1. JDK动态代理核心：Proxy、InvocationHandler。
2. InvocationHandler：每一个Proxy都与一个InvocationHandler关联，代理实例在调用方法时，方法调用会被分派到InvocationHandler的invoke方法。
3. Proxy.newProxyInstance：

接受一个类加载器ClassLoader，并由该代理器来加载代理类。

接受一个接口数组interfaces，表明要给代理对象提供的接口。

接受一个InvocationHandler，代理对象的方法调用由它实现。

1. Pointcut：提供切点匹配器的接口。
2. AspectJExpressionPointcut：由AspectJ提供切点匹配功能。
3. AdvisedSupport：提供通知功能的支持。包含被代理目标对象、方法拦截器、方法匹配器3个属性。
4. 动态代理实现方式：JDK(默认)、Cglib(配置中开启)。
5. AOP在实现动态代理的时候，使用AspectJ进行切点匹配。
6. 在创建bean且还未开始实例化时，会首先对Bean进行AOP处理，若处理过程中产生了Bean实例，则执行后置处理并返回。经过动态代理的Bean并不会被存进SingletonObjects的缓存中。**(存疑)**
7. AOP处理器的接口：InstantiationAwareBeanPostProcessor，该接口继承自BeanPostProcessor，是初始化前后置处理器的特化。