

Aop的术语：

切面（Aspect）：切面是你要实现的交叉功能。它是应用系统模块化的一个切面或领域。例如：日志记录，可以建立一个日志记录切面，使用AOP应用到系统

连接点（Joinpoint）：是应用程序执行过程中插入切面的地点；这个地方可以是方法调用，异常抛出等。切面代码在这个地方插入到你的应用的流程中，添加新的行为

通知（Advice）：通知切面的实际实现。它通知应用系统新的行为。

切入点（Pointcut）：定义了通知应该应用在哪些连接点。通知可以应用到AOP框架支持的任何连接点，但是并不希望把所有切面应用到所有可能的连接点上。切入点让你指定通知应用到什么地方。通常通过指定类名和方法名，或者匹配类名和方法名式样的正则表达式来指定切入点

目标对象（Target）：是被通知的对象。它既可以是你编写的类也可以是你需要添加定制行为

的第三方类；目标对象只要全身心地关注主要业务

代理（Proxy）：代理是将通知应用到目标对象后创建的对象。对于客户对象来说，目标对象和代理对象是一样的

织入（Weaving）：是将切面应用到目标对象从而创建一个新的代理对象的过程；织入发生在目标对象声明周期上的多个点：

编译期

类装载期

运行期

Spring的aop实现

用Java编写Spring通知（AspectJ需要特定的语法来编写切面和定义切入点）

Spring运行时通知对象：代理Bean在第一次被应用系统需要的时候才被创建；如果使用ApplicationContext，那么当BeanFactory载入所有Bean的时候被创建

Spring有两种代理创建方式：

如果目标对象实现了特定接口，Spring使用反射机制来创建代理对象

如果目标对象没有实现任何接口，Spring使用CGLIB库生成目标对象的子类

创建通知，就是在编写实现交叉功能的代码

Spring的连接点模型是建立在方法拦截上；所以通知可以织入到方法调用周围的各个地方

因为Spring可以在方法执行的多个地方织入通知，所以有多种通知类型

Spring中的通知类型

