# ****Spring 面试题****

## ****什么是Spring框架？****

Spring是一种轻量级框架，旨在提高开发人员的开发效率以及系统的可维护性。

我们一般说的Spring框架就是Spring Framework，它是很多模块的集合，使用这些模块可以很方便地协助我们进行开发。这些模块是核心容器、数据访问/集成、Web、AOP（面向切面编程）、工具、消息和测试模块。

6个特征：

核心技术：依赖注入（DI），AOP，事件（Events），资源，i18n，验证，数据绑定，类型转换，SpEL。

测试：模拟对象，TestContext框架，Spring MVC测试，WebTestClient。

数据访问：事务，DAO支持，JDBC，ORM，编组XML。

Web支持：Spring MVC和Spring WebFlux Web框架。

集成：远程处理，JMS，JCA，JMX，电子邮件，任务，调度，缓存。

语言：Kotlin，Groovy，动态语言。

## ****列举一些重要的Spring模块？****

Spring Core：基础，可以说Spring其他所有的功能都依赖于该类库。主要提供IOC和DI功能。

Spring Aspects：该模块为与AspectJ的集成提供支持。

Spring AOP：提供面向方面的编程实现。

Spring JDBC：Java数据库连接。

Spring JMS：Java消息服务。

Spring ORM：用于支持Hibernate等ORM工具。

Spring Web：为创建Web应用程序提供支持。

Spring Test：提供了对JUnit和TestNG测试的支持

## ****Spring IOC和AOP的理解****

### ****IOC****

IOC是基于反射机制来实现的。

IOC（Inversion Of Controll，控制反转）是一种设计思想，就是将原本在程序中手动创建对象的控制权，交由给Spring框架来管理。

将对象之间的相互依赖关系交给IOC容器来管理，并由IOC容器完成对象的注入。这样可以很大程度上简化应用的开发，把应用从复杂的依赖关系中解放出来。

### ****AOP****

AOP（Aspect-Oriented Programming，面向切面编程）能够将那些与业务无关，却为业务模块所共同调用的逻辑或责任（例如事务处理、日志管理、权限控制等）封装起来，便于减少系统的重复代码，降低模块间的耦合度，并有利于未来的可扩展性和可维护性。

Spring AOP是基于动态代理的，如果要代理的对象实现了某个接口，那么Spring AOP就会使用JDK动态代理去创建代理对象；

## ****Spring中的bean的作用域有哪些？****

1.singleton：唯一bean实例，Spring中的bean默认都是单例的。

2.prototype：每次请求都会创建一个新的bean实例。

3.request：每一次HTTP请求都会产生一个新的bean，该bean仅在当前HTTP request内有效。

4.session：每一次HTTP请求都会产生一个新的bean，该bean仅在当前HTTP session内有效。

5.global-session：全局session作用域，仅仅在基于Portlet的Web应用中才有意义，Spring5中已经没有了。Portlet是能够生成语义代码（例如HTML）片段的小型Java Web插件。它们基于Portlet容器，可以像Servlet一样处理HTTP请求。但是与Servlet不同，每个Portlet都有不同的会话。

## ****Spring框架中用到了哪些设计模式****

1.工厂设计模式：Spring使用工厂模式通过BeanFactory和ApplicationContext创建bean对象。

2.代理设计模式：Spring AOP功能的实现。

3.单例设计模式：Spring中的bean默认都是单例的。

4.模板方法模式：Spring中的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。

5.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。

6.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。

7.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。

## ****@Component和@Bean的区别是什么****

1.作用对象不同。@Component注解作用于类，而@Bean注解作用于方法。

2.@Component注解通常是通过类路径扫描来自动侦测以及自动装配到Spring容器中（我们可以使用@ComponentScan注解定义要扫描的路径）。@Bean注解通常是在标有该注解的方法中定义产生这个bean，告诉Spring这是某个类的实例，当我需要用它的时候还给我。

3.@Bean注解比@Component注解的自定义性更强，而且很多地方只能通过@Bean注解来注册bean。比如当引用第三方库的类需要装配到Spring容器的时候，就只能通过@Bean注解来实现。

**将一个类声明为Spring的bean的注解有哪些？**

我们一般使用@Autowired注解去自动装配bean。而想要把一个类标识为可以用@Autowired注解自动装配的bean，可以采用以下的注解实现：

1.@Component注解。通用的注解，可标注任意类为Spring组件。如果一个Bean不知道属于哪一个层，可以使用@Component注解标注。

2.@Repository注解。对应持久层，即Dao层，主要用于数据库相关操作。

3.@Service注解。对应服务层，即Service层，主要涉及一些复杂的逻辑，需要用到Dao层（注入）。

4.@Controller注解。对应Spring MVC的控制层，即Controller层，主要用于接受用户请求并调用Service层的方法返回数据给前端页面。

## ****Spring事务管理的方式有几种？****

1.编程式事务：在代码中硬编码（不推荐使用）。

2.声明式事务：在配置文件中配置（推荐使用），分为基于XML的声明式事务和基于注解的声明式事务。