

AOP介绍 (Aspect Oriented Programming)

一般配置AOP的代码为公共代码，例如工具类

一、概念

AOP (Aspect Oriented Programming)：面向切面编程。

面向切面编程（也叫面向切面编程），是目前软件开发中的一个热点，也是Spring框架中的一个重要内容。利用AOP可以对业务逻辑的各个部分进行隔离，从而使得业务逻辑各部分之间的耦合度降低，提高程序的可重用性，同时提高了开发的效率。

二、用途

日志记录，性能统计，安全控制，权限管理，事务处理，异常处理，资源池管理。

三、AOP 的实现方式

使用动态代理技术

四、AOP的专业术语

1.通知 (Advice)

就是你想要的功能，也就是上面说的 安全，事物，日志等。你给先定义好把，然后在想用的地方用一下。

（切入点使用的公共代码）

类型：前置通知， 后置通知， 异常通知， 最终通知， 环绕通知

2. 连接点 (JoinPoint)

即使用到的所有方法

3. 切入点 (Pointcut)

上面说的连接点的基础上，来定义切入点（添加方法）

4. 切面 (Aspect)

切入点和通知（引介）的结合

5. 引入 (introduction)

允许我们向现有的类添加新方法属性。这不就是把切面（也就是新方法属性：通知定义的）用到目标类中吗

6. 目标 (target)

引入中所提到的目标类，也就是要被通知的对象，也就是真正的业务逻辑，他可以在毫不知情的情况下，被咱们织入切面。而自己专注于业务本身的逻辑。

7. 代理 (proxy)

怎么实现整套aop机制的，都是通过代理，这个一会给细说。

8. 织入 (weaving)

把切面应用到目标对象来创建新的代理对象的过程。有3种方式，spring采用的是运行时，为什么是运行时，后面解释。

关键就是：切点定义了哪些连接点会得到通知