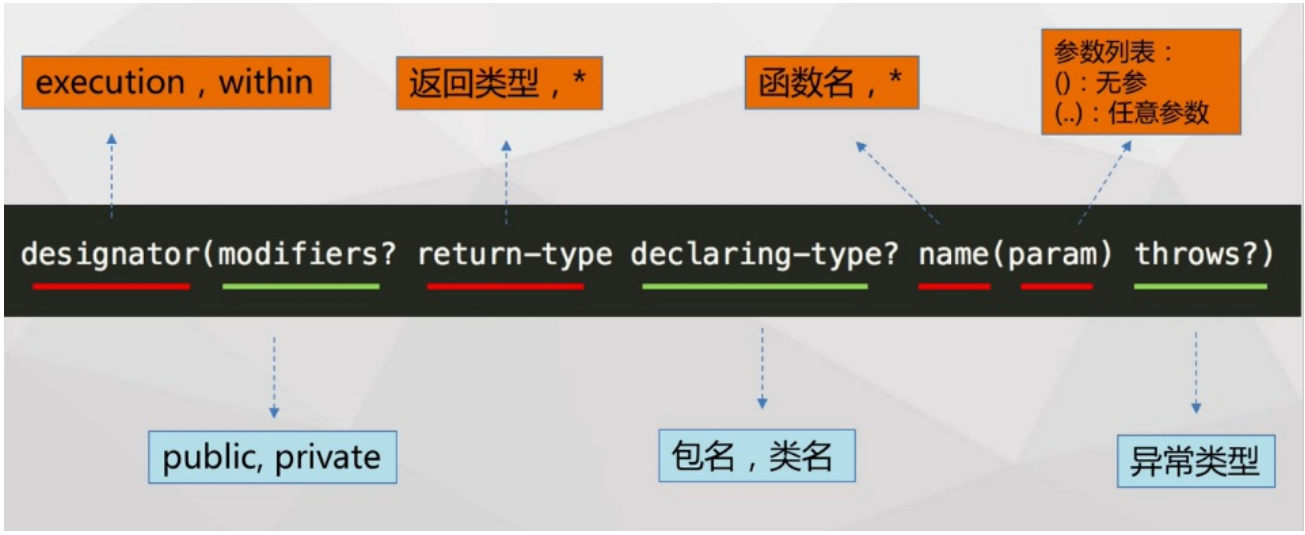
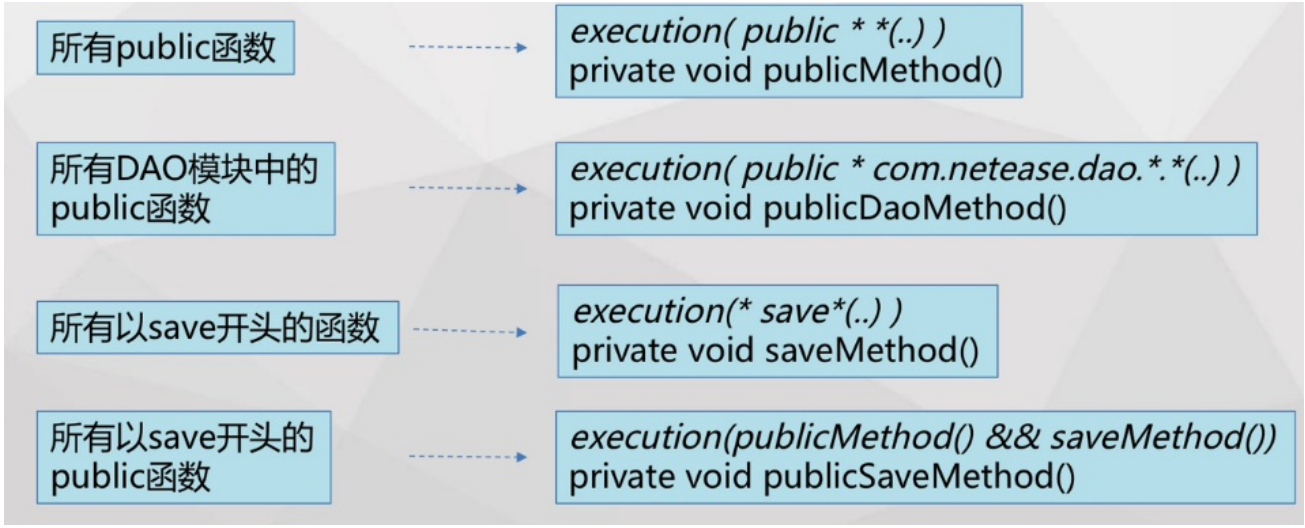
3.3 补充

pointcut语法:



pointcut 事例:



获得函数上下文:

```
@Before("com.netease.course.LoggingAspect.arithmetic()")
public void doLog(JoinPoint jp) {
    System.out.println(jp.getSignature() + ", " + jp.getArgs())
}
```

Around 获得上下文,比较特殊:

```java
@Around("com.netease.course.LoggingAspect.arithmetic()")
public void doLog(ProceedingJoinPoint pjp) {
    System.out.println("start method: " + pjp.toString())

    Object retVal = pjp.proceed();

    System.out.println("stop method: " + pjp.toString())
    return retVal;
}
```

获得返回值:returning放前面

```java
@AfterReturning(
        pointcut="com.netease.course.LoggingAspect.arithmetic()",
        returning="retVal")
public void doLog(Object retVal) {
    // do something with retVal
}
```

获得异常:

```java
@AfterThrowing(
        pointcut="com.netease.course.LoggingAspect.arithmetic()",
        throwing="ex")
public void doLog(IllegalArgumentException ex) {
    // do something with ex
}
```

获得目标函数制定参数:

```java
@Before("com.netease.course.LoggingAspect.arithmetic() && args(a, ..)")
public void doLog(JoinPoint jp, int a) {
    // do something with parameters
}
```