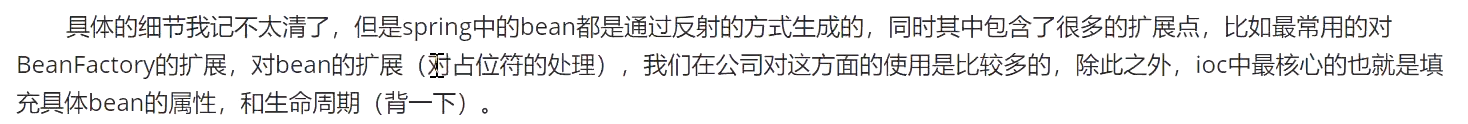
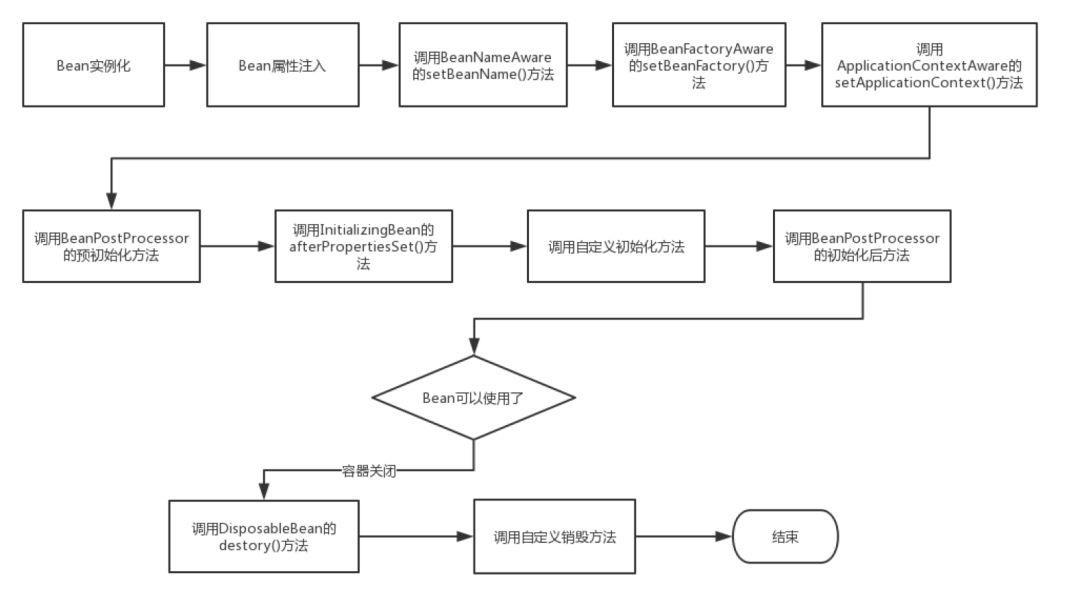
Spring面试问题汇总：

1：springIOC的理解，原理及实现



<https://zhuanlan.zhihu.com/p/29344811>

2：bean的生命周期





Spring启动，查找并加载需要被Spring管理的bean，进行Bean的实例化

Bean实例化后对将Bean的引入和值注入到Bean的属性中

如果Bean实现了BeanNameAware接口的话，Spring将Bean的Id传递给setBeanName()方法

如果Bean实现了BeanFactoryAware接口的话，Spring将调用setBeanFactory()方法，将BeanFactory容器实例传入

如果Bean实现了ApplicationContextAware接口的话，Spring将调用Bean的setApplicationContext()方法，将bean所在应用上下文引用传入进来。

如果Bean实现了BeanPostProcessor接口，Spring就将调用他们的postProcessBeforeInitialization()方法。

如果Bean 实现了InitializingBean接口，Spring将调用他们的afterPropertiesSet()方法。类似的，如果bean使用init-method声明了初始化方法，该方法也会被调用

如果Bean 实现了BeanPostProcessor接口，Spring就将调用他们的postProcessAfterInitialization()方法。

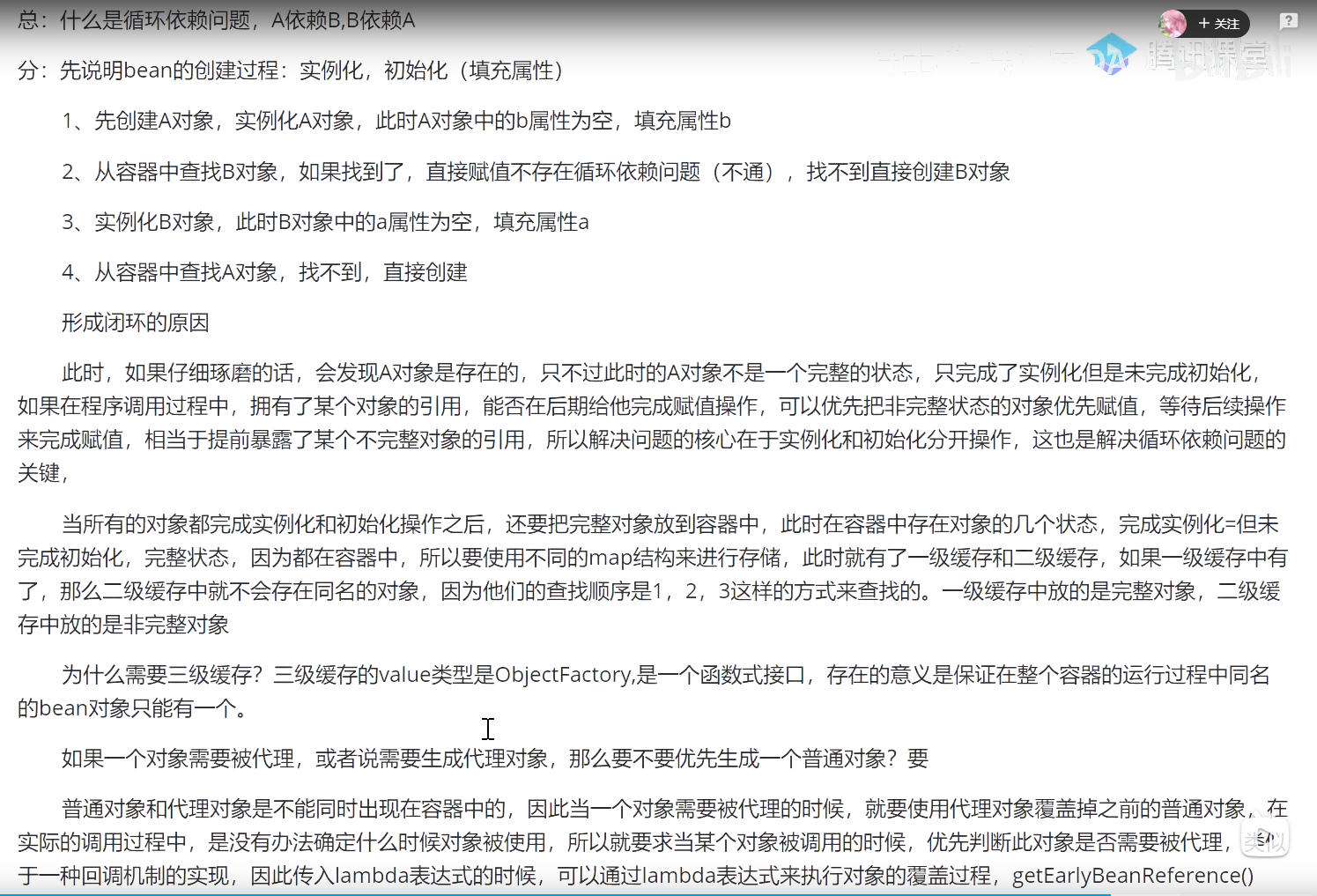
此时，Bean已经准备就绪，可以被应用程序使用了。他们将一直驻留在应用上下文中，直到应用上下文被销毁。

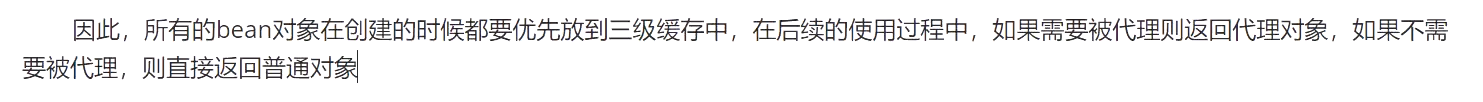
如果bean实现了DisposableBean接口，Spring将调用它的destory()接口方法，同样，如果bean使用了destory-method 声明销毁方法，该方法也会被调用。



3：spring是如何解决循环依赖的?

详情见问题闲记.word

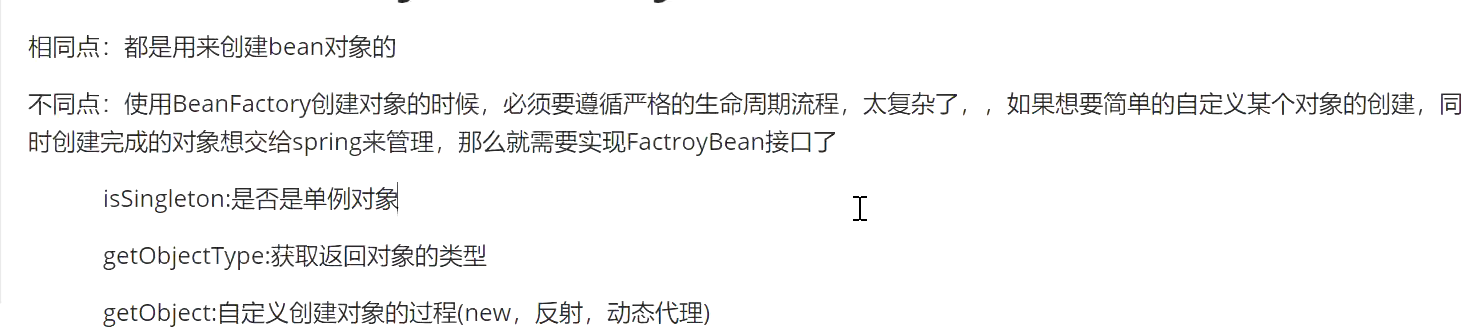




缓存的放置时间和删除时间:



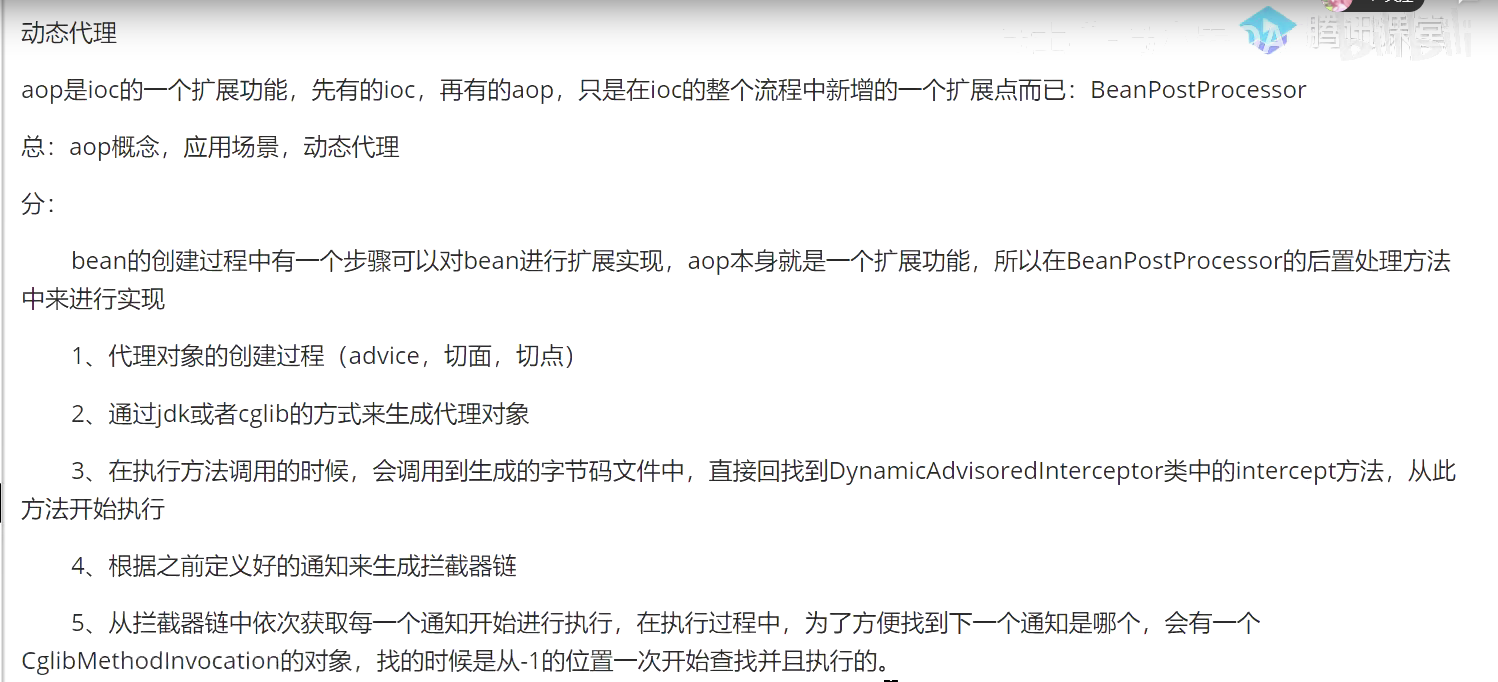
4：BeanFactory和FactoryBean有什么区别？



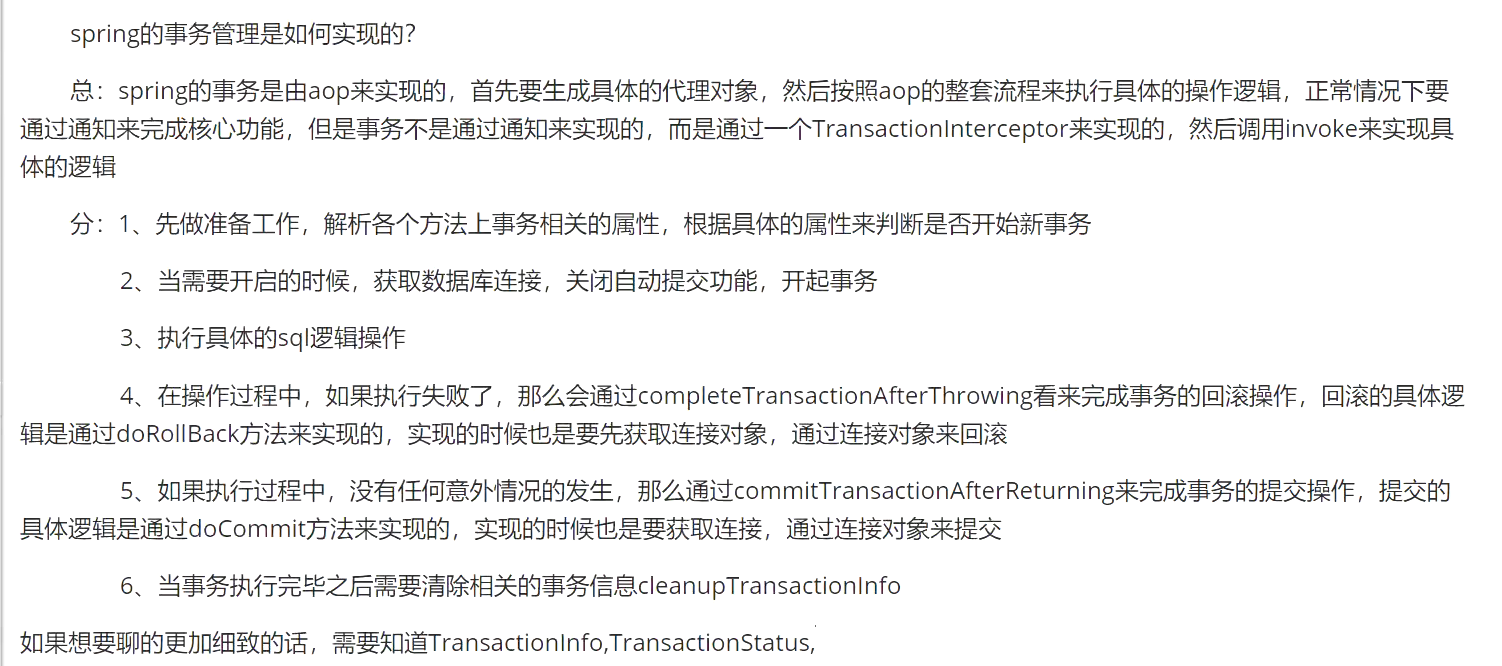
5：spring中用到的设计模式：



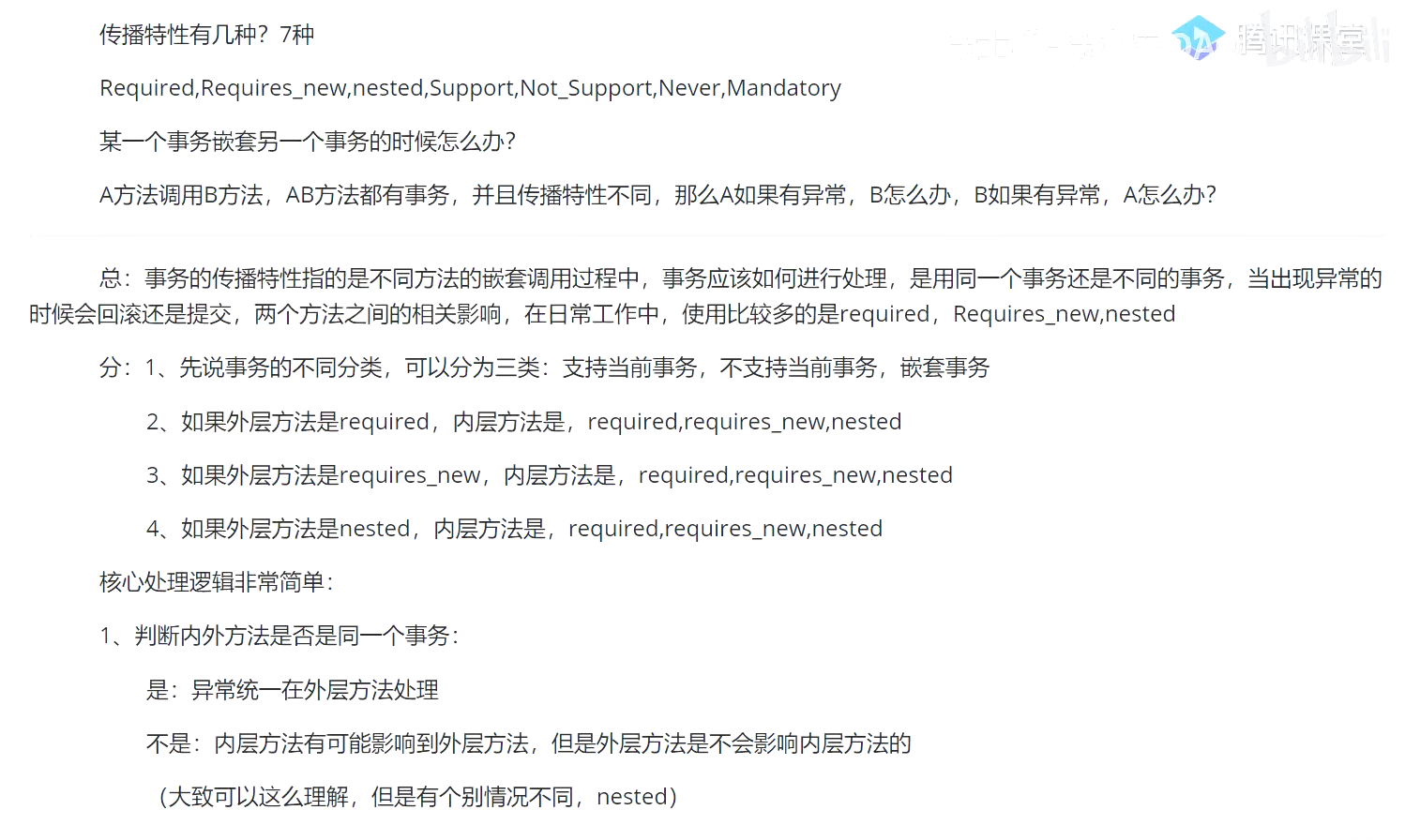
**6：spring的aop的底层实现原理**？

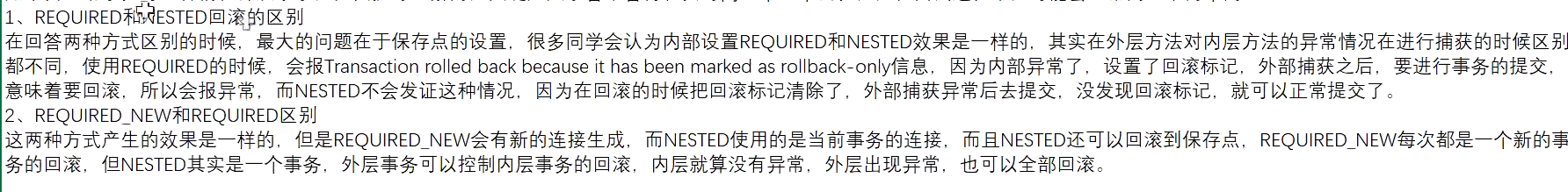


7：spring的事务是如何回滚的？



8：谈一下spring的事务传播？





详解扩展：

#### factorybean接口的作用是什么？

spring提供的一个接口，通过实现这个接口可以定制实例化bean。

 作用1：可以让我们自定义Bean的创建过程，更加灵活。Spring 中有两种类型的Bean，一种是普通Bean，另一种是工厂Bean 即 FactoryBean。FactoryBean跟普通Bean不同，其返回的对象不是指定类的一个实例，而是该FactoryBean的getObject方法所返回的对象。

 作用2：整体上简化了XML配置等。

#### cglib和jdk动态代理的机制是什么？

<https://blog.csdn.net/shallynever/article/details/103351299>

JDK动态代理整个实现过程其实很简单，包括三个步骤：

定义一个接口IHelloService，大家都知道Jdk的动态代理是基于接口，这个就是那个接口。

编写一个实现该接口的类HelloServiceImpl，这个就是目标对象，也就是被代理的对象类。

最后编写一个实现InvocationHandler接口的类JdkDynamicProxy，代理类的方法调用都会被转发到该类的invoke()方法。

JDK和CGLib动态代理区别

通过实现InvocationHandler接口创建自己的调用处理器；

1、jdk动态代理：

通过为Proxy类指定ClassLoader对象和一组interface来创建动态代理；

通过反射机制获取动态代理类的构造函数，其唯一参数类型就是调用处理器接口类型；

通过构造函数创建动态代理类实例，构造时调用处理器对象作为参数参入；

JDK动态代理是面向接口的代理模式，如果被代理目标没有接口那么Spring也无能为力，Spring通过Java的反射机制生产被代理接口的新的匿名实现类，重写了其中AOP的增强方法。

2、CGLib动态代理：

利用ASM开源包，对代理对象类的class文件加载进来，通过修改其字节码生成子类来处理。

3、两者对比：

JDK动态代理是面向接口的。

CGLib动态代理是通过字节码底层继承要代理类来实现，因此如果被代理类被final关键字所修饰，会失败。

4、使用注意：

如果要被代理的对象是个实现类，那么Spring会使用JDK动态代理来完成操作（Spirng默认采用JDK动态代理实现机制）；

如果要被代理的对象不是个实现类那么，Spring会强制使用CGLib来实现动态代理。