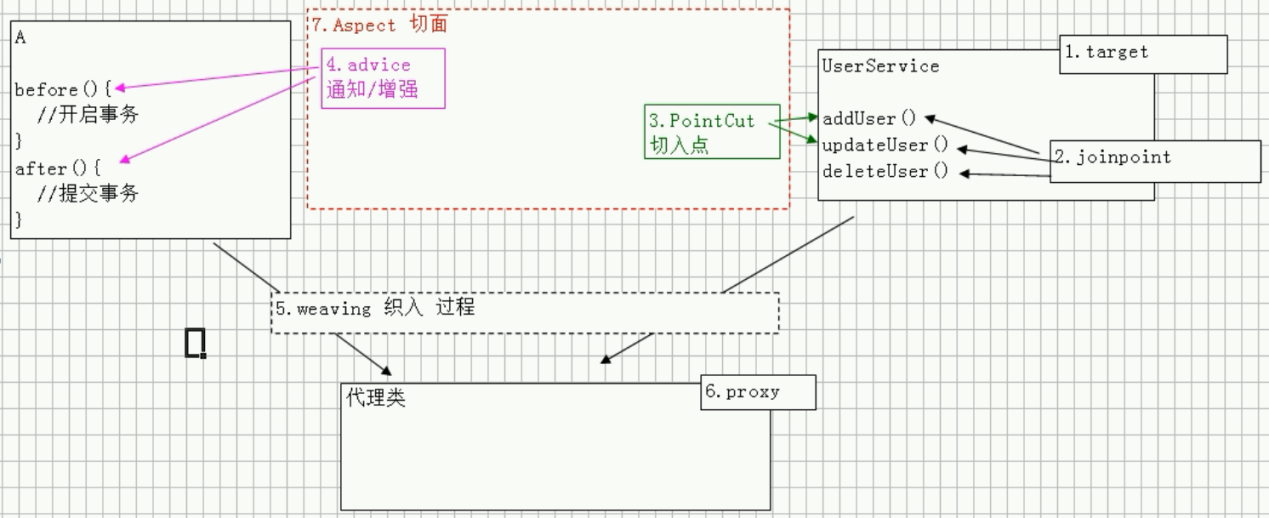
## Spring框架（AOP）

1. AOP介绍
2. 什么是AOP
3. AOP为Aspect Oriented Programming的缩写，意为：[面向切面编程](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%88%87%E9%9D%A2%E7%BC%96%E7%A8%8B/6016335" \t "https://baike.baidu.com/item/AOP/_blank)，是[OOP](https://baike.baidu.com/item/OOP" \t "https://baike.baidu.com/item/AOP/_blank)的延续，利用AOP可以对业务逻辑的各个部分进行隔离，从而使得业务逻辑各部分之间的[耦合度](https://baike.baidu.com/item/%E8%80%A6%E5%90%88%E5%BA%A6/2603938" \t "https://baike.baidu.com/item/AOP/_blank)降低，提高程序的可重用性，提高开发的效率。
4. AOP采取横向抽取机制，取代了传统纵向继承体系重复性代码
5. AOP应用场景：事务管理、性能监视、安全检查、缓存、日志等
6. AOP的实现原理
7. aop底层采用代理机制进行实现

2）接口+实现类：spring采用jdk的**动态代理proxy**

3）实现类：spring采用**cglib字节码增强**

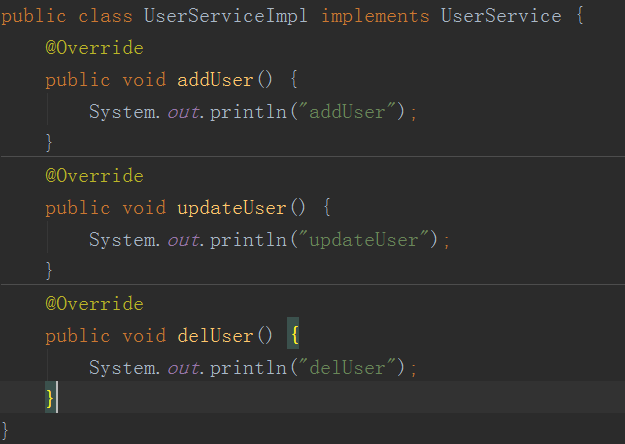
1. AOP术语（重点掌握）
2. target：目标类，需要被代理的类
3. Joinpoint：连接点，指可能被拦截的方法。如：所有方法
4. PointCut：切入点，已经被增强的连接点。如：addUser()
5. Advice：通知/增强，增强代码。如：after、before
6. Weaving：织入，指把增强应用到目标对象来创建新的代理对的过程
7. Proxy：代理类
8. Aspect：切面，是切入点和增强的结合（多点之间结合就成了面）



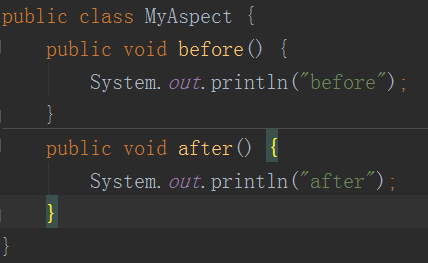
说明：代理的意思就是不能直接访问某个类，需要通过代理对象去访问。

举例：不能直接去见刘德华，而是要通过刘德华的经纪人去见刘德华。

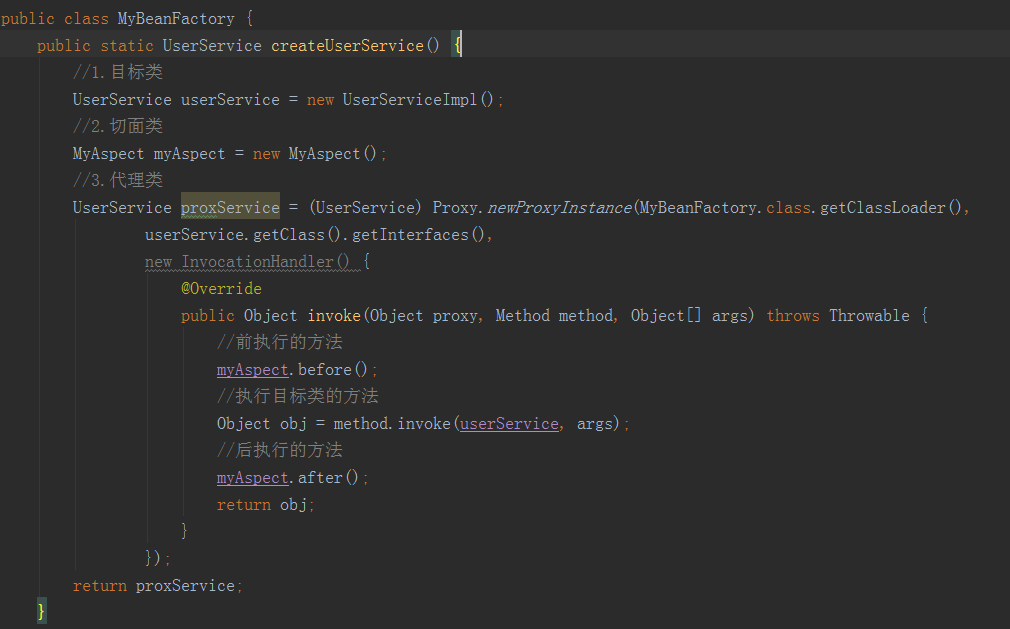
1. 手动配置代理
2. jdk动态代理（接口+实现类）
3. 目标类



1. 切面类



1. 代理类



1. jdk动态代理相关知识（对上面代理的解释）

代理类：将目标类（切入点）和切面类（通知）结合-->切面

参数1：loader：类加载器，动态代理类运行时创建，任何类都需要类加载器将其加载到 内存中

一般写法：当前类.class.getClassLoader();

其他写法：目标实例.getClass().get...

参数2：interfaces：代理对象需要实现的所有接口

写法1：目标实例.getClass().getInterfaces()；注意，只能获得自己的接口，不能获 得父类接口

写法2：new Class[]{目标类.class}

参数3：InvocationHandler：处理类，是一个接口，需要实现类，实现类一般采用匿名 内部类

该处理类提供invoke()方法，该方法有3个参数：

参数1：proxy：代理对象

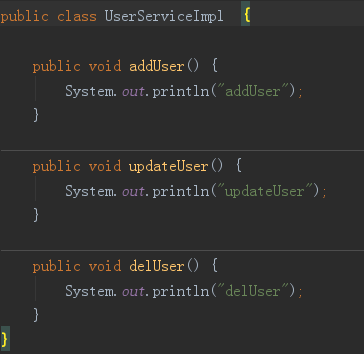
参数2：method：代理对象当前执行的方法的描述对象（反射知识）

\* 获得方法名：method.getName()

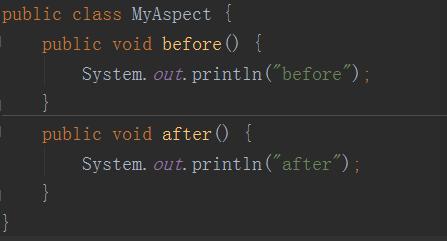
\* 执行方法：method.invoke（对象，实际参数）

参数3：args：方法的实际参数

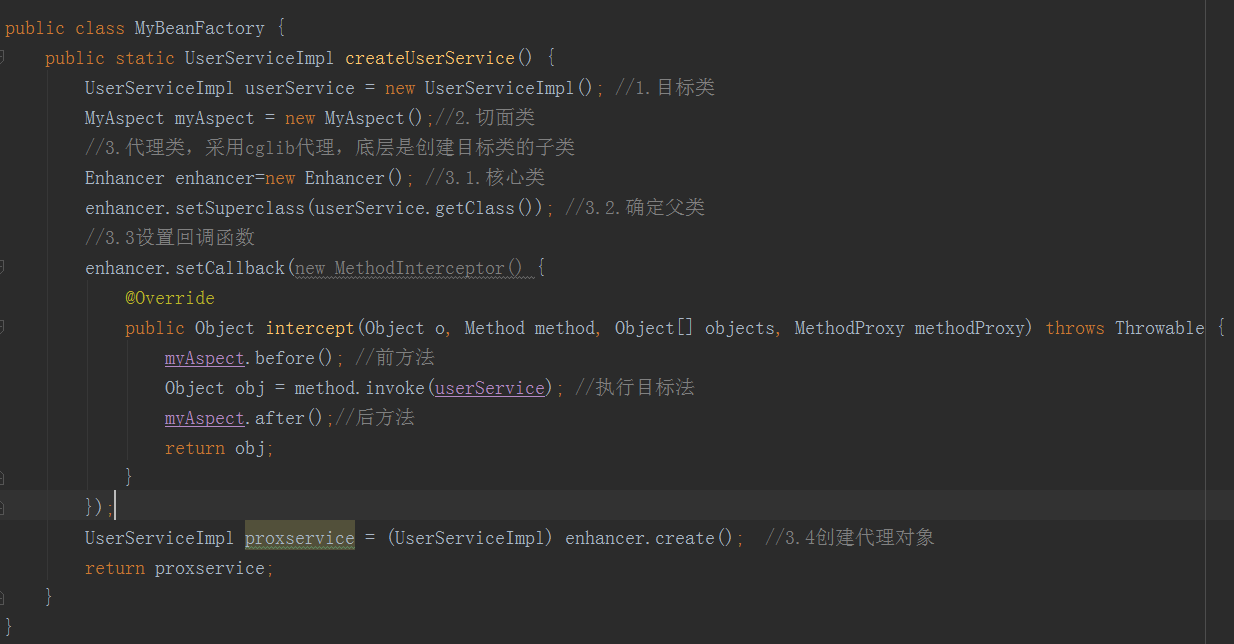
1. cglib代理（只有实现类）
2. 实现类



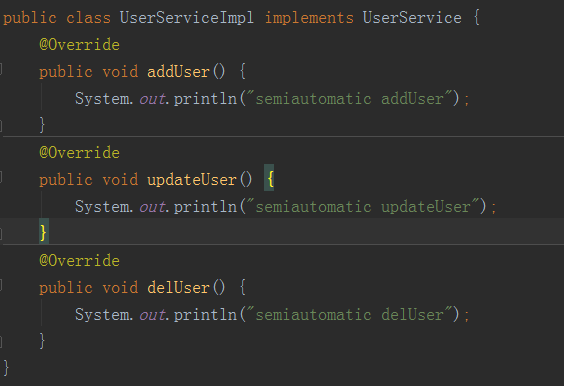
1. 增强类



1. 代理类



1. spring半自动配置代理
2. 目标类（接口和实现类）



注：接口省略

1. 切面类



说明：通知也称为增强，就是增强的方法

1. 配置文件



1. 测试

