# AOP-重学

1,什么是AOP?

通过预编译方式,在运行期动态代理实现程序功能统一维护的一种技术.

2,预编译

就是横切的一种说法,横切写法为 execution(\* com.turing.\*.service.impl.\*.\*(..))

解释: 方法的返回类型 类的全路径.方法名(参数个数和类型)

3,动态代理

为了实现AOP.必须 建立一个代理类,来写切面的程序.

4,统一维护

只要能实现横切,那么写的代码就可以作用在横切的方法上(横切的连接点上).

技术:

1,需要一个代理类

2,需要写横切 execution(\* com.turing.spring.aop.\*.\*(..)

3,写通知方式(前置,后置,最终,异常,环绕)

声明式:

1,代理类 <bean id=”” class=””>

2, 横切

<aop:pointcut id="point1" expression="execution(\* com.turing.spring.aop.\*.\*(..))" />

3,通知:

<aop:around method="run" pointcut-ref="point1"/>

注解式:

0:在spring容器里,添加启用aop注解的驱动.

1,代理类 <bean id=”” class=””> 需要在代理类上添加一个注解@Aspect

2, 横切 : 横切写在代理类的一个方法上@PointCut

3,通知: 在通知方式的注解里添加横切所在的方法名