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 Bean 的加载**

* 实例化BeanFactoryPostProcessor实现类
* 调用BeanFactoryPostProcessor#postProcessBeanFactory
* 实例化BeanPostProcessor实现类
* 调用InstantiationAwareBeanPostProcessor#postProcessBeforeInstantiation
* 实例化Bean
* 调用InstantiationAwareBeanProcessor#postProcessAfterInstantiation
* 调用InstantiationAwareBeanPostProcessor#postProcessPropertyValues为Bean注入属性
* 调用BeanNameAware#setBeanName
* 调用BeanClassLoaderAware#setBeanClassLoader
* 调用BeanFactoryAware#setBeanFactory
* 调用BeanPostProcessor#postProcessBeforeInitialization
* 调用InitializingBean#afterPropertiesSet
* 调用Bean的init-method
* 调用BeanPostProcessor#postProcessAfterInitialization

Bean 的生命周期

* Spring启动，查找并加载需要被Spring管理的bean，进行Bean的实例化
* Bean实例化后对将Bean的引入和值注入到Bean的属性中
* 如果Bean实现了BeanNameAware接口的话，Spring将Bean的Id传递给setBeanName()方法
* 如果Bean实现了BeanFactoryAware接口的话，Spring将调用setBeanFactory()方法，将BeanFactory容器实例传入
* 如果Bean实现了ApplicationContextAware接口的话，Spring将调用Bean的setApplicationContext()方法，将bean所在应用上下文引用传入进来。
* 如果Bean实现了BeanPostProcessor接口，Spring就将调用他们的postProcessBeforeInitialization()方法。
* 如果Bean 实现了InitializingBean接口，Spring将调用他们的afterPropertiesSet()方法。类似的，如果bean使用init-method声明了初始化方法，该方法也会被调用
* 如果Bean 实现了BeanPostProcessor接口，Spring就将调用他们的postProcessAfterInitialization()方法。
* 此时，Bean已经准备就绪，可以被应用程序使用了。他们将一直驻留在应用上下文中，直到应用上下文被销毁。
* 如果bean实现了DisposableBean接口，Spring将调用它的destory()接口方法，同样，如果bean使用了destory-method 声明销毁方法，该方法也会被调用。