# Spring

### 1、对Spring的理解

Spring 是一个开源的轻量级的Java开发框架，以 Ioc（Inverse Of control：反转控制）和AOP（Aspect Oriented Programming: 面向切面编程） 为核心，提供了展现层Spring MVC和持久层Spring JDBC以及业务层事务管理等众多企业级应用技术，其还整合众多开源的第三方框架，逐渐成为最受欢迎的Java EE企业应用框架

### 2、Spring框架特征

轻量高效：Spring框架本身体量小

IOC控制翻转：见下

AOP面向切面编程：见下

声明式事务：Spring可以使用注解的方式声明事务，省去了很多繁琐的代码，提高开发效率

容器：Spring管理每个应用对象的的配置和生命周期，在这个层面上它是一个容器，配置每个bean如何被创建

框架：Spring集成了各种框架，并降低了很多框架的使用复杂度，提供了对各种框架的直接支持

### 3、Spring IOC理解

IOC控制翻转是一种设计思想，它将原本是在程序运行时才创建并管理的对象交给Spring框架创建和管理，控制权由应用代码转移到了 Spring 容器，控制权发生了反转。

让开发者能够专注于业务逻辑层的开发，大大降低了模块之间的耦合度，提高开发效率和代码可维护性

### 4、依赖注入（DI）理解，依赖注入的方式

依赖注入是一种实现控制反转的设计模式，它指将对象的创建和依赖关系的分析从业务逻辑中分离开来，并将它们集中管理。核心思想是将依赖关系从代码中剔除出来，使得代码更加灵活，提高系统可扩展性和可维护性

依赖指的是一个对象需要使用的外部资源，比如其他对象、服务、数据库连接等

依赖注入主要有三种方式，set方法注入，构造器注入，自动注入 详情见：Spring积累(3)：依赖注入3种方式\_依赖注入的三种实现方式-CSDN博客

### 5、什么是AOP，AOP的应用场景有哪些？

Spring AOP 第三节\_哔哩哔哩\_bilibili

AOP就是面向切面编程，它是一种编程思想和方法，它将影响多个对象的公共行为抽离并封装，实现了与业务逻辑的解耦，减少重复代码，增强代码可维护性，提高开发效率

AOP中的概念：

目标：指被代理对象

切面：相当于是切入点和通知的集合，定义了切面的全部内涵

连接点：指被增强的方法

切入点：连接点的集合

通知：指拦截到连接点后要做的增强

织入：指将切入点增强的过程

AOP的应用场景：

记录操作日志

缓存处理

Spring内置事务处理

异常处理

### 6、AOP的原理是什么，静态代理和动态代理的区别

AOP是通过JDK动态代理和CGLIB动态代理实现的

如果被代理类实现了接口，就会通过JDK动态代理，JDK动态代理是通过Java的反射机制实现的，代理类必须实现InvocationHandler接口，然后通过Proxy类的new ProxyInstance方法创建代理对象，然后通过代理对象调用被代理类的方法，并在此实现方法增强

如果代理类没有实现接口，就会通过CGLIB动态代理，CGLIB是通过创建当前类子类的代理对象，然后通过这个代理对象去继承父类方法，并在此做增强

静态代理去动态代理的区别：

静态代理在编译期间就已经确定了代理类和代理方法，而动态代理在运行时才确定

静态代理需要为每一个被代理类写一个具体的代理类，而动态代理只需要写一个代理类

静态代理是通过实现接口或继承父类的方式，而动态代理是通过反射

### 7、Spring事务失效的场景

在同一个类中，一个方法调用一个事务方法（没有用到代理类）

@Transactional标记的方法不是public（对外包不可见，无法使用代理）

@Transactional标记的方法被final或static修饰（无法重写，不能动态代理）

数据库表不支持事务，如MyIsam

异常捕获处理而不是抛出

抛出异常与rollbackfor指定的异常不一致，rollbackfor默认抛出RuntimeException

Spring的传播行为导致事务失效，比如: PROPAGATION\_NEVER、PROPAGATION\_NOT\_SUPPORTED

### 8、Spring中的Bean是线程安全的吗？

Spring中的Bean不是线程安全的，如果Bean中有可以修改的成员变量，那么就会有线程安全问题，可以通过加锁或修改@Scope的值为多例解决