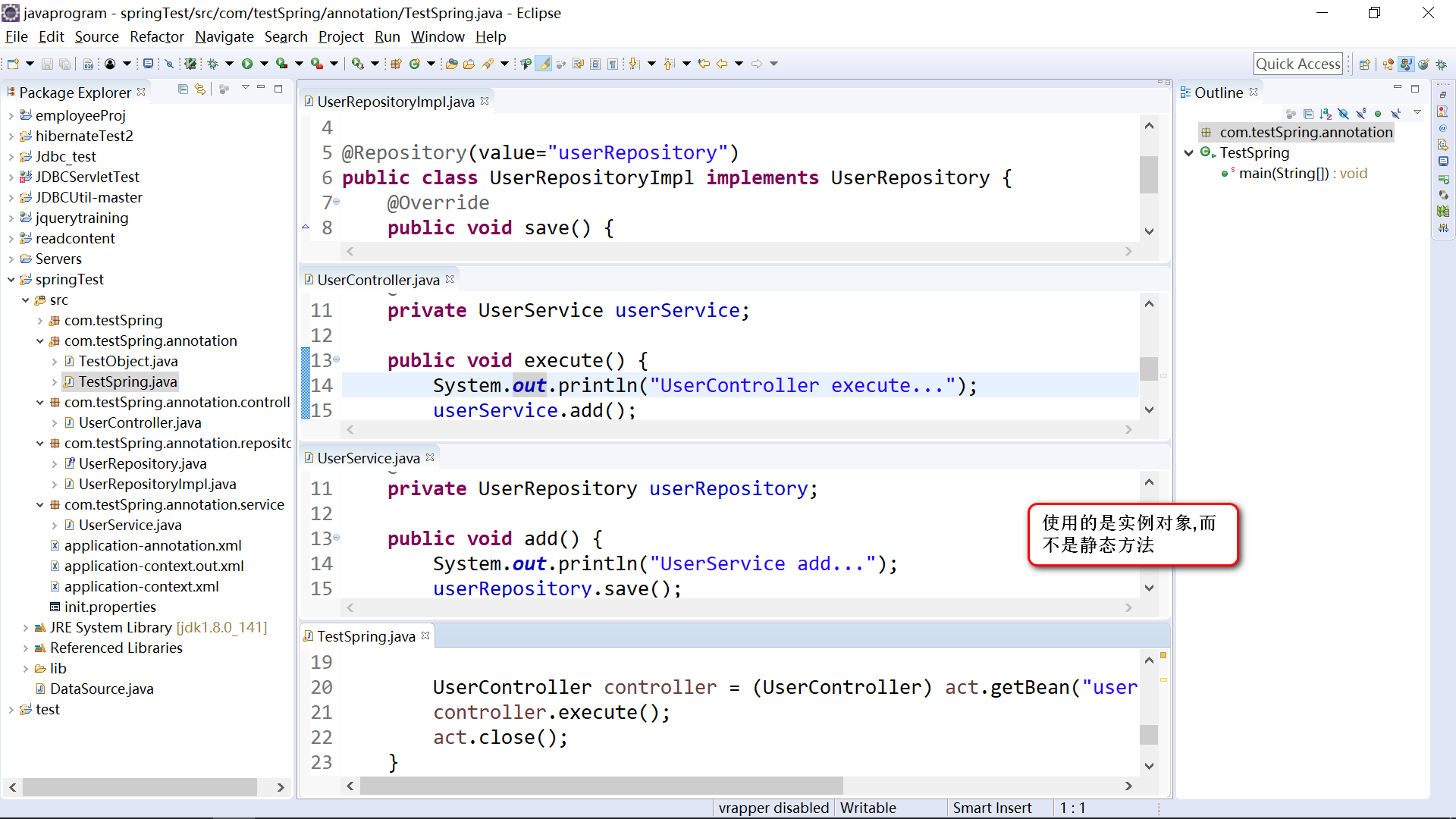
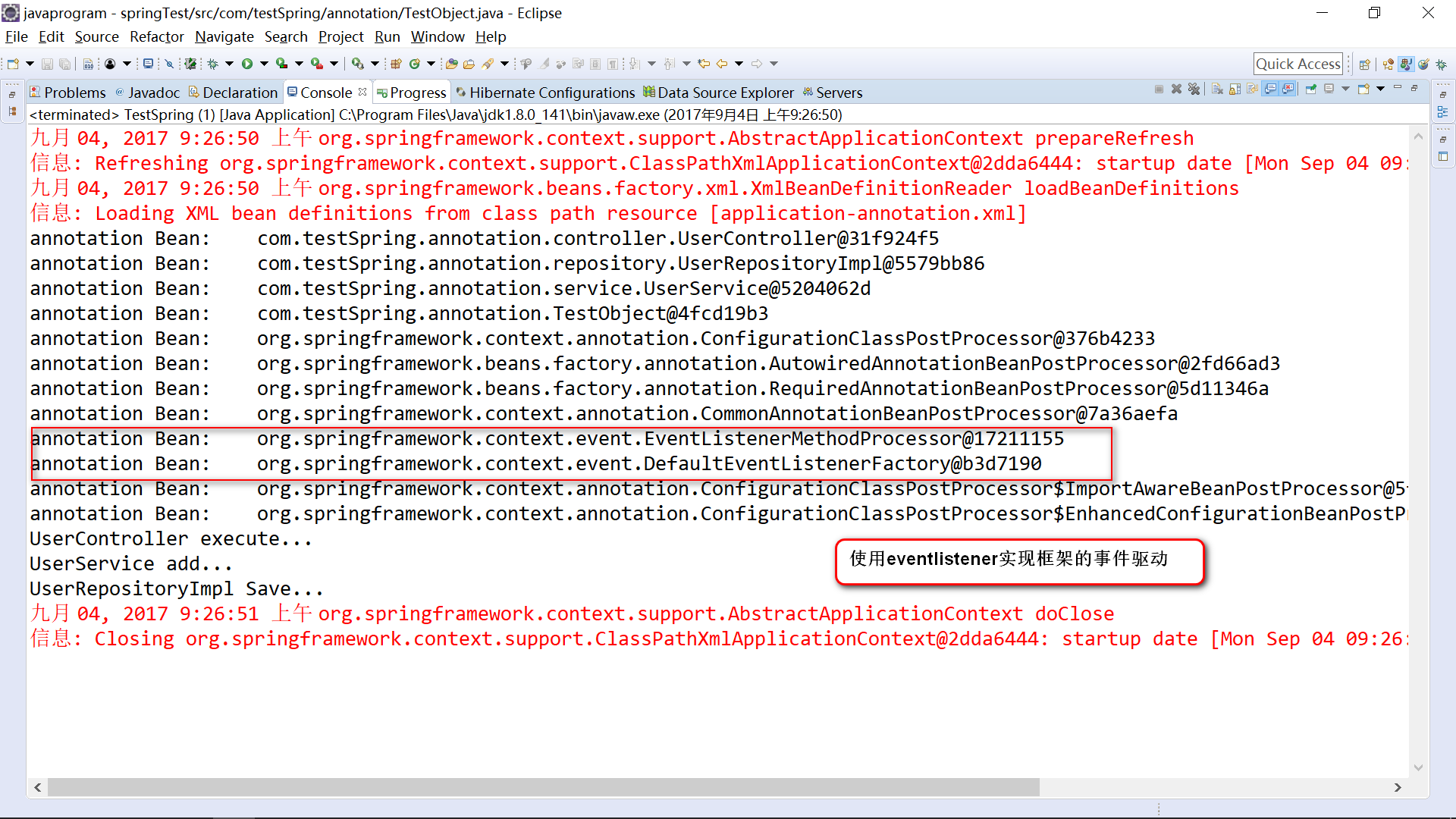
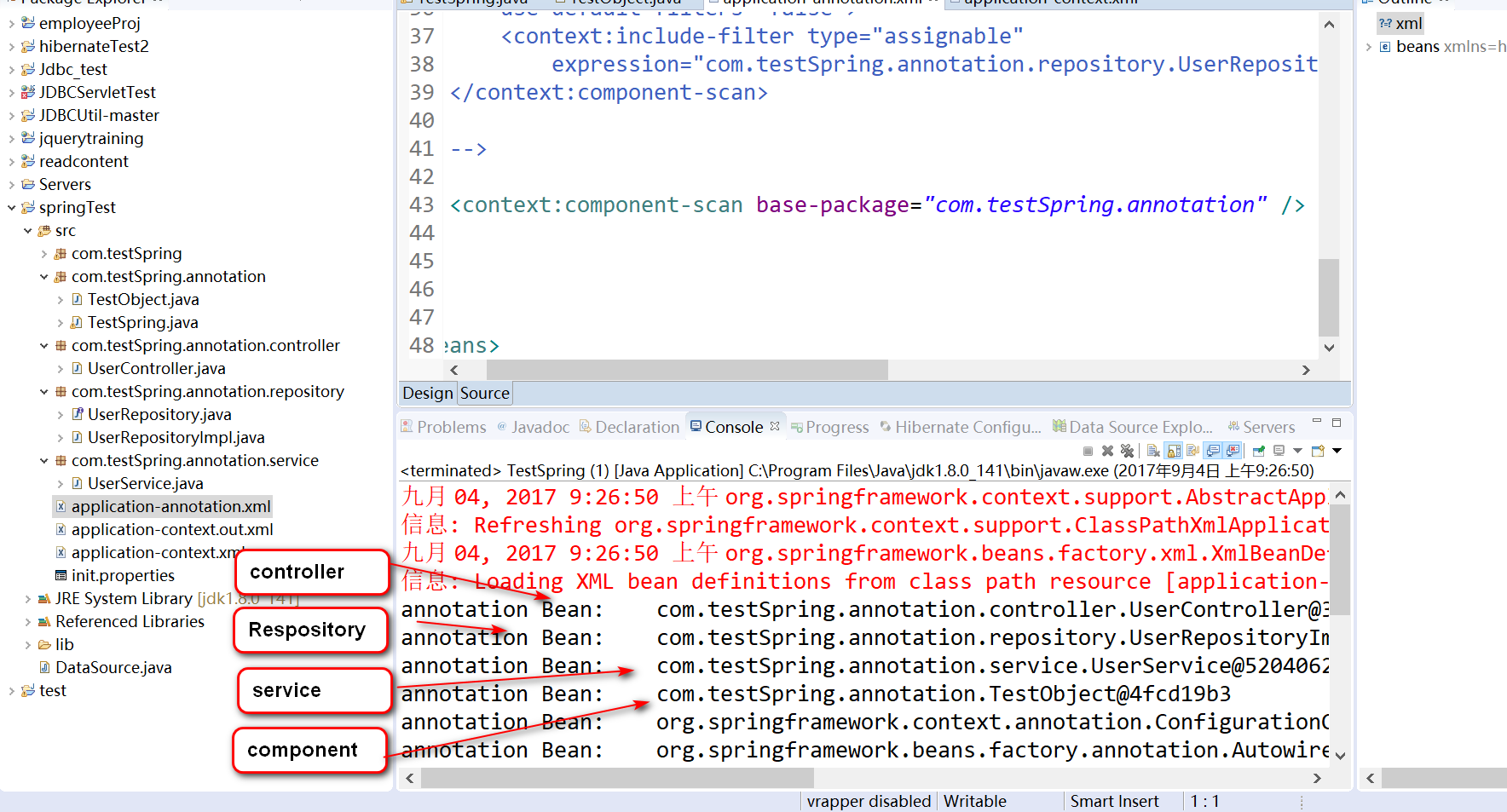
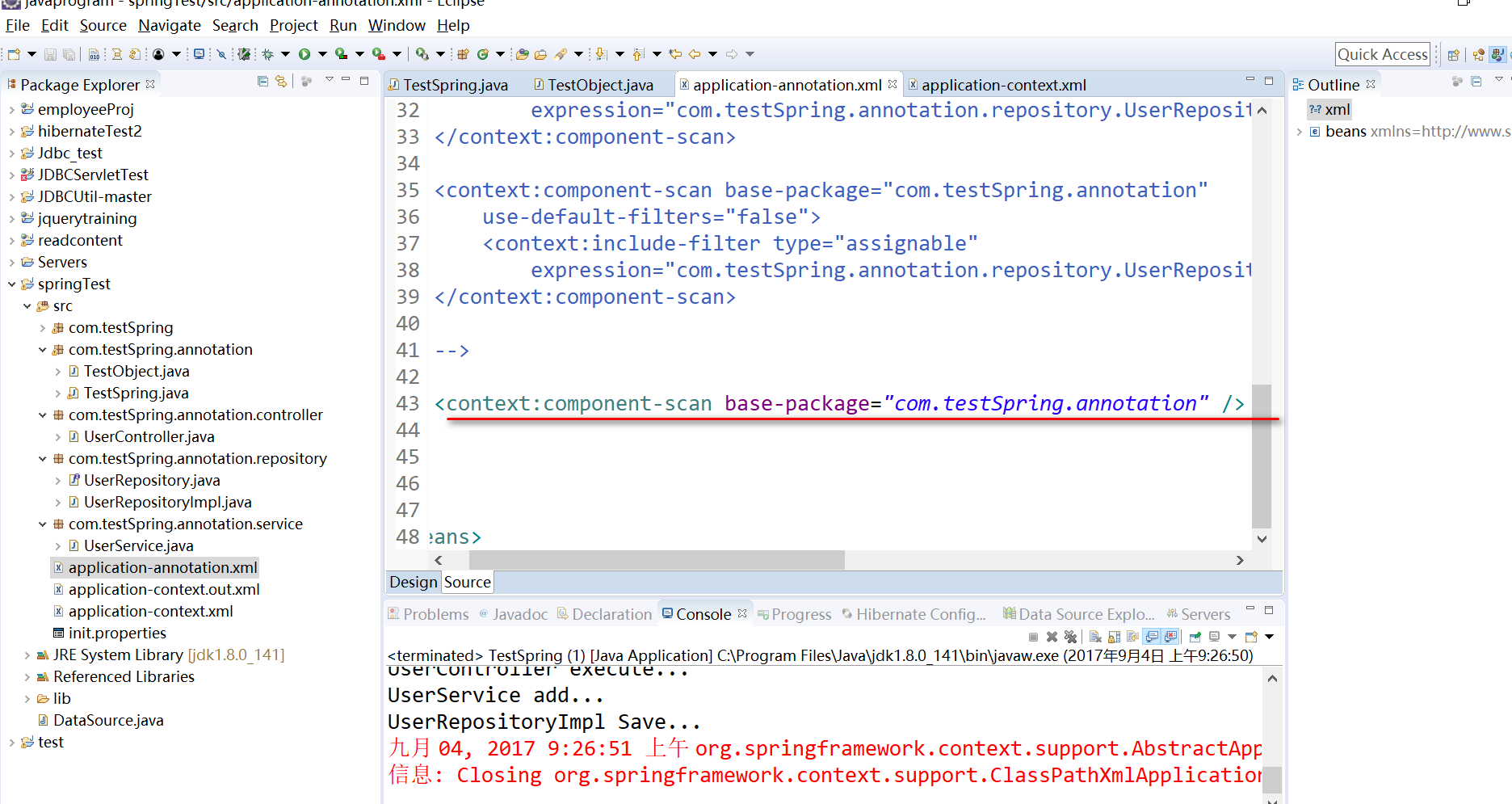
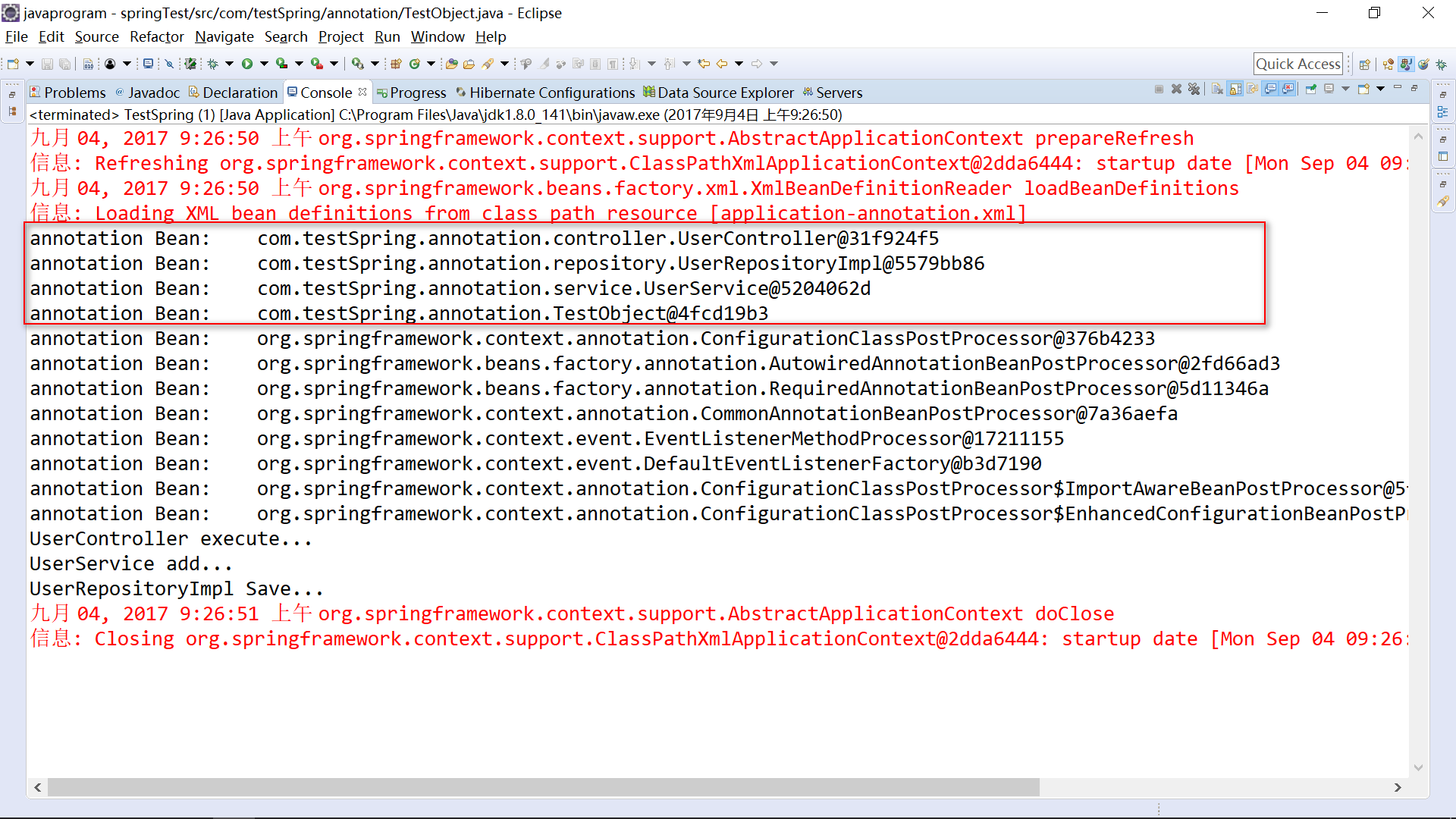


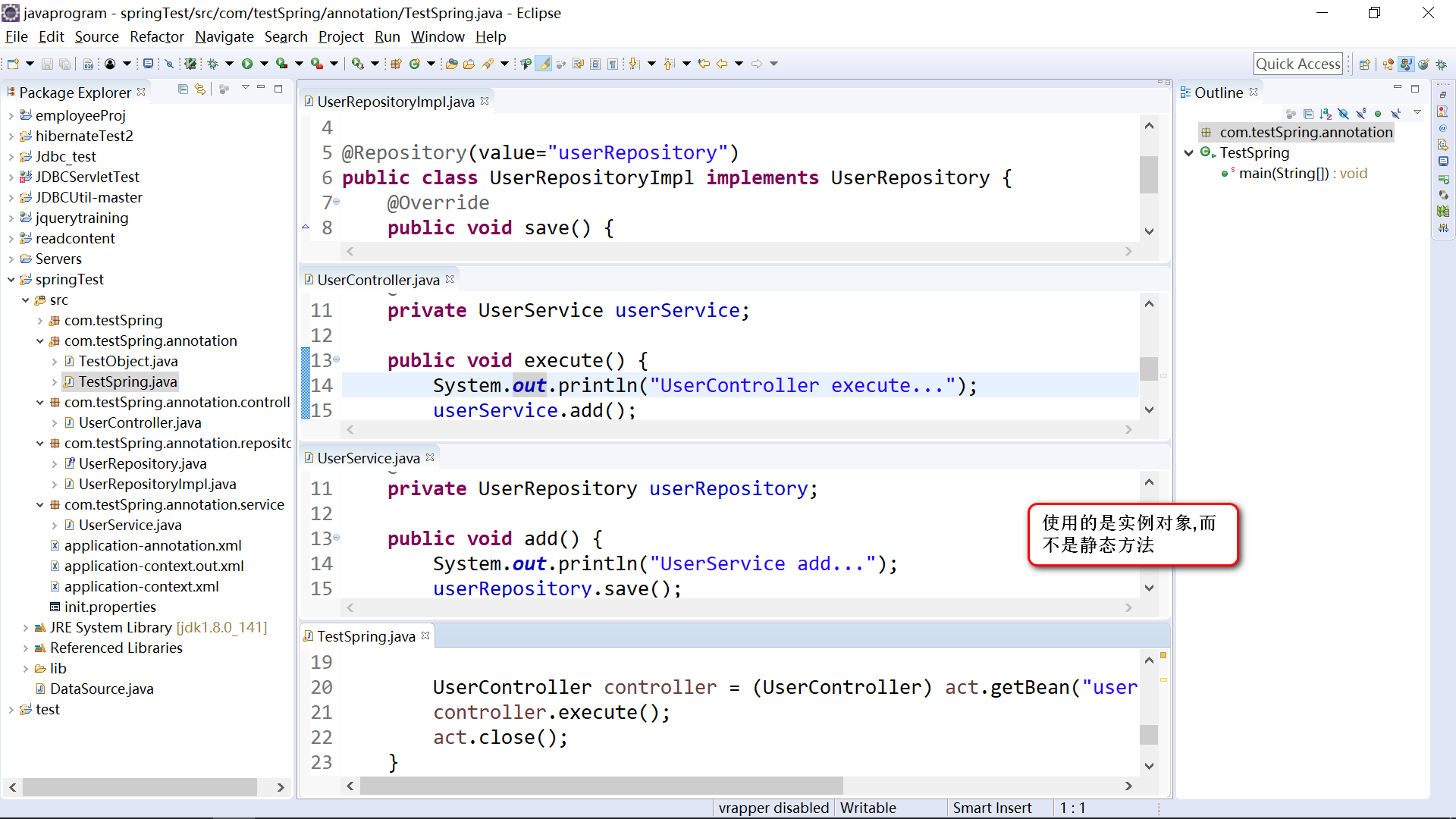
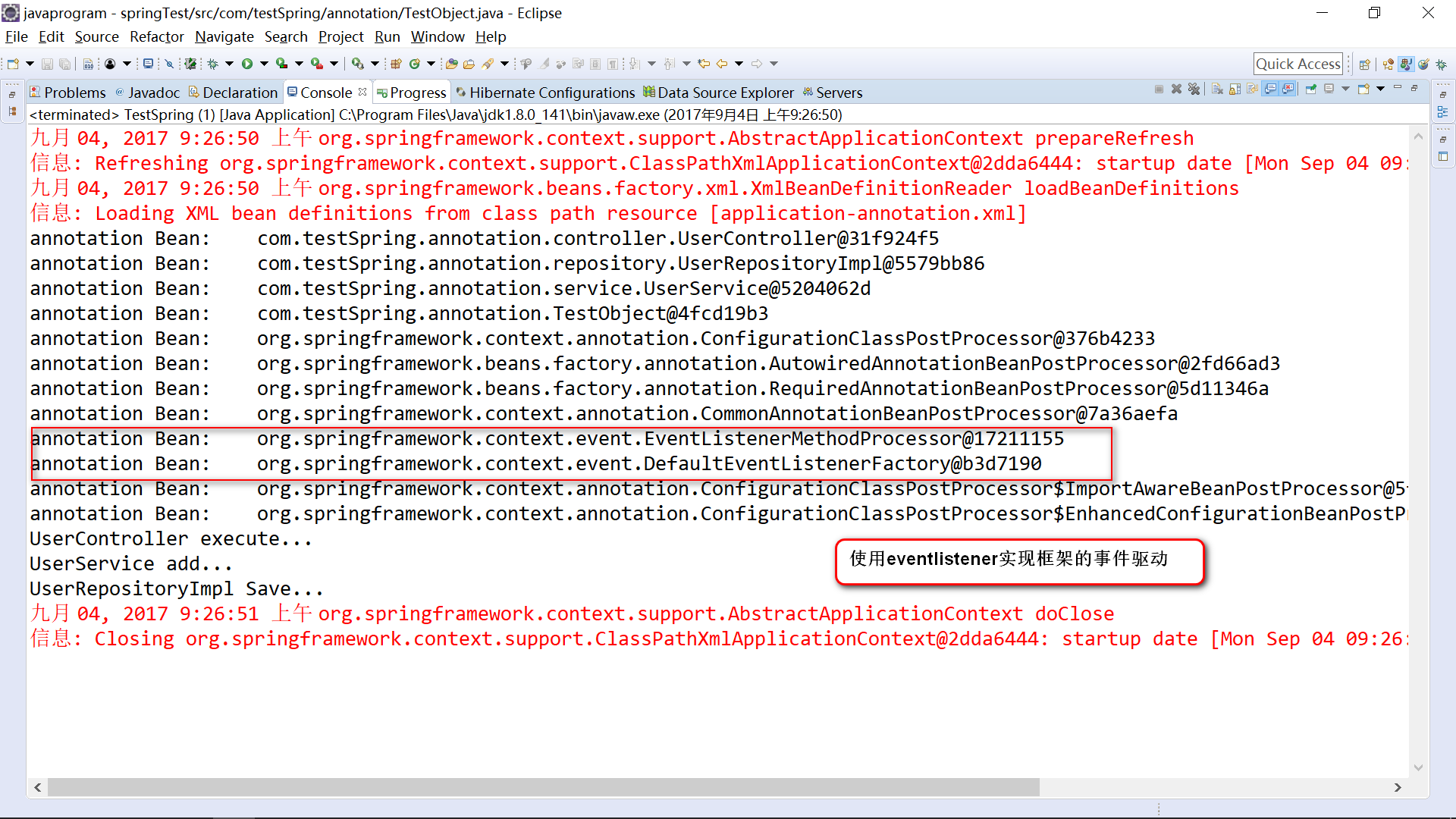
说明

1. 文件的加载顺序是 application-context.xml, import file,propertiy文件
2. 在调用applicationcontext对象的close方法的时候,会调用bean对象的destory方法
3. 配置文件中可以使用不同的id,多次引入同一个bean

## [spring <context:component-scan>使用说明](http://www.cnblogs.com/youngjoy/p/3817471.html)







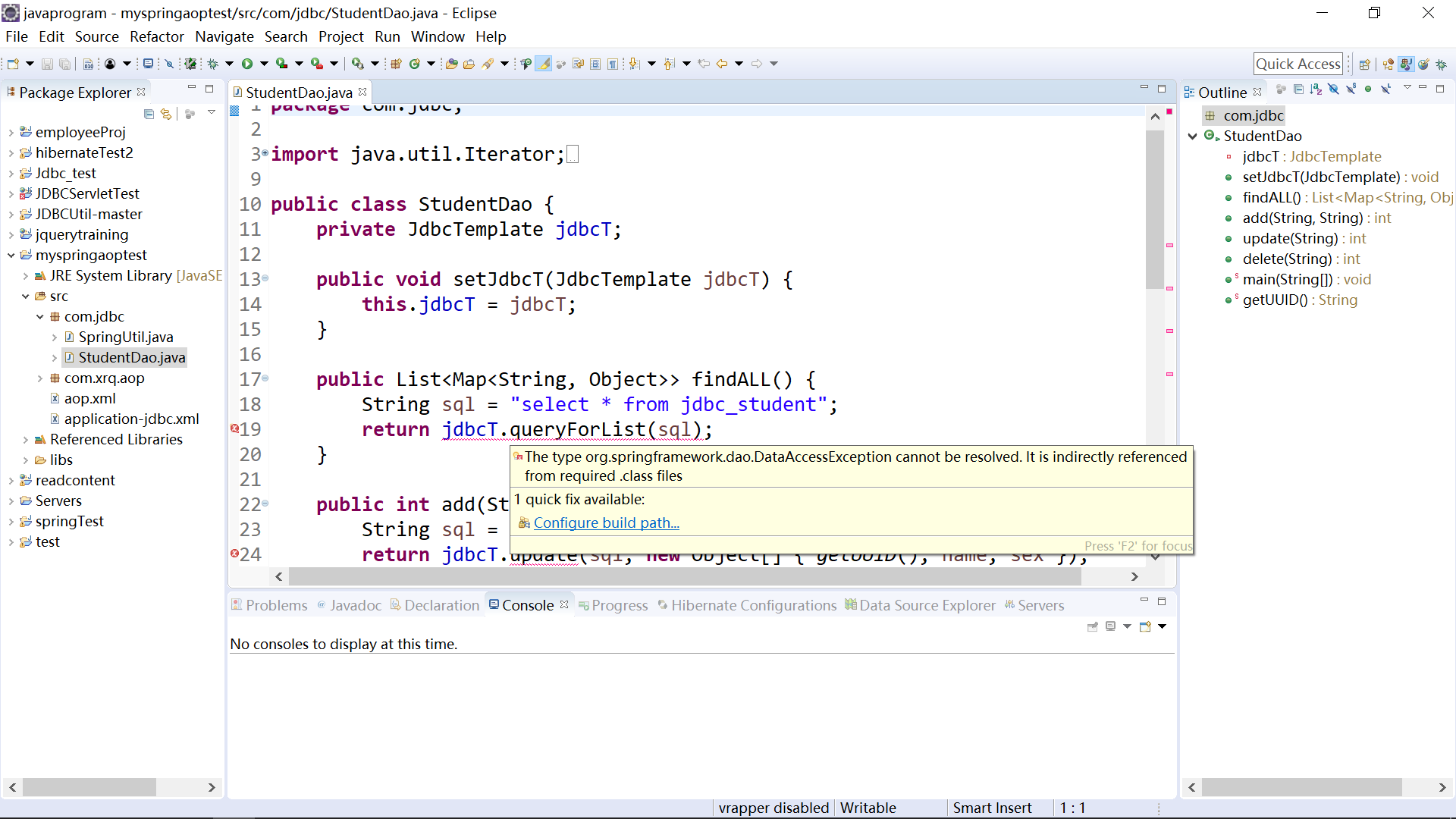
说明

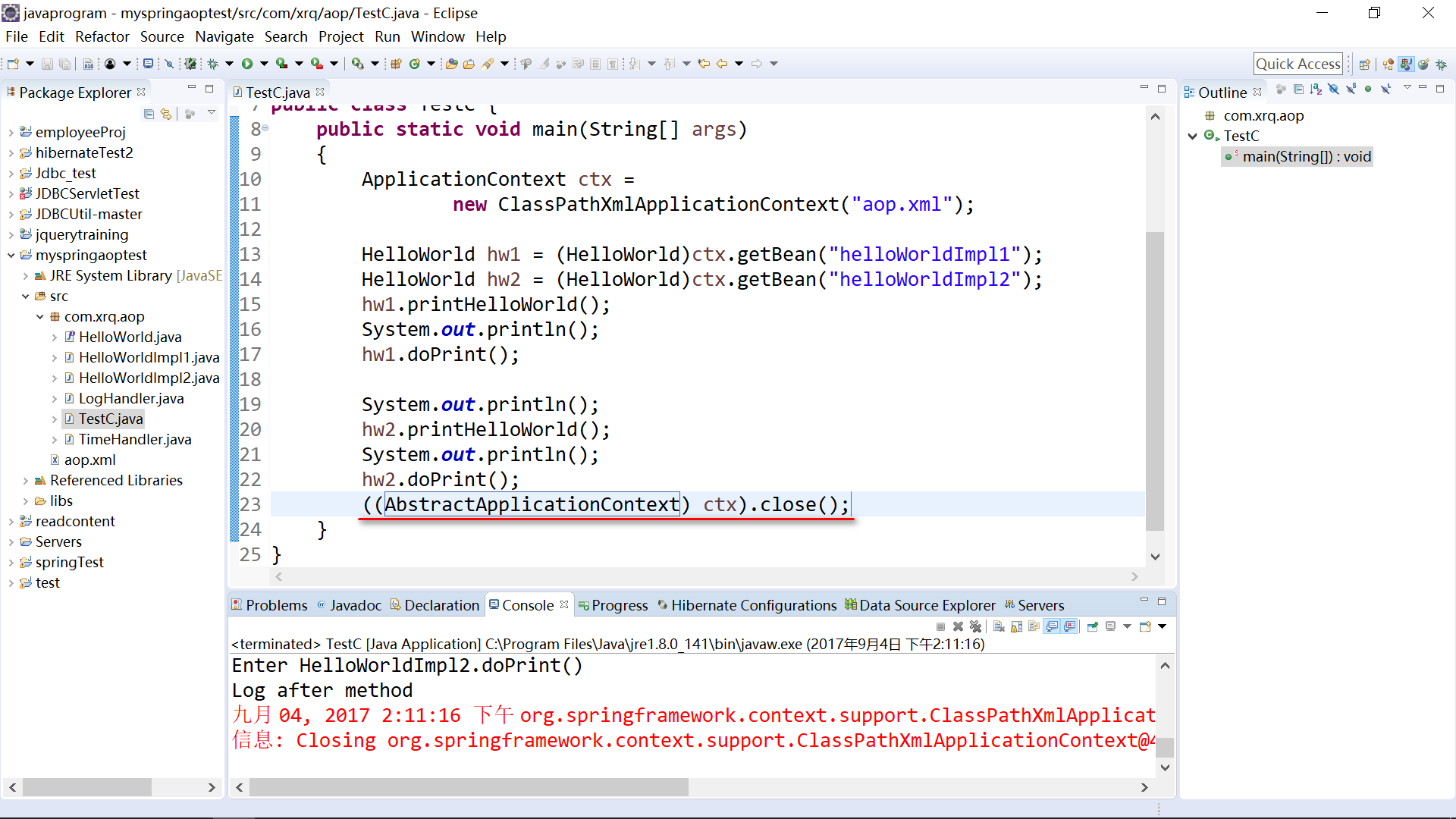
Sprng-aop的实现相比于正常的spring的区别是使用context-scan 引入了标签,同时自动调用默认构造函数,使用事件监听机制进行类驱动,初始化的顺序是 controller,resportity,service,compenent

## Spring 各个jar文件介绍

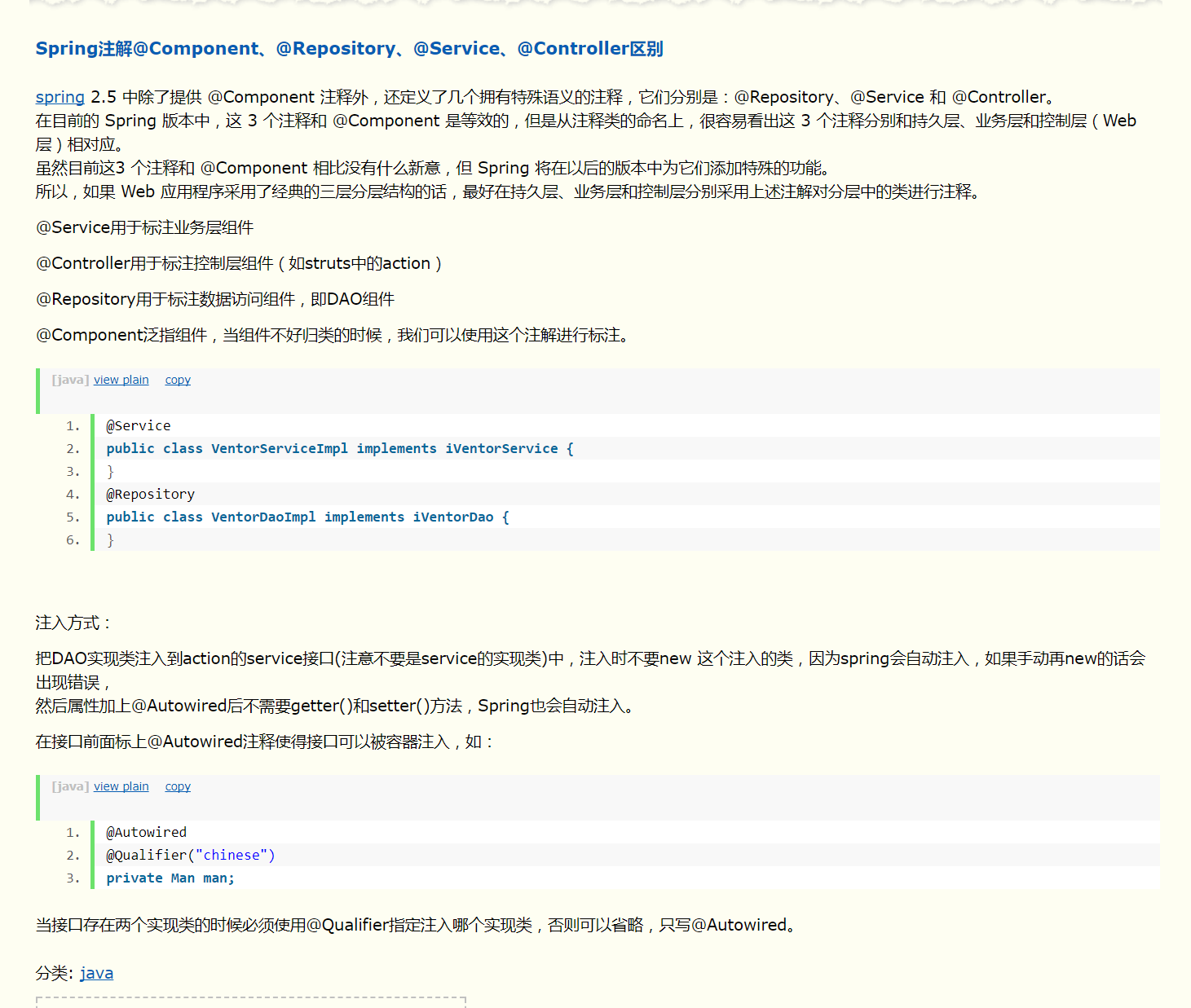


## 使用spring时的一些注意事项





## Spring context-scan注解的含义



## Spring的配置文件 applicationcontext.xml放置路径错误的报错

