## 6、请解释 Spring Bean 的生命周期？

Servlet 的生命周期：实例化，初始 init，接收请求 service，销毁destroy；

Spring 上下文中的 Bean 生命周期：

1实例化 Bean。

2设置对象属性。

3处理 Aware 接口。

4 BeanPostProcessor。

5 InitializingBean 与 init-method。

6 实现了 BeanPostProcessor 接口，将会调postProcessAfterInitialization方法。

7 DisposableBean：

8 destroy-method：

## 7、解释 Spring 支持的几种 bean 的作用域。

Spring 容器中的 bean 可以分为 5 个范围：

（1）singleton：默认，每个容器中只有一个 bean 的实例，单例的模式由BeanFactory 自身来维护。

（2）prototype：为每一个 bean 请求提供一个实例。

（3）request：为每一个网络请求创建一个实例，在请求完成以后，bean 会失效并被垃圾回收器回收。

（4）session：与 request 范围类似，确保每个 session 中有一个 bean 的实例，在 session 过期后，bean 会随之失效。

（5）global-session：全局作用域，global-session 和 Portlet 应用相关。

## 8、使用注解之前要开启自动扫描功能

其中 base-package 为需要扫描的包(含子包)。

Spring 容器中的 Bean：

@Scope 注解，作用域 。

@Lazy(true) 表示延迟初始化 。

@Service 用于标注业务层组件。

@Controller 用于标注控制层组件。

@Repository 用于标注数据访问组件。

@Component 泛指组件，当组件不好归类的时候，我们可以使用这个注解进行标注。

@Scope 用于指定 scope 作用域的。

@PostConstruct 用于指定初始化方法。

@PreDestory 用于指定销毁方法。

@Resource 默认按名称装配，当找不到与名称匹配的 bean 才会按类型装配。

@DependsOn：定义 Bean 初始化及销毁时的顺序 。

@Primary：自动装配时当出现多个 Bean 候选者时，被注解为@Primary 的 Bean 将作为首选者，否则将抛出异常 。

@Autowired 默认按类型装配，如果我们想使用按名称装配，可以结合@Qualifier 注解一起使用。

如下：

@Autowired @Qualifier("personDaoBean") 存在多个实例配合使用 。

## 9、Spring 框架中的单例 Beans 是线程安全的么？

Spring 框架并没有对单例 bean 进行任何多线程的封装处理。

实际上，大部分的 Spring bean 并没有可变的状态，所以在某种程度上说Spring 的单例 bean 是线程安全的。如果你的 bean 有多种状态的话，就需要自行保证 线程安全。

## 10、Spring 如何处理线程并发问题？

ThreadLocal 和线程同步机制都是为了解决多线程中相同变量的访问冲突问题。

ThreadLocal 采用了“空间换时间”的方式，会为每一个线程提供一个独立的变量副本，从而隔离了多个线程对数据的访问冲突。

ThreadLocal 提供了线程安全的共享对象，在编写多线程代码时，可以把不安全的变量封装进 ThreadLocal。

11、Spring 基于 xml 注入 bean 的几种方式：

1Set方法注入

2构造器注入：①通过 index 设置参数的位置；②通过 type 设置参数类型

3静态工厂注入

4实例工厂

## 12、Spring 的自动装配

在 spring 中，对象无需自己查找或创建与其关联的其他对象，由容器负责把需要相

互协作的对象引用赋予各个对象，使用 autowire 来配置自动装载模式。

在 Spring 框架 xml 配置中共有 5 种自动装配：

（1）no：默认的方式是不进行自动装配的，通过手工设置 ref 属性来进行装配 bean。

（2）byName：通过 bean 的名称进行自动装配

（3）byType：通过参数的数据类型进行自动装配。

（4）constructor：利用构造函数进行装配，并且构造函数的参数通过 byType 进行装配。

（5）autodetect：自动探测，如果有构造方法，通过 construct 的方式自动装配，否则使用 byType 的方式自动装配。

## 13、@Autowired 和@Resource 之间的区别

1@Autowired 默认是按照类型装配注入的，默认情况下它要求依赖对象必须存在（可以设置它 required 属性为 false）。

2 @Resource 默认是按照名称来装配注入的，只有当找不到与名称匹配的 bean才会按照类型来装配注入。

## 14、Spring 框架中都用到了哪些设计模式？

1工厂模式

2单例模式

3代理模式

4模板方法

5观察者模式

**15、Spring 事务的实现方式和实现原理：**

Spring 事务的本质其实就是数据库对事务的支持，没有数据库的事务支持，spring

是无法提供事务功能的。

**（1）Spring 事务的种类：**

spring 支持编程式事务管理和声明式事务管理两种方式：

①编程式事务管理使用 TransactionTemplate。

②声明式事务管理建立在 AOP 之上的，其本质是通过 AOP 功能，对方法前后进行拦截，将事务处理的功能编织到拦截的方法中，也就是在目标方法开始之前加入一个事务，在执行完目标方法之后根据执行情况提交或者回滚事务。

**（2）spring 的事务传播行为：**

spring 事务的传播行为说的是，当多个事务同时存在的时候，spring 如何处理这些

事务的行为。

① PROPAGATION\_REQUIRED：如果当前没有事务，就创建一个新事务，如果当前存在事务，就加入该事务，该设置是最常用的设置

② PROPAGATION\_SUPPORTS：支持当前事务，如果当前存在事务，就加入该事务，如果当前不存在事务，就以非事务执行。‘

③ PROPAGATION\_MANDATORY：支持当前事务，如果当前存在事务，就加入该事务，如果当前不存在事务，就抛出异常。

④ PROPAGATION\_REQUIRES\_NEW：创建新事务，无论当前存不存在事务，都创建新事务。

⑤ PROPAGATION\_NOT\_SUPPORTED：以非事务方式执行操作，如果当前存在事务，就把当前事务挂起。

⑥ PROPAGATION\_NEVER：以非事务方式执行，如果当前存在事务，则抛出异常。

⑦ PROPAGATION\_NESTED：如果当前存在事务，则在嵌套事务内执行。如果当

前没有事务，则按 REQUIRED 属性执行。

**（3）Spring 中的隔离级别：**

① ISOLATION\_DEFAULT：这是个 PlatfromTransactionManager 默认的隔离级别，使用数据库默认的事务隔离级别。

② ISOLATION\_READ\_UNCOMMITTED：读未提交

③ ISOLATION\_READ\_COMMITTED：读已提交

④ ISOLATION\_REPEATABLE\_READ：可重复读

⑤ ISOLATION\_SERIALIZABLE：一个事务在执行的过程中完全看不到其他事务对

数据库所做的更新。