# 一、简介

## 1. AOP定义

在运行时，**动态地将代码切入到类的指定方法、指定位置上的编程**，就叫做面向切面。

## 2. AOP应用背景

1. 即假如一个流程分三个步骤，分别是X,A,Y，另一个流程的三个步骤是X,B,Y。
2. 写在程序里，两个方法体分别是XAY和XBY，出现了重复，违反了DBY原则，如果把XY抽成一个方法，那么A和B都会和XY产生耦合。
3. 应用AOP就是在逻辑代码AB需要时，随时加入XY的代码，不用在AB中调用XY的方法体。在哪里加入XY就是切点，XY就是增强即通知。

## 3. 核心概念

1. 横切关注点 -- 用来插入到切面的方法

对哪些方法进行拦截，拦截后怎么处理，这些关注点称之为横切关注点

1. 切面（aspect） -- 通知和切点的结合

类是对物体特征的抽象，切面就是对横切关注点的抽象。

1. 连接点（joinpoint）-- 在应用执行过程中能够插入切面的一个点

被拦截到的点，因为Spring只支持方法类型的连接点，所以在Spring中连接点指的就是被拦截到的方法，实际上连接点还可以是字段或者构造器

1. 切入点（pointcut）

一组连接点的总称，用于指定某个增强应在哪个方法被调用

1. 通知（advice）（增强） -- 在特定连接点执行的动作

所谓通知指的就是指拦截到连接点之后要执行的代码，通知分为前置、后置、异常、最终、环绕通知五类

1. 目标对象

代理的目标对象

1. 织入（weave）

将切面应用到目标对象并导致代理对象创建的过程

1. 引入（introduction）

在不修改代码的前提下，引入可以在运行期为类动态地添加一些方法或字段

# 二、AOP实际应用

## 1. AOP环境搭建

(1) 配置文件

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:aop="http://www.springframework.org/schema/aop"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-3.0.xsd ">

<!-- bean definition & AOP specific configuration -->

</beans>

(2) Jar包准备

aspectjrt.jar、aspectjweaver.jar、aspectj.jar、aopalliance.jar

## 2. 利用XML实现切面

(1) aop.xml配置



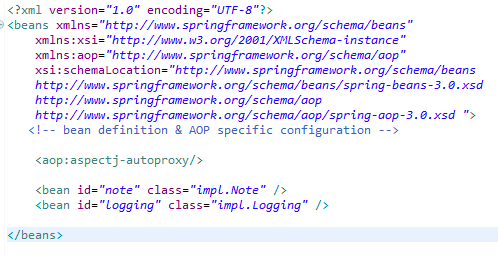
expression="execution(\* com.lin.springDemo.PaperExt.\*(..))"

(2) 实现代码



## 3. 利用注解实现切面

(1) autoAop.xml配置



(2) 声明一个@Aspect @Pointcut @Before

