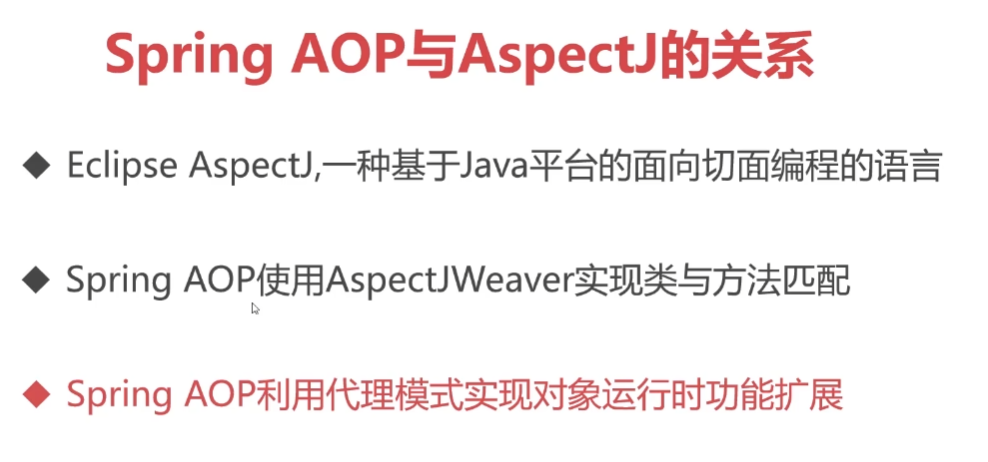
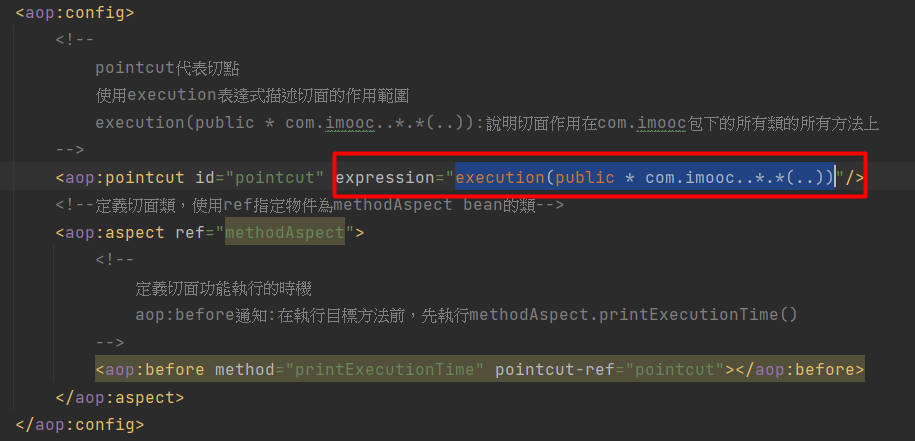
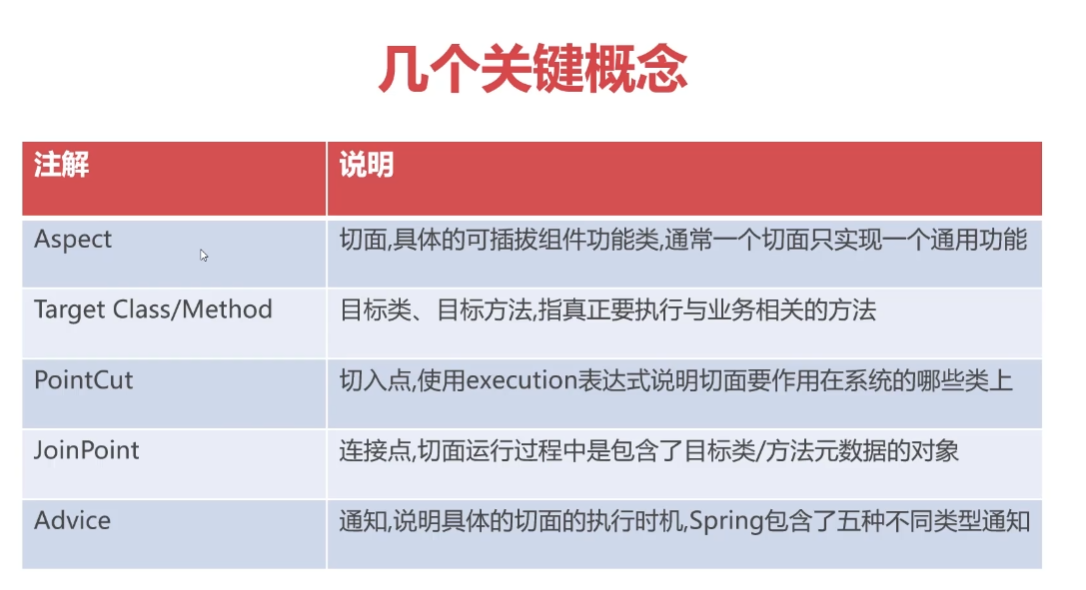
**AOP相關概念**

一、Spring AOP與AspectJ關係



AspectJWeaver功能就是將下圖的表達式進行解析，然後與實現類和方法進行匹配。

**二、關鍵概念點**



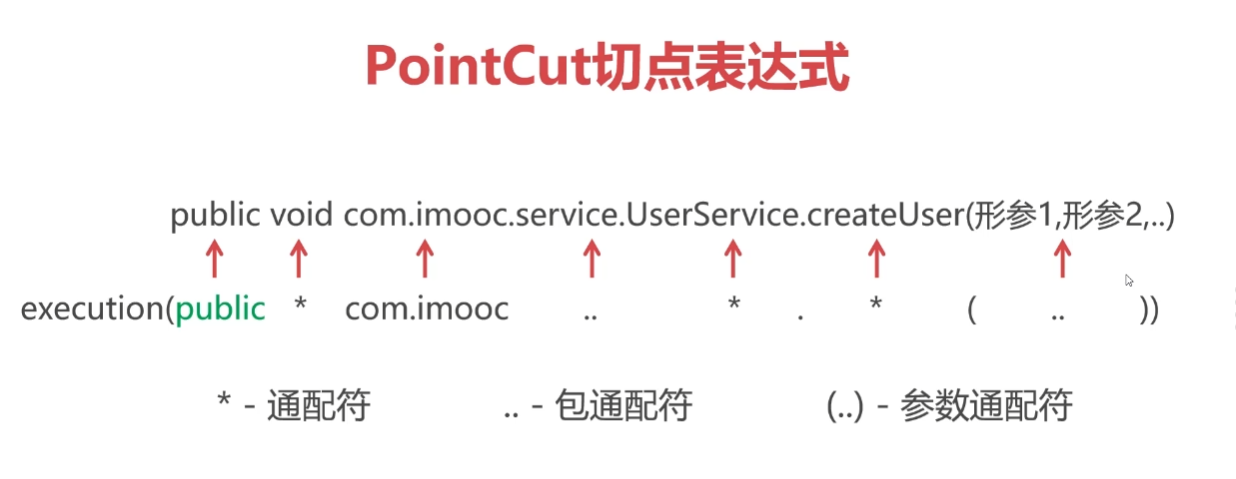
1.JoinPoint物件的核心方法



使用方法如下:



2.PointCut中的expression表達式



Execution表達式中，要輸寫要匹配的method名稱，如圖:

匹配所有再com.imooc包底下所有Class的所有方法中，用public修飾，不在乎有沒有返回值且不限參數數量的method。其中public可以省略不寫。

(1)範例1



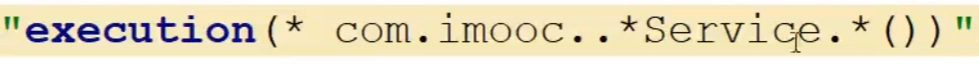
指定在com.imooc包下，以Service結尾的類。

(2)範例2



在範例1的基礎上再限制返回值為String的method上才會作用效果。

(3範例3



限定無參數的method



限定method的參數類型時，在星號之間使用逗號表示限定參數個數。

如上圖就是限定參數為兩個的method

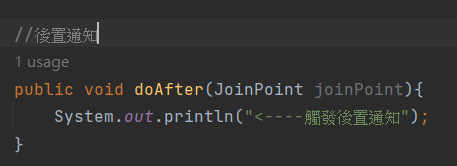


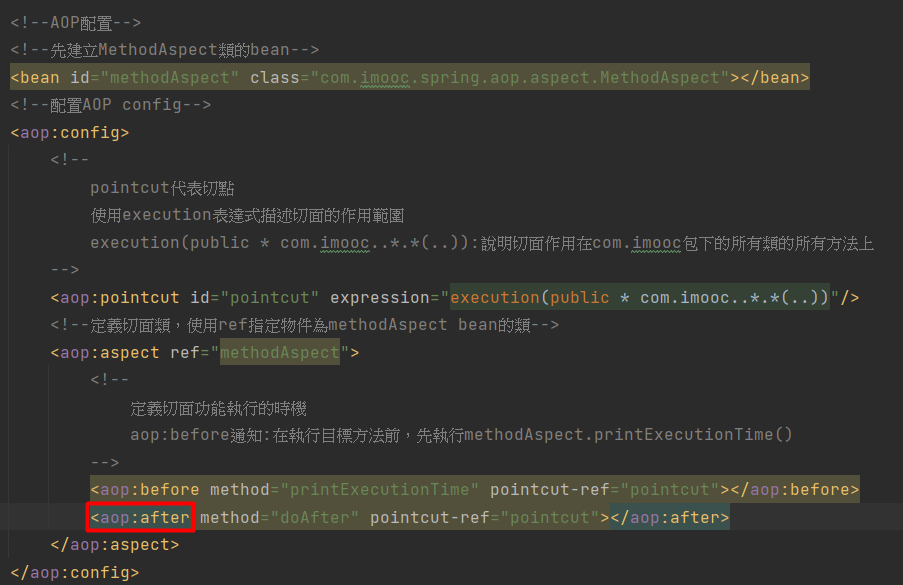
同時，也可以指定參數的類型，如上圖就是限定method為兩個參數，且第一個參數一定要是String。

**三、通知類型**



(1)After Advice

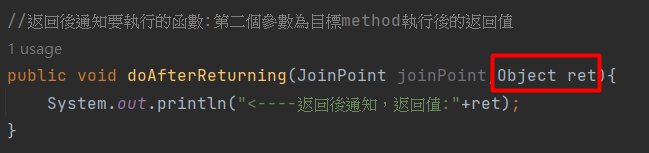




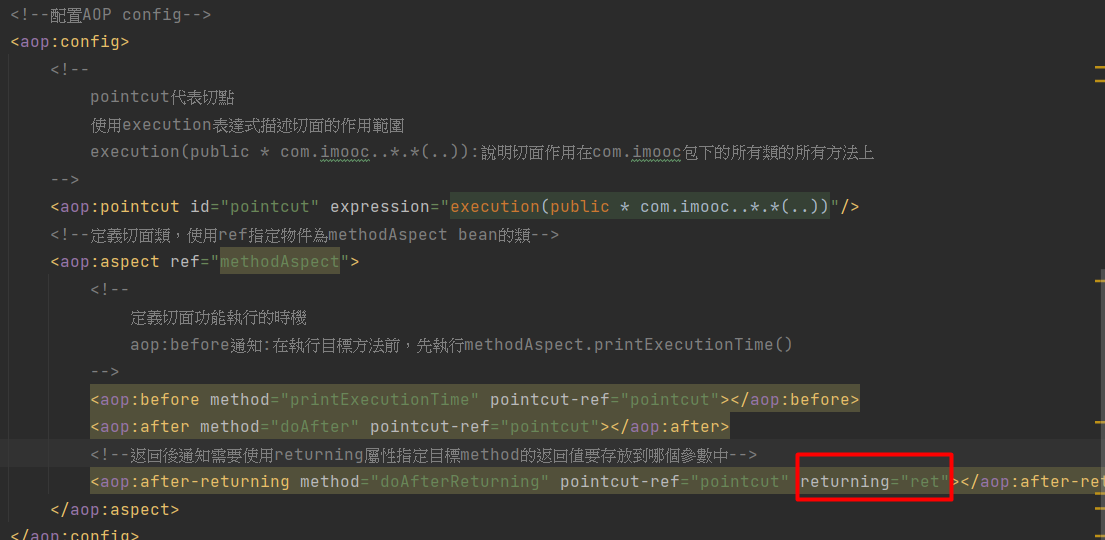
後置通知無法獲取目標method的返回值和異常。

(2)After Returning Advice

返回後通知要執行的函數必須包含一個Object參數，用來保存目標method的返回值。



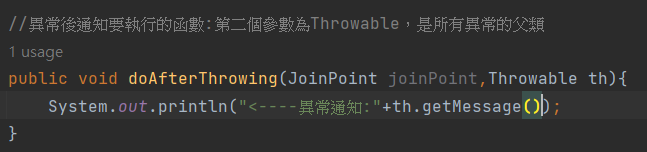
然後在AOP配置中，使用returning屬性指定目標method返回值要存放到哪個參數，也就是我們剛剛在doAfterReturning中書寫的ret參數。



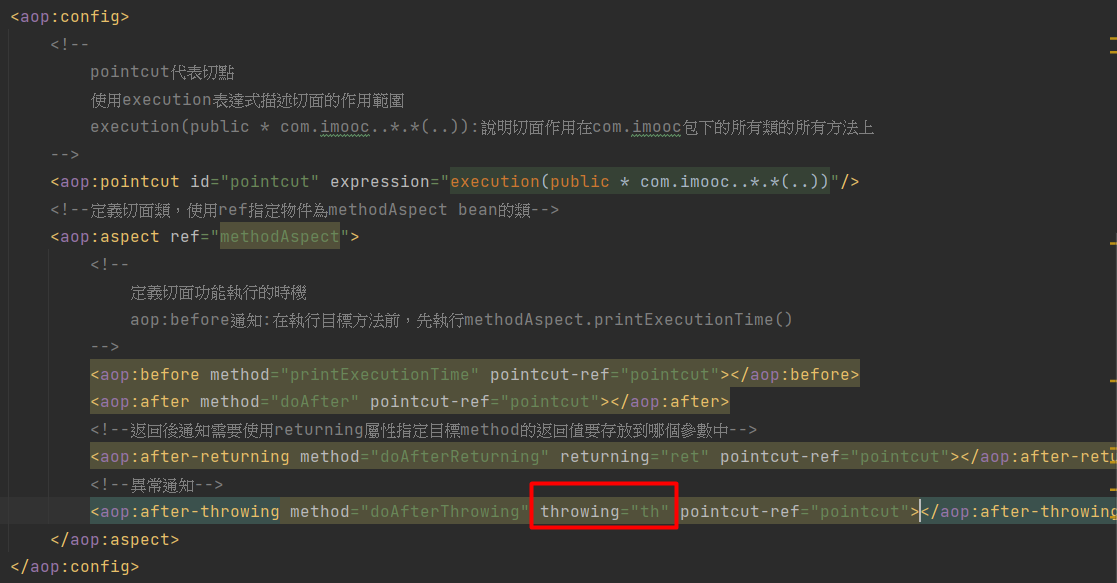
值得注意的是後置通知和返回後通知，這兩個通知的執行順序是依照AOP中書寫的順序來決定的，如上圖，是先寫aop:after再寫aop:after-returning因此，會先執行後置通知，再執行返回後通知。

(3)After Throwing Advice

異常通知，要執行的函數必須包含一個Throwable參數，用來保存異常物件

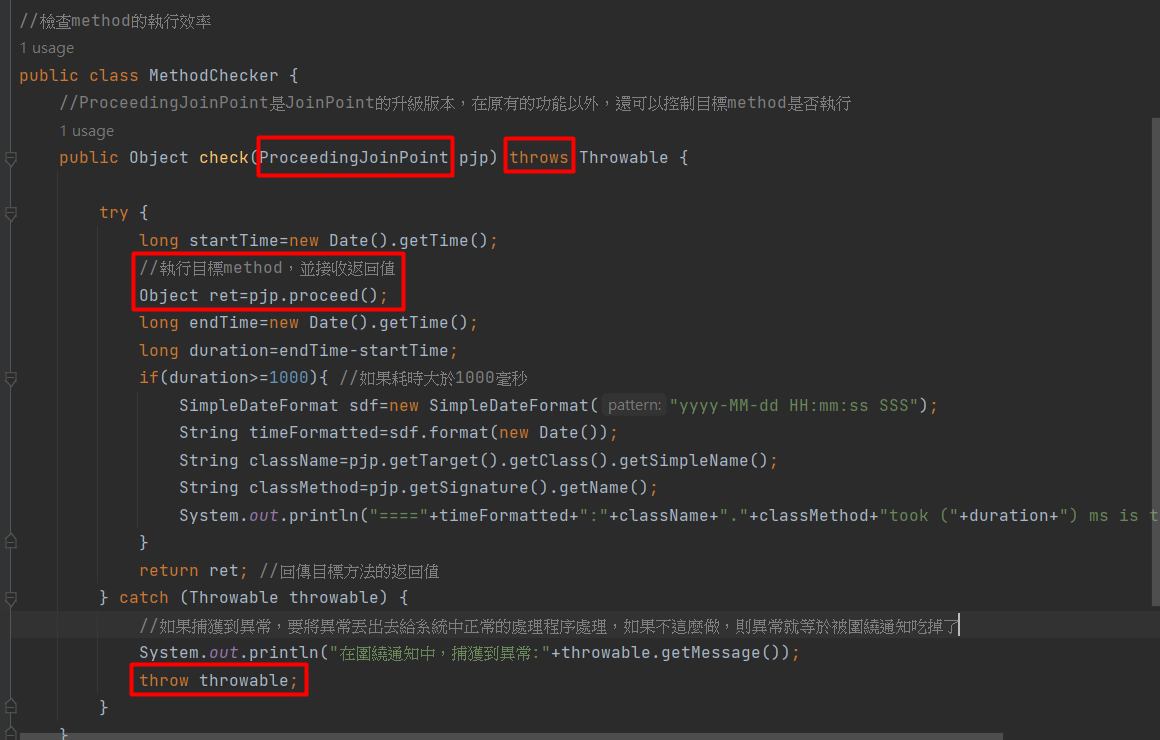


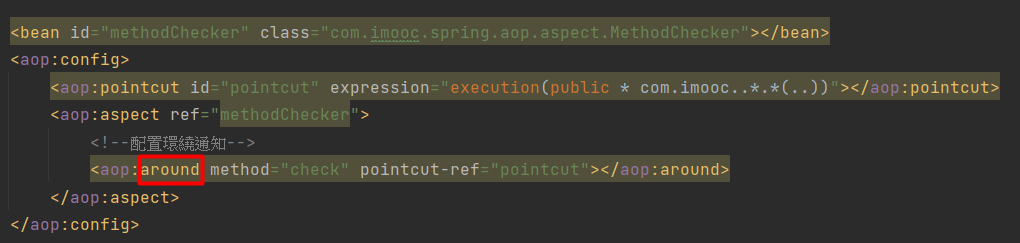
在AOP配置中，使用throwing屬性指定目標method拋出的異常要保存到的屬性。



(4)Around Advice

圍繞通知的實作方法要求有一個Object返回值，用來返回目標method的返回值。圍繞通知，可以控制目標method，決定是否要執行，透過調用ProceedingJoinPoint.proceed()來執行目標方法，並使用Object物件接收該方法的返回值。





**四、特殊的通知-引介增強(IntroductionInterceptor)**

