# Spring

## 什么是Spring？

Java应用开发框架

## Spring的优点

1. 方便解耦，简化开发
2. 支持AOP
3. 支持事务
4. 方便程序测试

## SpringIOC

### 什么是SpringIOC(控制反转)？

IOC就是控制反转,通过XML配置文件或者注解方式让第三方容器去产生或获取特定对象的方式

### 什么是DI(依赖注入)？

在使用spring框架创建Bean对象时，动态地将其所依赖的对象注入Bean组件中

#### 依赖注入的方式

三种,构造方法,,setter方法和注解方式

## SpringBean

### 什么是SpringBean?

Bean是一个实例化对象,通过Spring框架的注解方式或XML配置文件创建  
Bean的作用域

**Singleton:** 默认的作用域,在Bean容器中只有一个Bean实例

Prototype: Spring每次获取prototype定义的Bean,容器都将创建一个新的Bean实例

**request,::** 在每一次HTTP请求中容器都将返回一个Bean实例,不同的HTTP请求返回不同的Bean实例

**session:** 在一个HTTP Session中，容器将返回同一个Bean实例

**,application::** 为每个ServletContext对象创建一个实例,即同一个应用共享一个Bean实例

**,websocker.:** 为每个WebSocket对象创建一个Bean实例

### Bean的生命周期

实例化->初始化->使用->销毁

## SpringAOP

### 什么是SpringAOP?

面向切面编程

### Bean的装配方式

no：不进行自动装配，手动设置Bean的依赖关系。

byName：根据Bean的名字进行自动装配。

byType：根据Bean的类型进行自动装配。

constructor：类似于byType，不过是应用于构造器的参数，如果正好有一个Bean与构造器的参数类型相同则可以自动装配，否则会导致错误。

autodetect：如果有默认的构造器，则通过constructor的方式进行自动装配，否则使用byType的方式进行自动装配。

### SpringAOP的作用

将应用中需要被重复使用的功能以切面来表示,大大减少了代码量与代码繁琐度.

### SpringAOP的原理

要分为两大类：一是采用动态代理技术，利用截取消息的方式，对该消息进行装饰，以取代原有对象行为的执行；二是采用静态织入的方式，引用特点的语法创建“方面”，从而使可以在编译期间织入有关“方面”的代码。

### SpringAOP的应用场景

AOP的使用场景：权限，缓存，日志记录，事务，性能优化，异常处理。

### SpringAOP的术语

#### 切面

切面就是横切的封装类。也可以理解为共用方法类

#### 连接点

连接点就是程序运行时可以插入切面的点

#### 切入点

切入点就是确定要插入切面的连接点

#### 通知

通知就是在定义好的切入点处要执行的程序代码

#### 引入

允许我们向现有的类添加新方法属性。这不就是把切面（也就是新方法属性：通知定义的）用到目标类中吗

#### 目标对象

目标对象就是被织入切面的对象

#### 代理对象

代理对象就是为了实现织入切面到目标对象的功能的对象

#### 织入

织入就是将切面代码插入到目标对象,从而生成代理对象的过程。

##### 分类

##### 编译器织入

需要有特殊的java编译器

##### 类装载期织入

需要有特殊的类装载器

##### 动态代理织入

在运行期为目标类添加通知生成子类的方式。

AO/P默认使用这种方式

## 注解

### 声明Bean的注解

@Component 组件，没有明确的角色

@Service 在业务逻辑层使用（service层）

@Repository 在数据访问层使用（dao层）

@Controller 在展现层使用，控制器的声明（C）

### 注入Bean的注解

@Autowired：

@Resource：

### 事务注解

### Web注解

’

## 问题

**1.Spring支持的事务管理类型有哪些？你在项目中使用哪种方式？**

编程式事务管理和声明式事务管理。

声明式事务管理[@Transactional](https://www.nowcoder.com/profile/986587404)(可以保证方法或者类中所有方法具有原子性等

)

**2.持久层设计要考虑的问题有哪些？你用过的持久层框架有哪些？**

(所谓”持久”就是将数据保存到可掉电式存储设备中以便今后使用，简单的说，就是将内存中的数据保存到关系型数据库、文件系统、消息队列等提供持久化支持的设备中。持久层就是系统中专注于实现数据持久化的相对独立的层面。)

持久性设计考虑的问题:

数据存储逻辑的分离，提供抽象化的数据访问接口。

数据访问底层实现的分离，可以在不修改代码的情况下切换底层实现。

资源管理和调度的分离，在数据访问层实现统一的资源调度（如缓存机制）。

数据抽象，提供更面向对象的数据操作。

只有Mybatis

**3.Spring里面注解用过没有？autowired 和resource区别？**

@Autowired是Spring注解,@Resource是JDK注解

@Autowired通过类型装配,@Resource默认通过参数名,然后才是类型