1. **引入其它配置文件：**

<import resource=*"spring01.xml"*/>

1. **注解：**

**I：在spring配置文件中添加**<context:component-scan base-package=*"com.hw"*>

</context:component-scan>，**作用是扫描指定包以及子包下的注解类并生成相应的bean**

**II：可以在类上添加注解@Component为该类生成相应的bean**

@Component(“userBean”)等同于

<bean id=”userBean” class=””/>

III：@Controller**一般用于控制层，作用与@Component一样**

**IV：@Service，一般用于业务层，作用与@Component一样**

**V：@Repository，一般用于持久层，可以生成bean，还可以将所标注类中的数据访问异常封装为spring数据访问异常，不可以替换**

1. **AOP——面向切面编程**

**I:单位是方法、切面是类**

**II:配置方式**

**a首先定义一个切面类，在spring配置文件中生成这个切面类的bean**

<bean id=*"aspectA"* class=*"com.hw.service.impl.AspectA"* ></bean>

**b 添加aop配置：<aop:config>**

**c 添加切入点pointCut**

<aop:pointcut expression=*"execution(\* com.hw.service..\*(..))"* id=*"pointcutA"*/>

d 添加通知方法

<aop:before method=*"doBefore"* pointcut-ref=*"pointcutA"*/>

<aop:after-returning method=*"doAfterReturning"* pointcut-ref=*"pointcutA"*/>

<aop:after-throwing method=*"doAfterThrowing"* pointcut-ref=*"pointcutA"* throwing=*"e"*/>

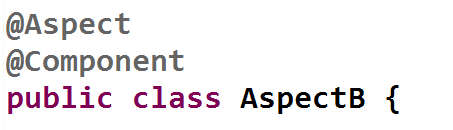
<aop:after method=*"doAfter"* pointcut-ref=*"pointcutA"*/>

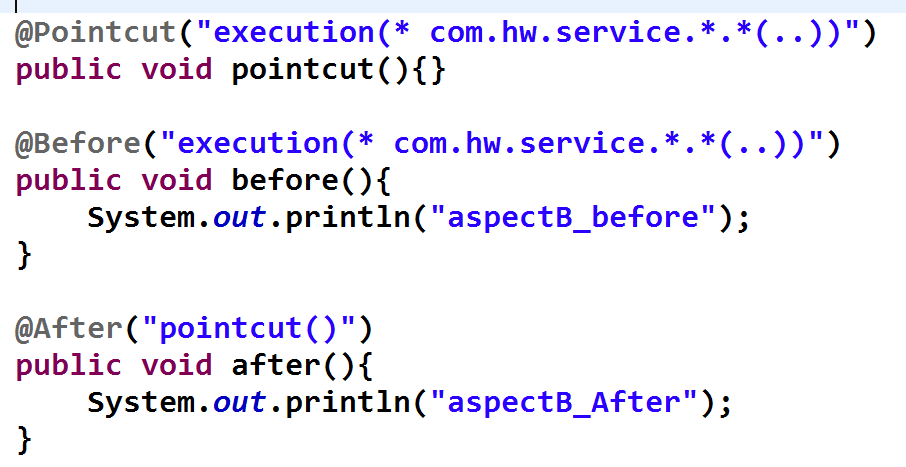
<aop:around method=*"doAround"* pointcut-ref=*"pointcutA"*/>

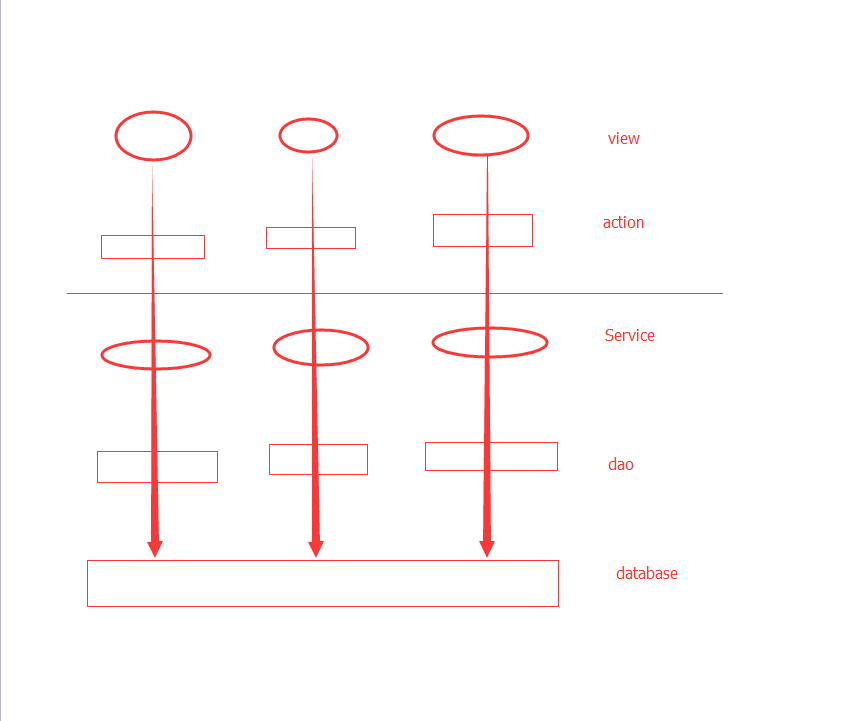
**III：注解方式**

**在配置文件中添加**



**切面类中添加：**





**练习：给整个项目的Service层添加记录日志功能，当业务层出现异常时，将异常信息写入日志文件**

**作业：**

**1、是否理解AOP的概念，如果是，请简述你对AOP的理解**

**2、搭建hibernate框架，新建账户类Account，包含账户名称（主键）、真实姓名、账户余额等属性**

**3、在Service层添加转账方法，可以通过接收参数，将钱从A的账户转到B的账户**

**4、通过spring AOP给该转账功能添加事务控制，如果转账失败，则回滚事务，成功，则提交事务**