

# aspectj框架的具体使用

## 1、切面的执行时间，在AOP规范中叫做Advice（通知，增强）

使用五个注解（表示五个时间点）：

- 1、@Before
- 2、@AfterReturning
- 3、Around
- 4、@AfterThrowing
- 5、@After

当然也可以使用xml的方式，使用其中的标签和属性进行切面执行时间点的配置，但是不常用

## 2、切面执行位置的配置（就是定位寻找添加功能增强的目标方法）。

使用的切入点表达式（Pointcut）：

aspectj框架中指定了专门的表达式用于指定切入点，表达式如下：

```
execution(modifiers-pattern?
           ret-type-pattern
           declaring-type-pattern?name-pattern(param-pattern)
           throws-pattern?)
```

解释：

**execution**（中文：处决，执行，实施）表达式内容中可以填的内容有四个标签部分，其中有问号的部分表示可选项，

可填可不填

就是说通过表达式中的两个必填部分可以指定目标方法，指定一个目标就是指定了连接点；若表达式指定了多个目标

方法就是说指定了切入点，一般在项目中都是多个目标方法，所以指定切入点用的多。

每个部分用空格隔开，上面用空格连接太长了，所以用回车表示更明了一些。

**modifiers-pattern**：代表填写的是切入点的访问权限修饰符（**public**、**protected**..）可填可不填

**ret-type-pattern**：切入点（目标方法）的返回值类型，必填

**declaring-type-pattern**：切入点的包名类名，和下面的方法名连着的，可填可不填

**name-pattern(param-pattern)**：切入点的方法名以及所需的参数，必填，和上面包名类名是一部分

**throws-pattern**：切入点（目标方法）抛出的异常，可填可不填

**execution**(访问权限修饰符 返回值类型 方法名(参数)或者完整的方法名(参数) 异常类型)

通配符的使用：

**\***：表示0到多个字符

**..**：用在方法参数中：**name-pattern(param-pattern)**表示任意多个参数

用在包名类名位置：**declaring-type-pattern**表示当前包及子包路径

**+**：用在包名类名位置：**declaring-type-pattern**表示当前类及子类路径，**//**类和包是区分开的

用在接口后，表示当前接口及实现类

//举例

```
execution(public * *(..))
```

翻译为指定公共的返回值任意方法名任意参数类型和数量也任意的目标类

就是指定切入点（目标方法）为：任意的公共方法

`execution(* set*(..))`

可以看到这里只有两部分，所以只有返回值类型和方法参数

翻译就是指定切入点（目标方法）为：以`set`开头的任意方法

`execution(* com.xyz.service.*.*(..))`

两部分，只有返回值和包名类名方法名参数

因为完整的方法名是：包名.类名.方法名(参数)

翻译就是指定切入点（目标方法）为：`com.xyz.service`包下任意类中的任意方法

`execution(* *..service.*.*(..))`

两部分，只有返回值和包名类名方法名参数

翻译就是指定切入点（目标方法）为：所有包下的任意`service`包下的任意类的任意方法