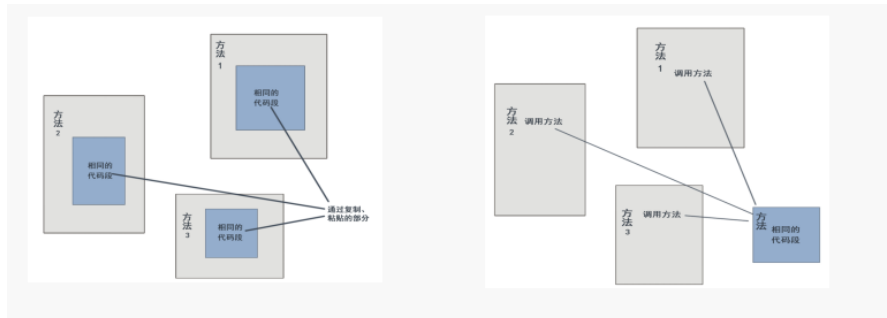


AOP 专门用于处理系统中分布于各个模块中的交叉关注点的问题，
在 Java EE 应用中，常常通过 AOP 来处理一些具有横切性质的系统级服务，如事务管理、
安全检查、缓存、对象池管理等，AOP 已经成为一种非常常用的解决方案：



AOP 代理其实是由 AOP 框架动态生成的一个对象，该对象可作为目标对象使用，
AOP 代理所包含的方法与目标对象的方法如下图所示：



以下是官方文档所给出的AOP的关键概念的解释：

切面 - Aspect

连接点 - Join Point

通知 - Advice

切入点 - Point Cut

引入 - Introduction

目标对象 - Target Object

AOP代理 - AOP Proxy

织入 - Weaving

AOP通俗的理解：

一个组件A，不关心其他常用的服务组件B，

但是这个组件A使用组件B的时候，不是组件A自身去调用，

而是通过配置等其他方式，比如Spring中可以通过xml配置文件。

这样就使得A压根就不需要知道服务组件B是怎样的，

爱存在不存在，爱怎么存在都与A无关。

A只关心自己的业务逻辑，具体A使用B的时候，配置文件去做，与具体的A组件无关。