6、请解释 Spring Bean 的生命周期？

首先说一下Servlet的生命周期；实例化，初始化init，接收请求service，销毁destroy；

Spring上下文的Bean生命周期也类似，如下：

1. 实例化Bean
2. 设置对象属性（依赖注入）
3. 处理Aware接口
4. BeanPostProcessor
5. InitializingBean与init-method
6. 如果这个Bean实现了BeanPostProcessor接口，将会调用postProcessAfterInitialization（Object obj,String s）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。
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。

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）

@Configuration把一个类作为IOC容器

@Scope 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

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。

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。

10 、Spring 如何处理线程并发问题？

在一般情况下，只有无状态的Bean才可以在多线程环境下共享，，

绝大部分Bean都可以声明为singleton作用域，因为Spring对一些Bean中非线程安全状态采用ThreadLocal进行处理，解决线程安全问题。

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。

11 、Spring 基于 xml 注入 bean 的几种方式：

1. set方法注入；
2. 构造器注入；
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。

Spring框架xml配置中共有五种自动装配：

1. no：默认是通过手工设置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。
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，默认情况下它要求依赖对象必须存在。

（2）@Resource默认是按照名称装配注入，找不到与名称匹配的bean才会按照类型装配注入。

14 、Spring 框架中都用到了哪些设计模式？

1. 工厂模式：BeanFactory就是简单工厂模式的体现，用来创建对象的实例；
2. 单例模式：Bean默认单例模式；
3. 代理模式：spring的AOP功能用到了JDK动态代理和CGLIB字节码生成技术；
4. 模板方法：用来解决代码重复的问题；
5. 观察者模式：定义对象键一种一对多的依赖关系。

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物的支持，没有数据库事物支持，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俩种方式。

1. 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. PROPAFATION\_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. Spring中的隔离级别：
4. ISOLATION\_DEFAULT:默认的隔离级别
5. ISOLATION\_READ\_UNCOMMITTED:读未提交
6. ISOLATION\_READ\_COMMITTED:读已提交
7. ISOLATION\_REPEATABLE\_READ:可重复读
8. ISOLATION\_SERIALIZABLE：一个事物在执行的过程完全看不到其他事务对数据库所做的更新。