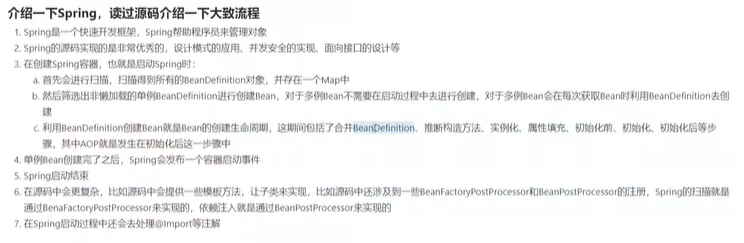
1. 一个回调接口有多种回调逻辑，在项目启动时需要将回调接口的实现类都放到一个list中然后回调每个实现类的具体方法。在Springboot中就可以通过@Autowired来将回调接口的所有实现类进行注入。
2. SpringBoot默认包扫描机制：从启动类所在包开始，扫描当前包及其子级别包下的所有文件。
3. 谈谈你理解的Spring是什么： spring是一个轻量级的开源容器框架；spring是为了解决企业级应用开发的业务逻辑层和其他各层对象和对象之间的耦合问题；spring是一个IOC和AOP的容器框架。（IOC：控制反转，AOP:面向切面编程（对程序进行无缝的增强），容器：包含并管理应用对象的生命周期），（spring是一个生态，可以构建java应用所需的一切基础设施；通常Spring指的就是Spring Framework）



1. Bean的生命周期

单例bean是不会重复创建的，通过map<beanName, Bean对象>去查有没有，没有就执行bean的创建过程将生成的对象放到map里面去，有的话就直接取用，map就叫单例池

Bean的创建过程图解

通过@Scope注解可以修改bean的单例或者多例（原型），如果是多例bean则不用往map里面放值



初始化前：对于要在bean创建的时候去执行某些方法，来给bean中特定字段赋值，可以在方法上加@PostConstruct注解，去执行该方法（发生在初始化前阶段）

初始化：实现InitializingBean接口，重写afterPropertiesSet()方法，在初始化阶段执行该方法，没有通过反射查找要执行的方法，直接调用，效率高（优点），但是只能执行afterPropertiesSet()这一个方法（缺点），初始化前可以有好多个方法

Bean的实例化和Bean的初始化：通过无参方法得到一个bean对象就是实例化，初始化就是去执行afterPropertiesSet方法

初始化后(AOP)：会生成代理对象，之后往map里面放的就会是代理对象

在spring创建对象，调用构造方法时，会推断要调用的构造方法，只要有无参不管有没有有参就调用无参构造方法除非加@Autowired注解会调用加了注解的构造方法，如果只有一个有参构造方法那么调用有参构造方法，如果有多个有参构造方法没有无参构造，那么报错除非加@Autowired注解会调用加了注解的构造方法。

对于传入有参构造方法的参数来源问题：如果为单例，会从map（单例池里面取），然后传入，如果在map里找不到，那么报错；如果为多例那么就会创建bean传入，如果无法创建bean那么报错，通过byType的方式去找要传入的bean，如果bytype找到多个，那再通过byname去找，如果byname找不到那么报错

单例bean也算单例模式（在一个jvm中一个类只能有一个对象，是有范围的），单例bean的单例模式范围非常小仅限于名字，是指按照名字保证的单例，既一个名字只有一个bean

SpringAOP：基于JDK的动态代理和CGLAB的继承代理，面向切面编程，通过预编译方式和运行期动态代理实现在不修改源代码的情况下给程序动态统一添加功能的一种技术。AOP的编程思想就是把很多类对象中的横切问题点，从业务逻辑中分离出来，从而达到解耦的目的，增加代码的重用性，提高开发效率。开启AspectJ的AOP需要在切面类加上@EnableAspectJAutoProxy注解。此时通过spring获取到的bean是切面类的代理对象，通过spring获取到的代理对象里面的注入属性为null，因为spring并没有针对代理对象的依赖注入。但是调用的时候里面的属性是有值的因为是代理模式，代理对象里面有target字段类型为切面类，该字段为原来的切面类对象是被spring依赖注入的，所以属性不为空，参考下图：

