

Spring 框架中用到了哪些设计模式？

* **工厂设计模式** : Spring使用工厂模式通过 BeanFactory、ApplicationContext 创建 bean 对象。
* **代理设计模式** : Spring AOP 功能的实现。
* **单例设计模式** : Spring 中的 Bean 默认都是单例的。
* **模板方法模式** : Spring 中 jdbcTemplate、hibernateTemplate 等以 Template 结尾的对数据库操作的类，它们就使用到了模板模式。
* **包装器设计模式** : 我们的项目需要连接多个数据库，而且不同的客户在每次访问中根据需要会去访问不同的数据库。这种模式让我们可以根据客户的需求能够动态切换不同的数据源。
* **观察者模式:** Spring 事件驱动模型就是观察者模式很经典的一个应用。
* **适配器模式** :Spring AOP 的增强或通知(Advice)使用到了适配器模式、spring MVC 中也是用到了适配器模式适配Controller。

Spring AOP

**PointCut【切点】** 其实切点的概念很好理解，你想要去切某个东西之前总得先知道要在哪里切入是吧，切点格式如下：execution(\* com.nuofankj.springdemo.aop.*Service.*(..)) 可以看出来，格式使用了正常表达式来定义那个范围内的类、那些接口会被当成切点，简单明了。

**Advice** Advice行内很多人都定义成了通知，但是我总觉得有点勉强。所谓的Advice其实就是定义了Aop何时被调用，确实有种通知的感觉，何时调用其实也不过以下几种：

* Before 在方法被调用之前调用
* After 在方法完成之后调用
* After-returning 在方法成功执行之后调用
* After-throwing 在方法抛出异常之后调用
* Around 在被通知的方法调用之前和调用之后调用

**JoinPoint【连接点】** JoinPoint连接点，其实很好理解，上面又有通知、又有切点，那和具体业务的连接点又是什么呢？没错，其实就是对应业务的方法对象，因为我们在横切代码中是有可能需要用到具体方法中的具体数据的，而连接点便可以做到这一点

#### @Resource和@Autowired的区别

* @Resource和@Autowired都是做bean的注入时使用
* @Resource不是Spring中的注解，但是Spring支持该注解，而@Autowired是Spring的注解
* @Autowired是按照类型(byType)来装配Bean的，不回去匹配name，默认情况下他要求依赖对象必须存在，如果需允许null，可以设置它的required属性为false，如果想让@Autowired按照名称（byName）来装配，则需要配合@Qualifier一起使用