###### AOP简介

AOP（Aspect Orient Programming），面向切面编程，是面向对象编程OOP的一种补充。面向对象编程是从静态角度考虑程序的结构，而面向切面编程是从动态角度考虑程序运行过程。

AOP底层就是采用动态代理模式实现的。采用了两种代理：JDK的动态代理（被代理对象必须要实现接口，才能产生代理对象），与CGLIB的动态代理（可以对任何类生成代理，代理的原理是对目标对象的继承代理，如果目标对象被final修饰，name该类型无法被cglib代理）。

面向切面编程，就是将交叉业务逻辑封装成切面，利用AOP容器的功能将切面织入到主业务逻辑中。所谓的交叉业务逻辑是指，通用的、与主业务逻辑无关的代码，如安全检查、事物、日志等。

若不使用AOP，则会出现代码纠缠，即交叉逻辑与主业务逻辑或者在一起。这样，会使主业务逻辑变得混杂不清。

例如，转账，在真正的业务逻辑前后，需要权限控制、日志记录、加载事务、结束事务等交叉业务逻辑，而这些业务逻辑与主业务逻辑间并无直接关系。但，它们的代码量所占比重能达到总代码量的一半甚至还多。

###### AOP编程术语

1. 切面（Aspect）

切面泛指交叉业务逻辑。事物处理、日志处理就可以理解为切面。常用的切面有通知与顾问。实际就是对主业务逻辑的一种增强。

1. 织入（Weaving）

织入就是指将切面代码插入到目标对象的过程。上例中MyInvocationHandler类中的invoke()方法完成的工作，就可以称为织入。

1. 连接点（JoinPoint）

连接点指可以被切面织入的方法。通常业务接口中的方法均为连接点。

1. 切入点（Pointcut）

切入点指切面具体织入的方法。被标记为final的方法是不能作为连接点与切入点的。以为最终是不能被修改，不能被增强的。

1. 目标对象（Target）

目标对象指要被增强的对象。

1. 通知（Advice）

通知是切面的一种实现，可以完成简单织入功能。换个角度说，通知定了增强代码切入到目标代码的时间点，是目标方法执行执行，还是之后执行等。通知类型不同，切入的时间不同。

切入点定义切入的位置，通知定义切入的时间。

1. 顾问（Advisor）

顾问是切面的另一种实现，能够将通知以更复杂的方式织入到目标对象中，是将通知包装为更复杂切面的装配器。

###### AOP编程术环境搭建

除了原Spring基本Jar包，还需要两个jar包：AOP联盟的规范接口包及Spring对其的实现包。