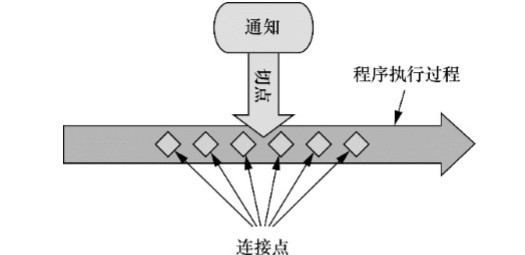
面向切面编程

下图展示了AOP的概念如何关联在一起：



在一个或者多个连接点上可以把切面功能（通知）织入到程序的执行过程

1. 通知：advice

切面的工作被称为通知。在Spring中可以使用5种类型的通知：

* 前置通知（Before），在目标方法被调用之前调用通知功能
* 后置通知（After），在目标方法完成之后调用通知，此时不会关心方法的输出
* 返回通知（After-returning），在目标方法成功执行后调用通知
* 异常通知（After-throwing），在目标方法抛出异常后调用通知
* 环绕通知（Around），通知包裹了被通知的方法，在被通知的方法调用之前和调用之后执行自定义的行为

1. 切点：pointcut

通常使用明确的类和方法名称，或是利用正则表达式定义所匹配的类和方法名称来指定这些切点

1. 连接点：joint point

连接点是在应用执行过程中能够插入切面的一个点。这个点可以是调用方法时、抛出异常时、甚至修改一个字段时。切面代码可以利用这些点插入到应用的正常流程之中，并添加新的行为

1. 切面：Aspect

切面是通知和切点的结合，通知和切点共同定义了切面的全部内容

1. 引入，Introduction

引入允许我们向现有的类添加新方法或属性

1. Weaving

织入是把切面应用到目标对象并创建新的代理对象的过程。切面在指 定的连接点被织入到目标对象中。在目标对象的生命周期里有多个点可以进行织入：

* 编译期：切面在目标类编译时被织入。这种方式需要特殊的编译 器。AspectJ的织入编译器就是以这种方式织入切面的。
* 类加载期：切面在目标类加载到JVM时被织入。这种方式需要特 殊的类加载器（ClassLoader），它可以在目标类被引入应用 之前增强该目标类的字节码。AspectJ 5的加载时织入（load-time weaving，LTW）就支持以这种方式织入切面。
* 运行期：切面在应用运行的某个时刻被织入。一般情况下，在织 入切面时，AOP容器会为目标对象动态地创建一个代理对象。 Spring AOP就是以这种方式织入切面的。