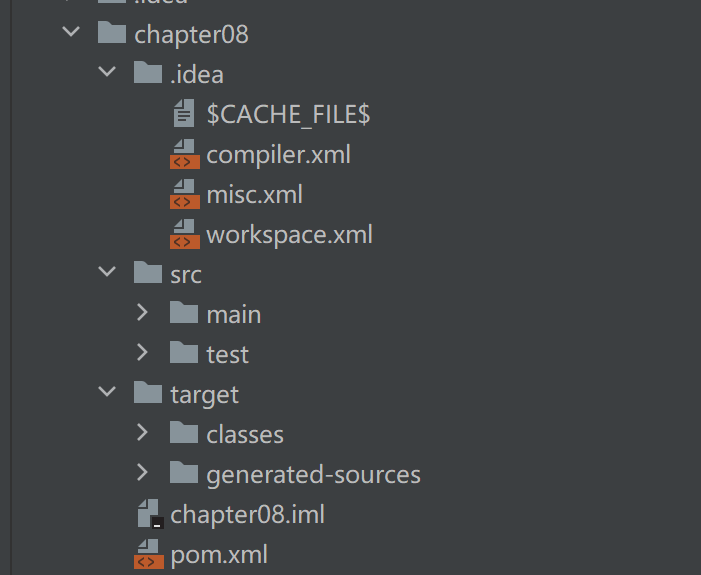
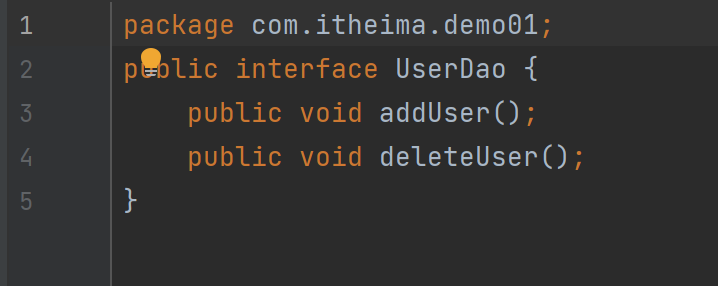
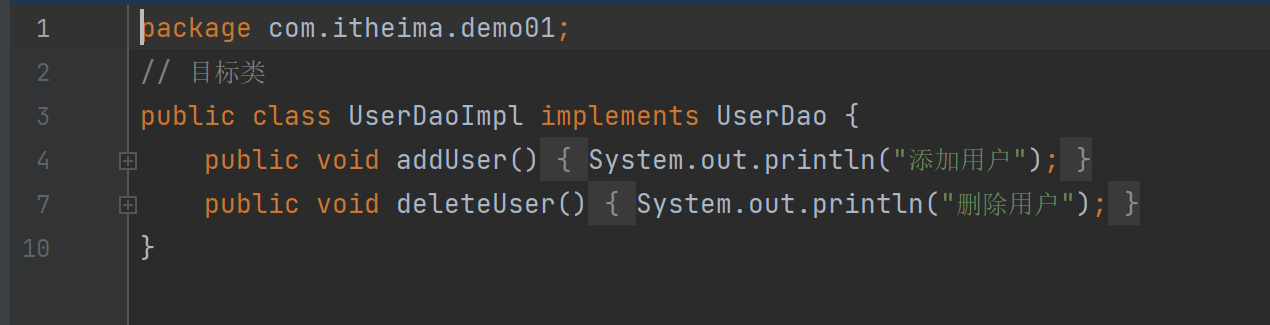
**Spring AOP的实现机制**

**（1）通过一个案例演示Spring中JDK动态代理的实现过程，案例具体实现步骤如下。**

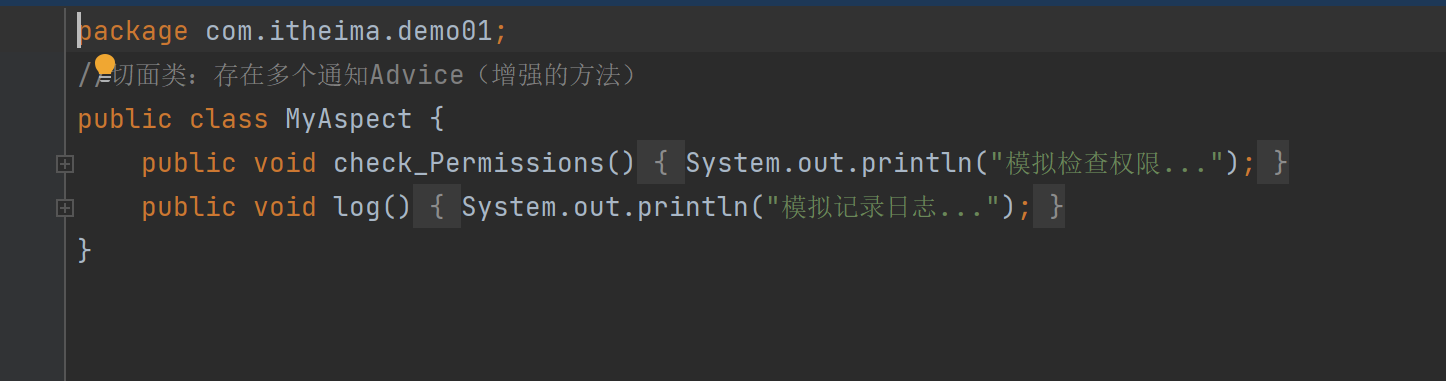
****

**（2）在项目的src/main/java目录下，创建一个com.itheima.demo01包，在该包下创建接口UserDao，在UserDao接口中编写添加和删除的方法。** 

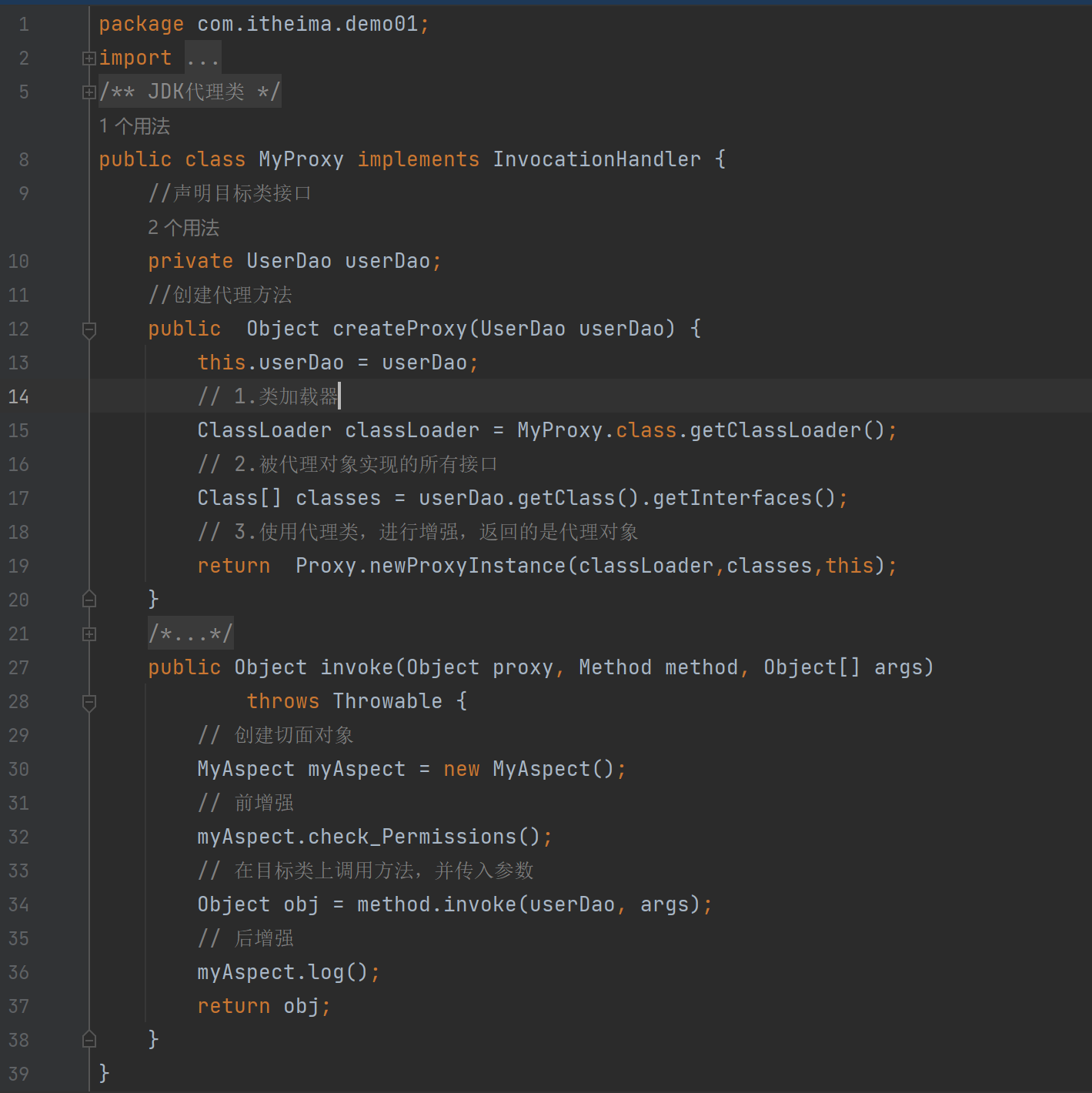
**（3）在com.itheima.demo1包中，创建UserDao接口的实现类UserDaoImpl，分别实现接口中的方法。**



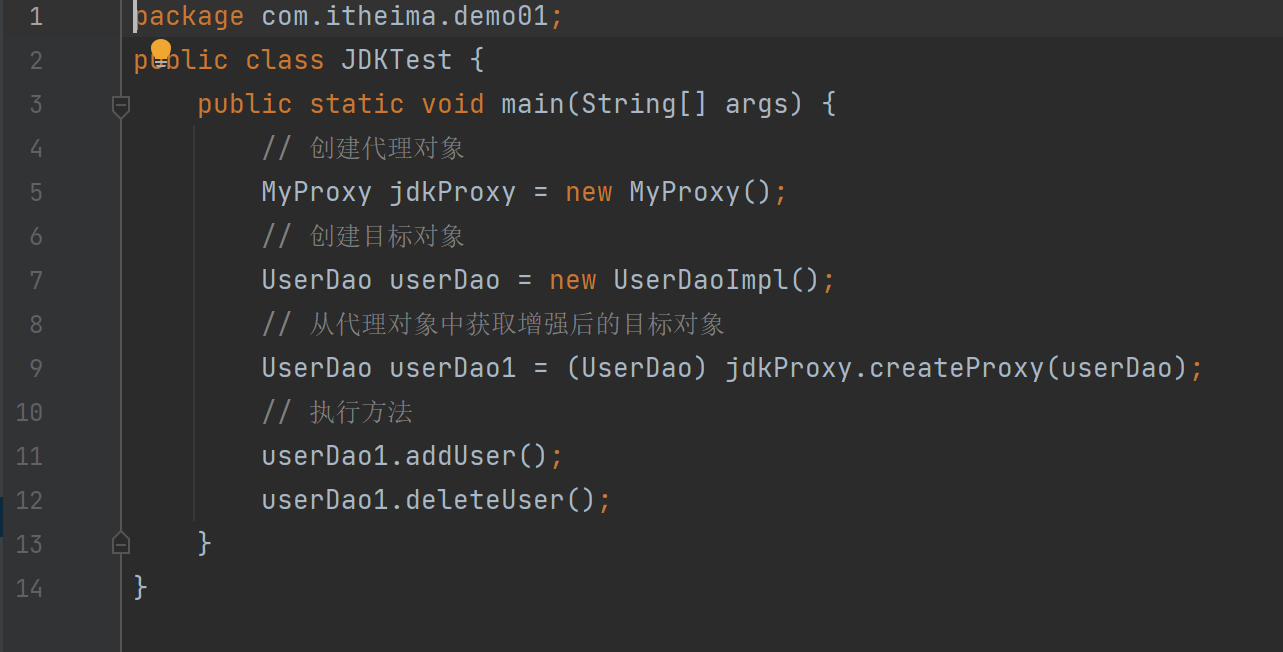
**（4）在com.itheima.demo01包下创建切面类MyAspect，在该类中定义一个模拟权限检查的方法和一个模拟日志记录的方法，这两个方法就是切面中的通知。**



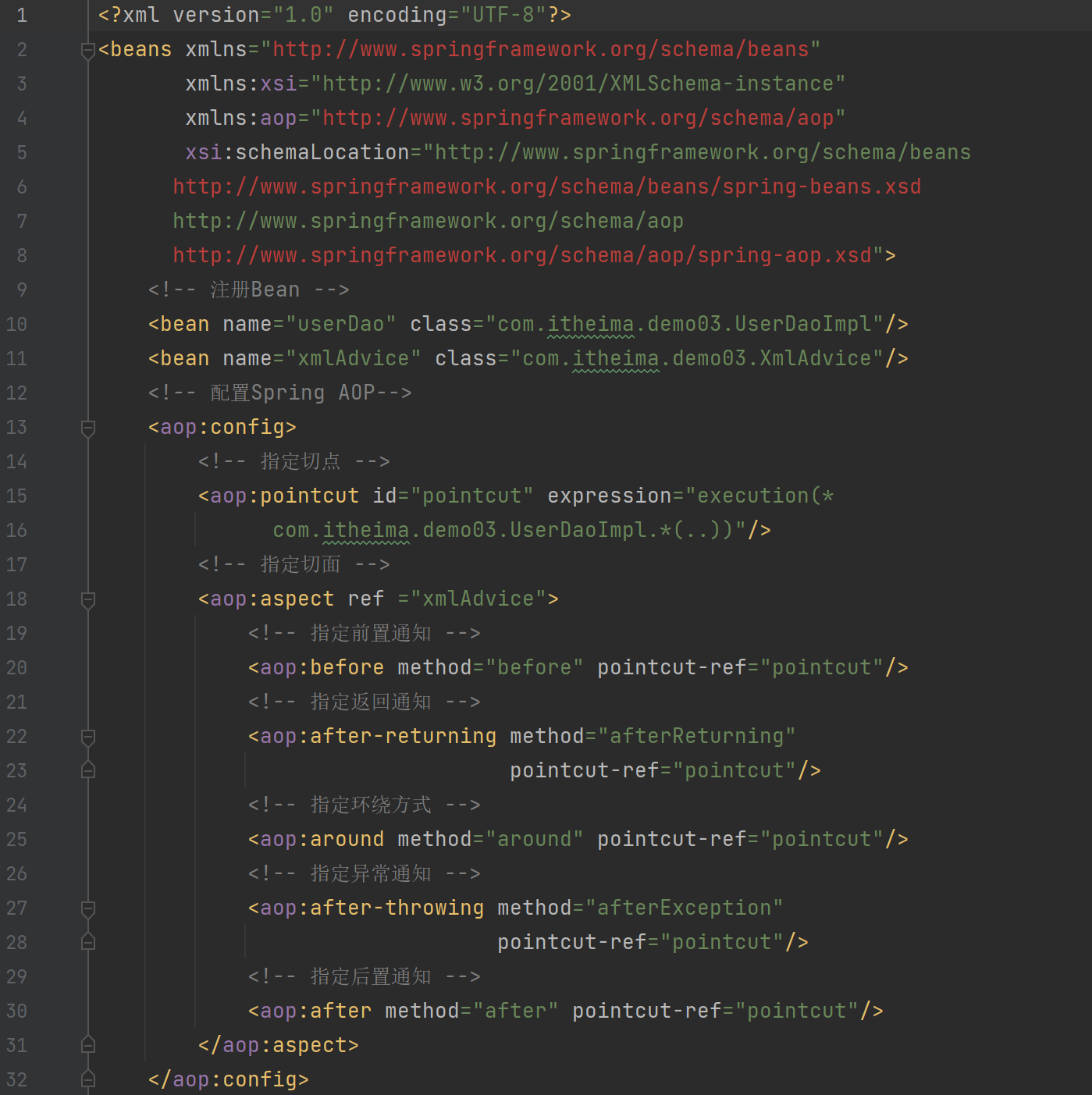
**（5）在com.itheima.demo01包下创建代理类MyProxy，该类需要实现InvocationHandler接口设置代理类的调用处理程序。在代理类中，通过newProxyInstance()生成代理方法。**



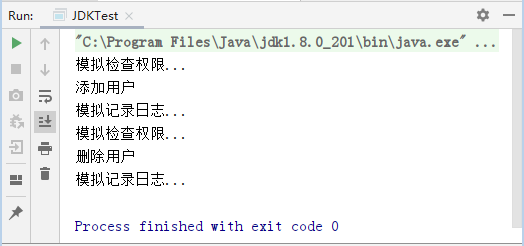
**（6）JDK动态代理：在com.itheima.demo01包中，创建测试类JDKTest。在该类中的main()方法中创建代理对象jdkProxy和目标对象userDao，然后从代理对象jdkProxy中获得对目标对象userDao增强后的对象userDao1，最后调用userDao1对象中的添加和删除方法。**



**（7）在chapter06项目的com.itheima文件夹下创建测试类TestHelloSpring，在main()方法中初始化Spring容器并加载applicationContext.xml配置文件，通过Spring容器获取HelloSpring类的helloSpring实例，调用HelloSpring类中的show()方法在控制台输出信息。**

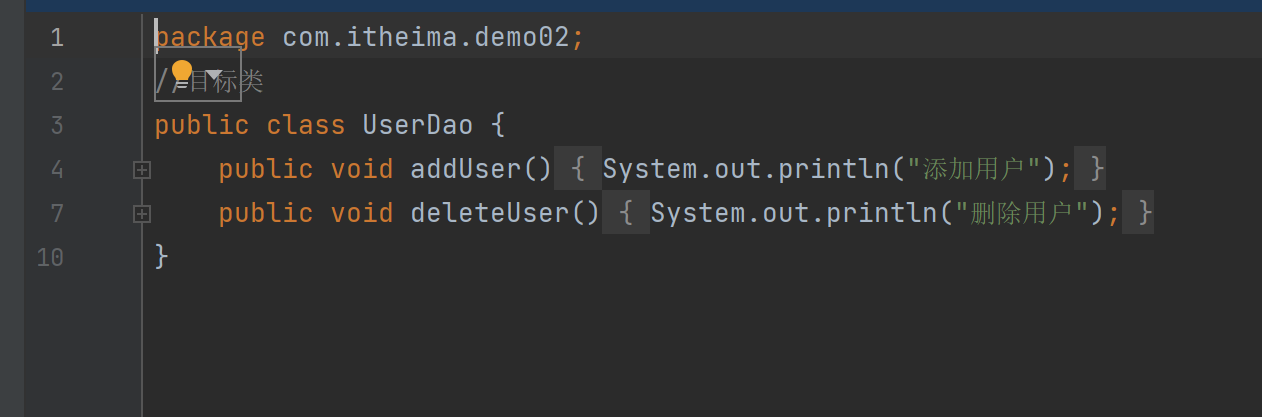


**（8）在IDEA中启动JDKTest类，控制台会输出结果。**

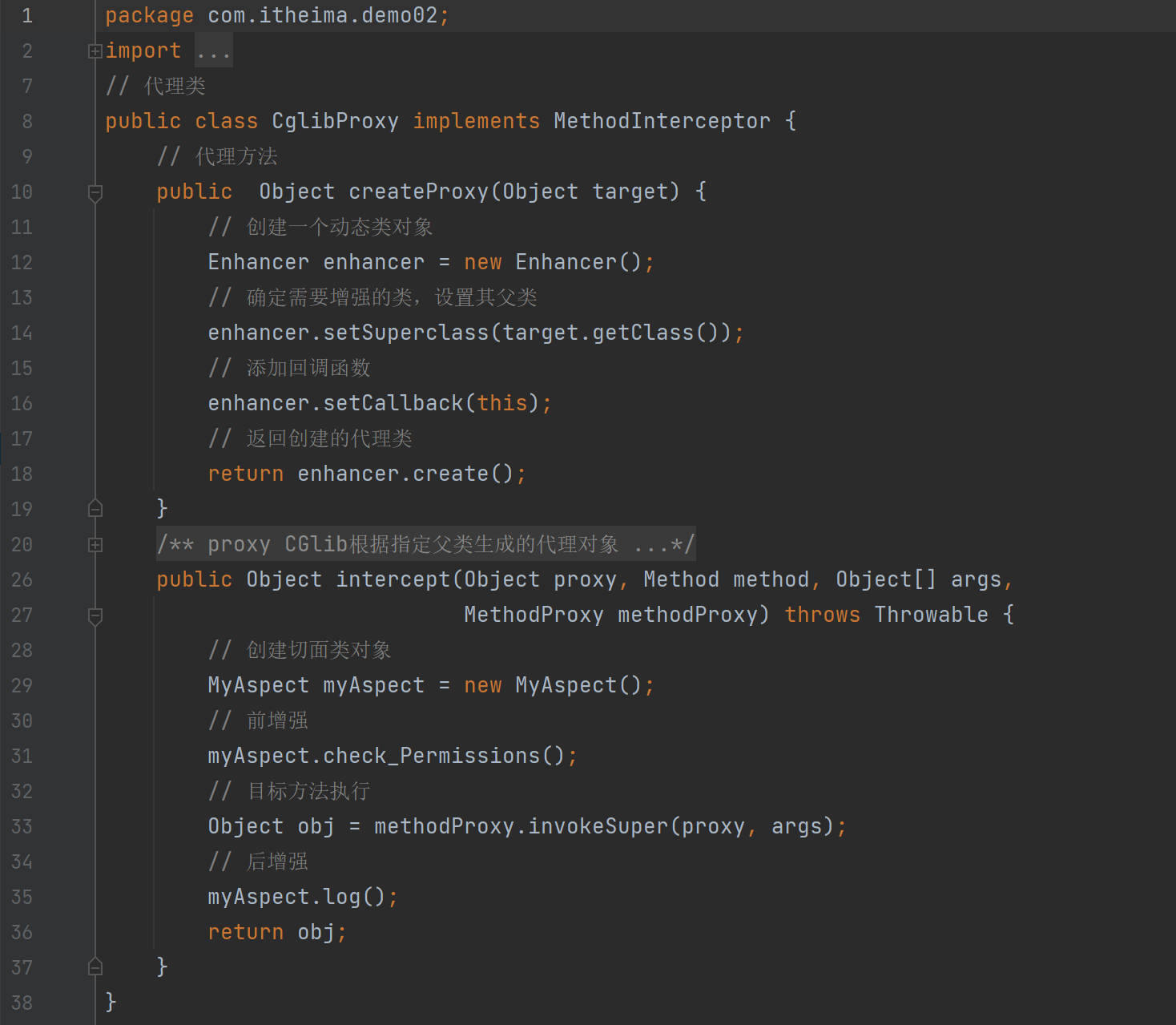
****

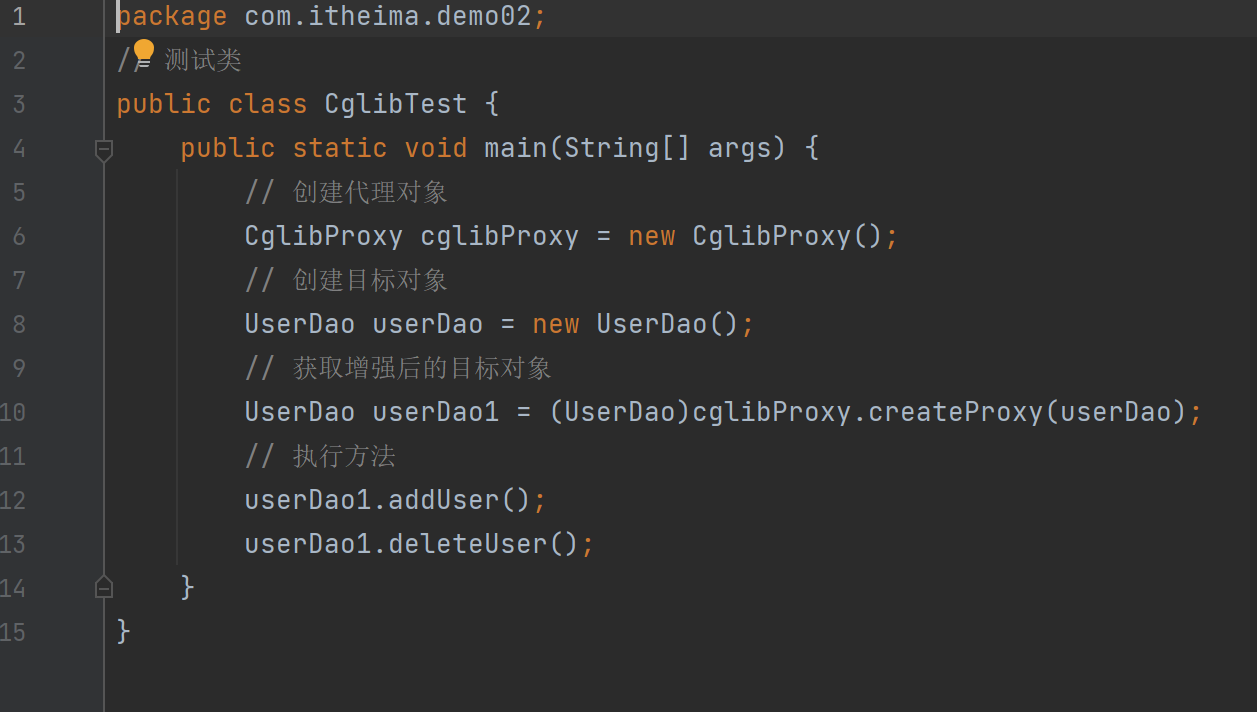
**CGLib动态代理**

**（1）在chapter08项目的src/main/java目录下创建一个com.itheima.demo02包，在该包下创建目标类UserDao，在该类中编写添加用户和删除用户的方法。**

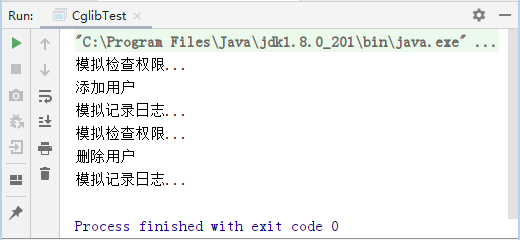


**（2）在com.itheima.demo02包下创建代理类CglibProxy，该代理类需要实现MethodInterceptor接口用于设置代理类的调用处理程序，并实现接口中的intercept()方法。**



**（3）在com.itheima.demo02包中创建测试类CglibTest，在main()方法中首先创建代理对象cglibProxy和目标对象userDao，然后从代理对象cglibProxy中获得增强后的目标对象userDao1，最后调用userDao1对象的添加和删除方法。**

**（4）在IDEA中启动CglibTest类，控制台会输出结果。**

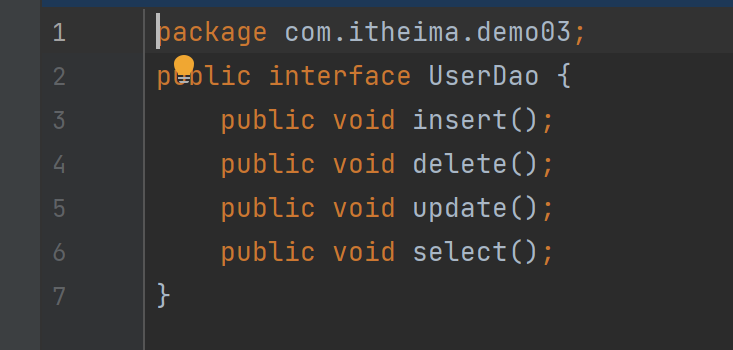
****

**基于XML的AOP实现**

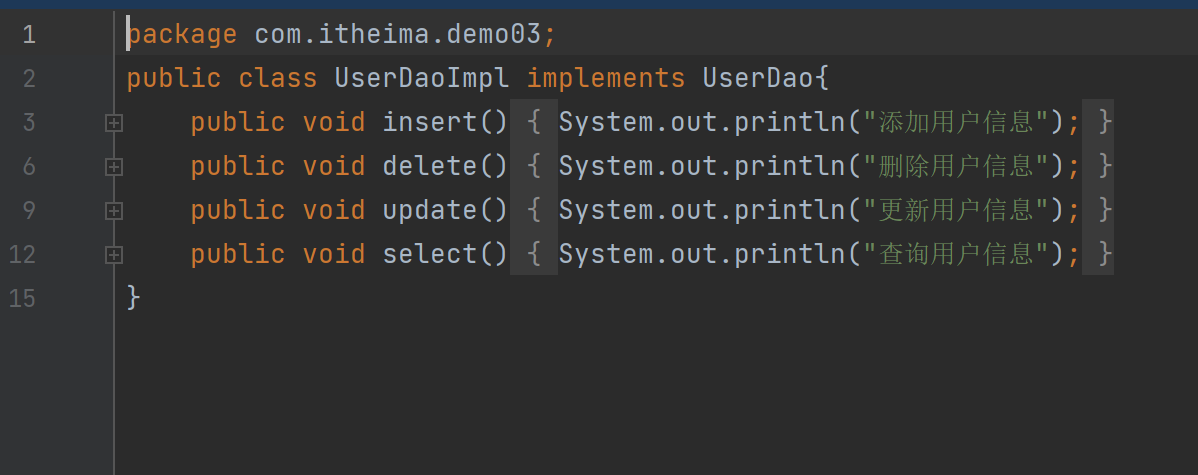
**（1）在chapter08项目的pom.xml文件中导入AspectJ框架的相关JAR包。**



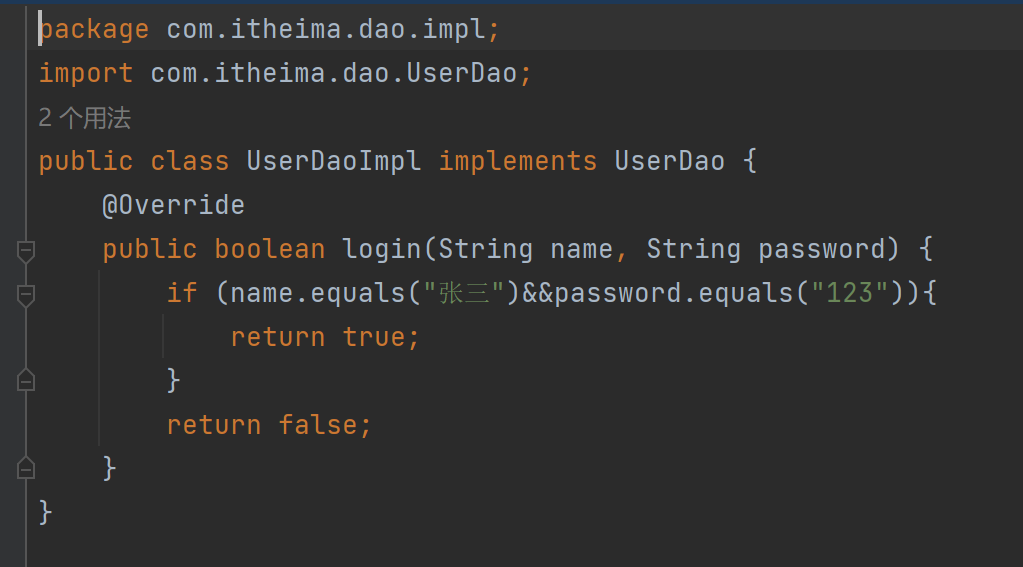
**（2）在chapter08项目的src/main/java目录下创建一个com.itheima.demo03包，在该包下创建接口UserDao，并在该接口中编写添加、删除、修改和查询的方法。**



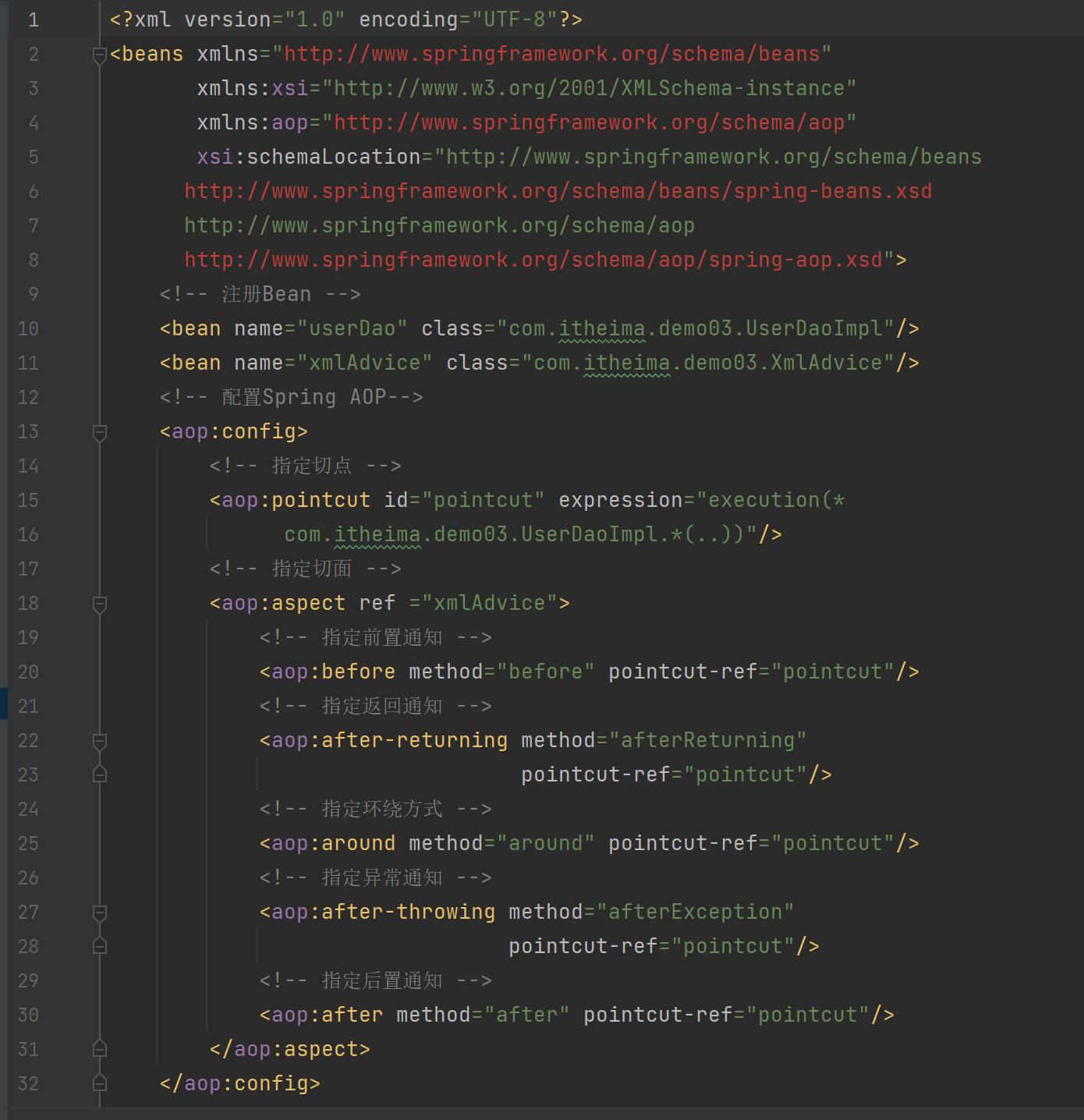
**（3）在com.itheima.demo03包下创建UserDao接口的实现类UserDaoImpl，实现UserDao接口中的方法。**



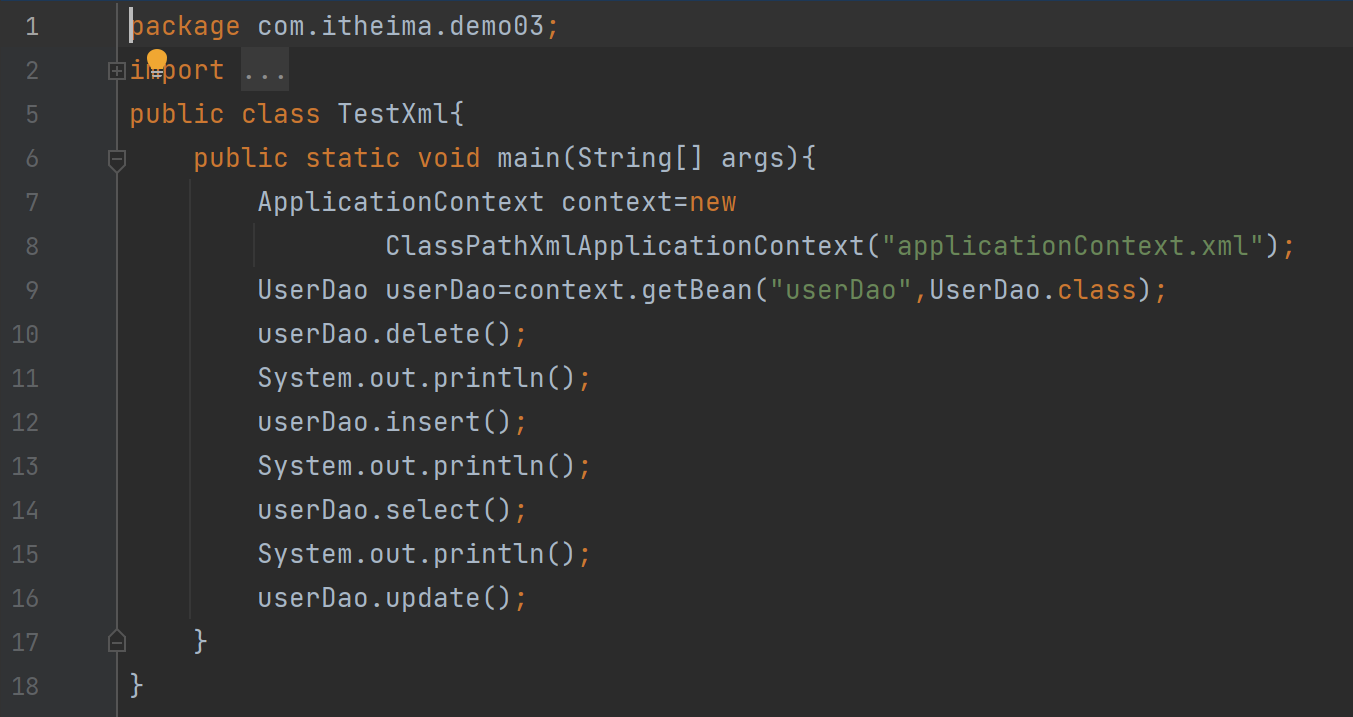
**（4）在com.itheima.demo03包下创建XmlAdvice类，用于定义通知。**

****

**（5）在chapter08项目的resource文件夹下创建applicationContext.xml文件，在该文件中引入AOP命名空间，使用<bean>元素添加Spring AOP的配置信息。**



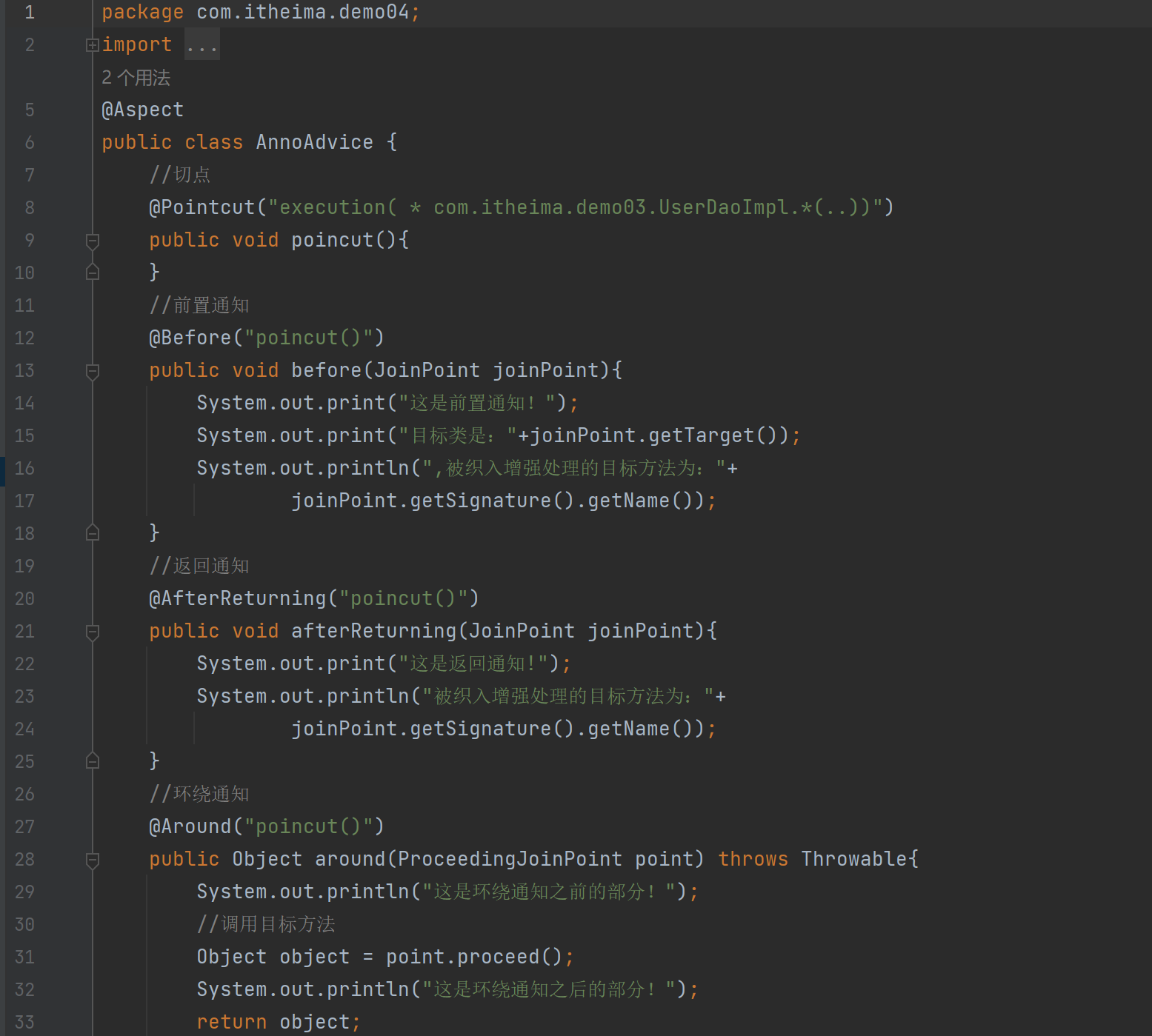
**（6）在com.itheima.demo03包中创建测试类TestXml。**



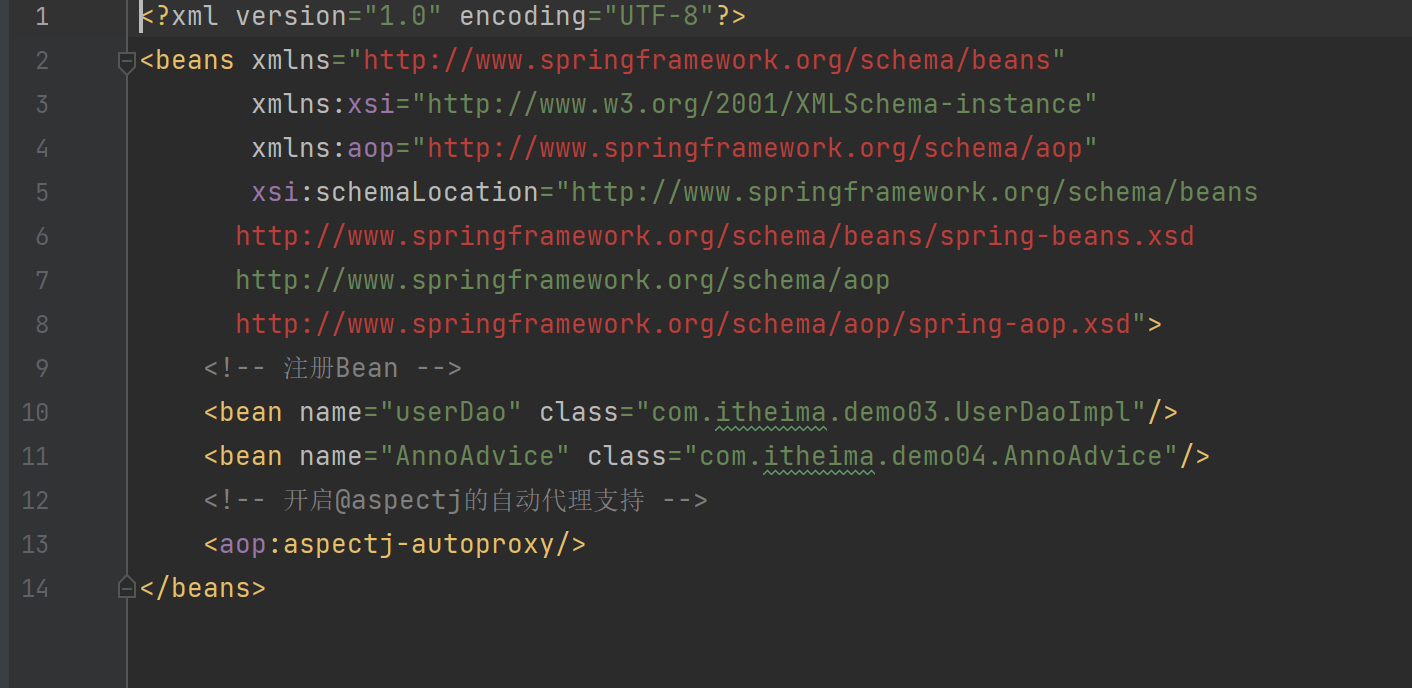
**（7）在IDEA中启动TestXml类，控制台会输出结果。**

****

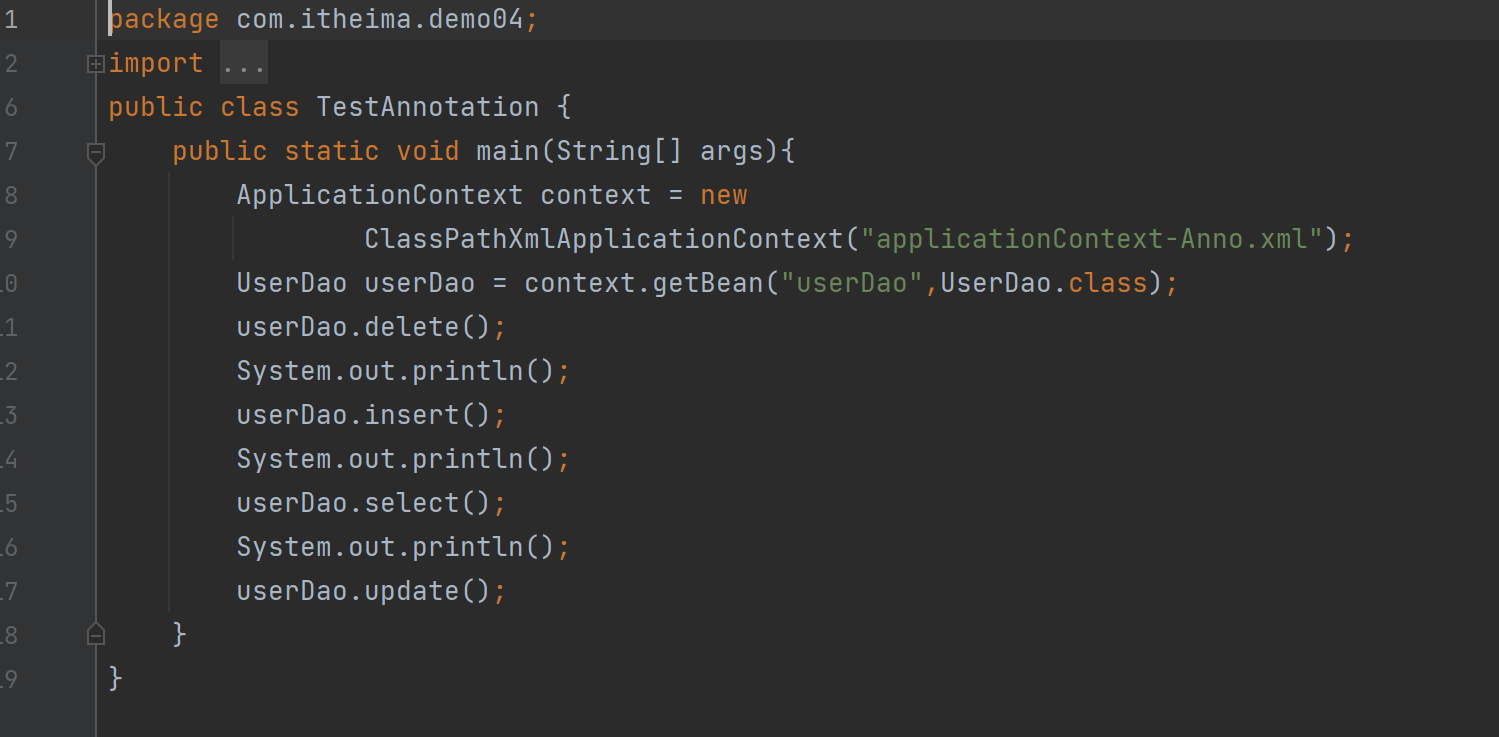
**基于注解的AOP实现**

**（1）在chapter08项目的src/main/java目录下创建一个com.itheima.demo04包，在该包下创建AnnoAdvice类，用于定义通知。**

**（2）在chapter08项目的resource文件夹下创建applicationContext-Anno.xml文件，在该文件中引入AOP命名空间，使用<bean>元素添加Spring AOP的配置信息。**



**（3）在com.itheima.demo04包中创建测试类TestAnnotation。**



**（4）在IDEA中启动TestAnnotation类，控制台会输出结果。**

****