

```
package com.aop;
```

```
import org.aspectj.lang.ProceedingJoinPoint;
```

```
//如果pointcut有参数,则通知也必须有参数
```

```
public class MyAdvice {
```

```
    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);  
    }
```

```
    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;  
    }
```

```
}
```