JoinPoint、ProceedingJoinPoint接口

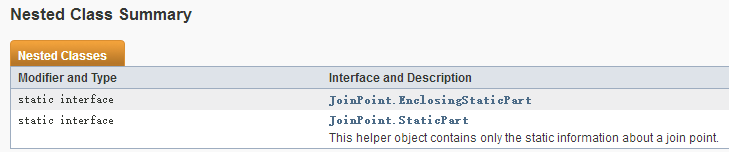
# JoinPoint连接点：

**JoinPoint接口**存在于**org.aspectj.lang**包。有一个子接口**ProceedingJoinPoint。**



功能：Provides reflective access to both **the state available at a join point** and static information about it. This information is available from the body of advice using the special form thisJoinPoint. The primary use of this reflective information is for tracing and logging applications.

# 内部接口：NestedInterfaces



# 主要方法：Methods

## getXxx()方法

### getArgs() : 返回Object[] 数组，目标方法的参数数组。



### getKind() ： String 获取该连接点的类型。



### getTarget() ： Object 获取目标对象。



### getThis() ： Object 获取当前执行的对象。



### getSignature() ： Signature，获取连接点的签名对象。



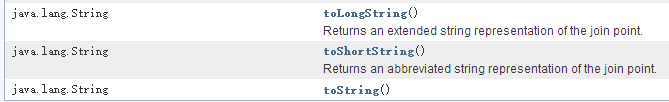
### getSourceLocation() ： SourceLocation 获取该连接点对应的SourceLocation对象。



### getStaticPart()：JoinPoint.StaticPart,获取连接点的静态部分。



## toString()、toXxxString()方法



# 子接口：ProceedingJoinPoint

## ProceedingJoinPoint：

存在于包org.aspectj.lang中。是**JoinPoint的子接口**。



## 功能：

ProceedingJoinPoint exposes the proceed(..) method in order to support around advice in @AJ aspects

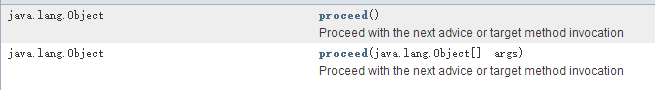
主要是用来支持环绕通知Around。ProceedingJoinPoint接口中定义了**proceed()、proceed(Object[] args)**方法。

## 除了继承JoinPoint方法外，特有方法

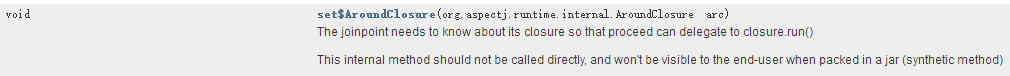
### proceed()、proceed(Object[] args)方法

无参数的proceed方法用来调用原来待执行的方法，不修改参数；

有参数Object[] args方法可以传入修改后的参数，并执行原方法。



### set$AroundClosure



# Signature接口

## Signature接口

存在于包org.aspectj.lang中，有很多子接口，有一个实现类FieldSignatureImpl。



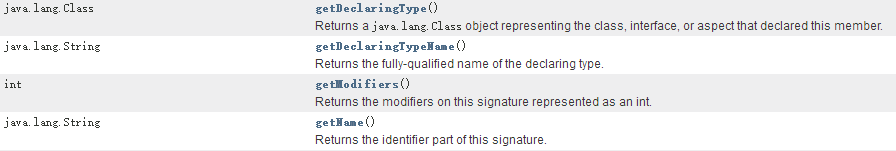
Represents the signature at a join point. This interface parallels java.lang.reflect.Member.

This interface is typically used for tracing or logging applications to obtain reflective information about the join point, i.e. using the j2se 1.4 java.util.logging API

## 方法：

### getName()、getModifiers()、getDeclaringTypeName()、getDeclaringType()

modifier修饰符



### toString、toXxxString

