

```

package com.aop.annotation;

import org.aspectj.lang.ProceedingJoinPoint;
import org.aspectj.lang.annotation.After;
import org.aspectj.lang.annotation.AfterThrowing;
import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Before;
import org.springframework.stereotype.Component;

@Component
@Aspect
public class MyAdvice {
    @Before("execution(public void com.aop.annotation.Demo.demo1(String,int)
)&& args(name,id)")
    public void before(String name,int id){
        System.out.println("执行前置通知name:"+name+"id:"+id);
    }

    public void after(String name,int id){
        System.out.println("执行after通知name:"+name);
    }

    public void afterReturning(String name,int id){
        System.out.println("执行afterReturing通知name:"+name);
    }

    @AfterThrowing(value = "execution(public void
com.aop.annotation.Demo.demo1(String,int) )&& args(name,id)",throwing = "e")
    public void mythrowing(Exception e,String name,int id){
        System.out.println("执行异常通知:"+e.getMessage());
    }

    public Object myarround(ProceedingJoinPoint p,String name,int id) throws
Throwable {
        System.out.println("执行前置通知");
        Object result=p.proceed();
        System.out.println("执行后置通知");
        return result;
    }
}

```