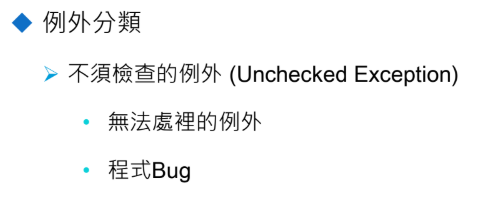
第十二章：例外處理

* 例外處理是在程式中負責處理當程式無法運行或出現錯誤時的補救機制，一般來說是在處理執行時發生的錯誤，編譯時的錯誤在一開始即會提示使用者。

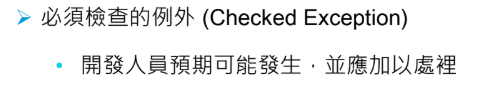
1. 錯誤的種類

* 無法處理的例外(error)

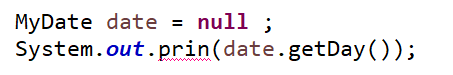


* 必須檢查的錯誤

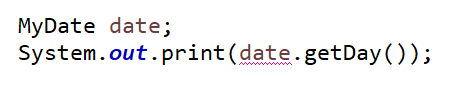
有分兩種：編譯時的錯誤 & 執行時的錯誤，屬於必須檢查的例外



1. 編譯時的錯誤(compile error)：程式在撰寫時期產生的錯誤，例如命名錯誤、斷句錯誤、格式問題等等，在執行前即會提示使用者修改



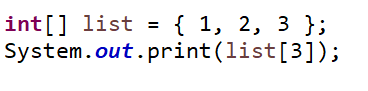
→拼字錯誤

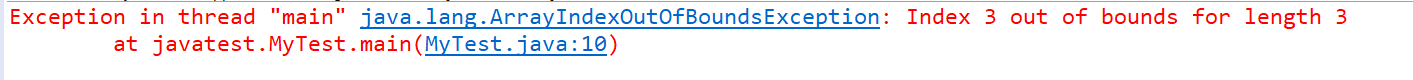


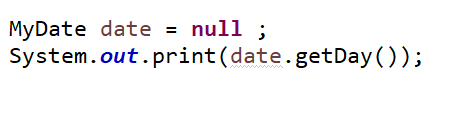
→尚未實體化的錯誤

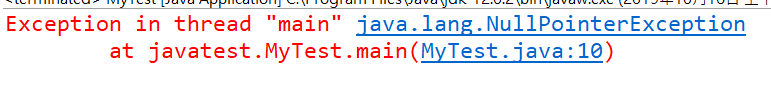
1. 執行時的錯誤(runtime error)：程式在編譯時期沒有任何問題，但執行後發現錯誤並終止程式繼續執行的狀況：例如除數為0、物件參照為null、無窮迴圈、資料不存在等等。大多數情況會造成程式直接中斷或當掉，造成無法挽回的錯誤！

* 執行時的錯誤範例

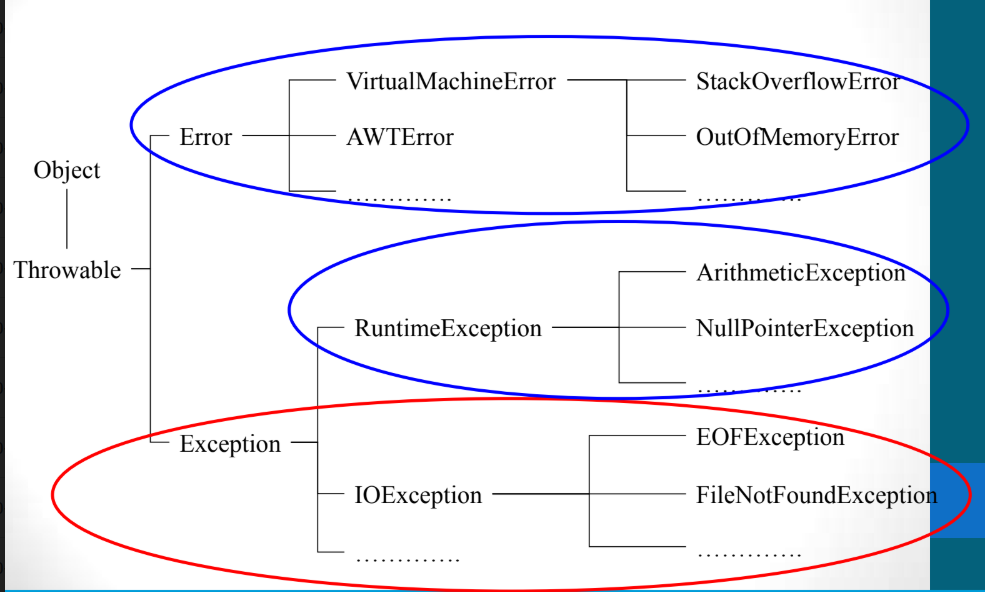
→編譯有過

↓執行後





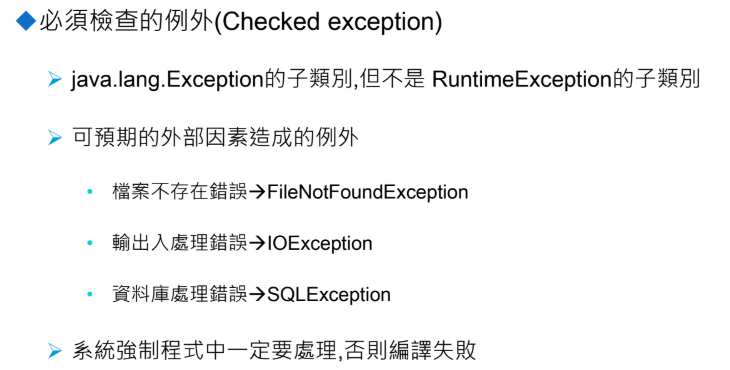
* 以上錯誤皆會造成程式當機中斷，若是正在傳輸重要資料或交易動作等，會造成資料流失！



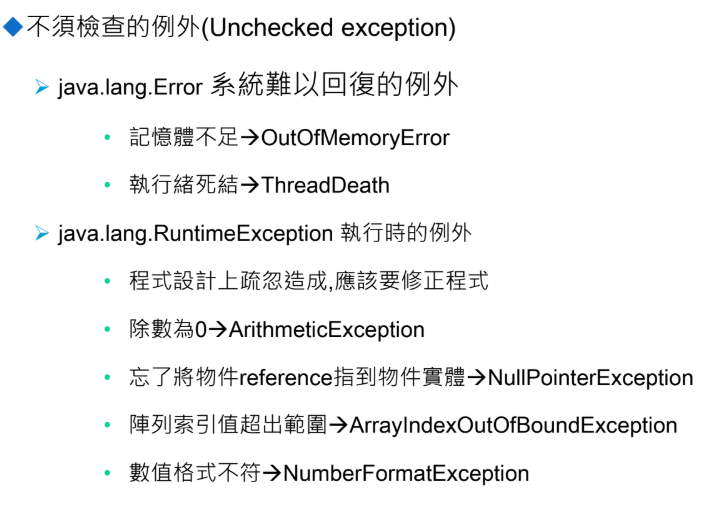
需要處理的，編譯時檢查不出來

執行前必需要處理的

無法處理的

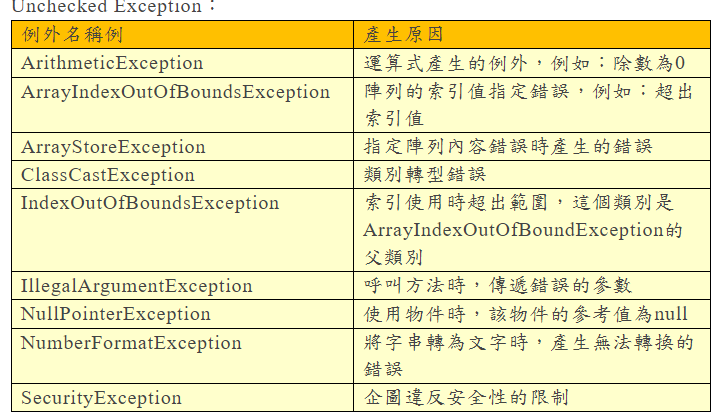


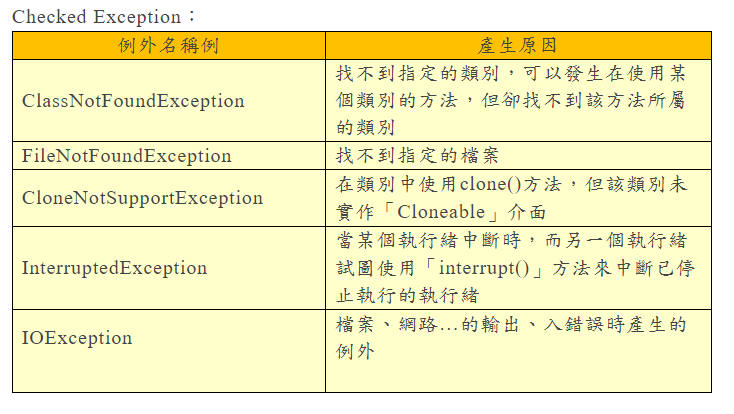
unchecked↓vs↑checked



→編譯時期會過，但執行時會出錯的情況稱為Unchecked exception

→常見的例外如下：

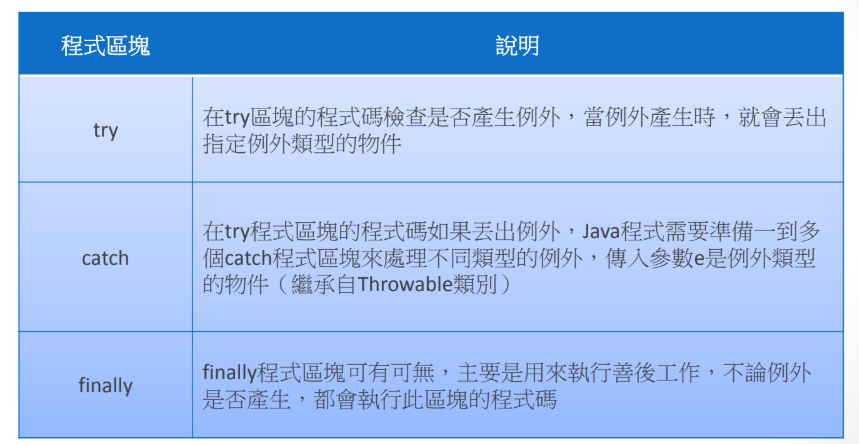




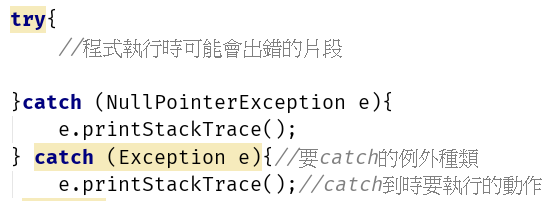
1. 例外處理的機制：try/catch

* 將可能會出現錯誤的程式碼用try/catch語法包住

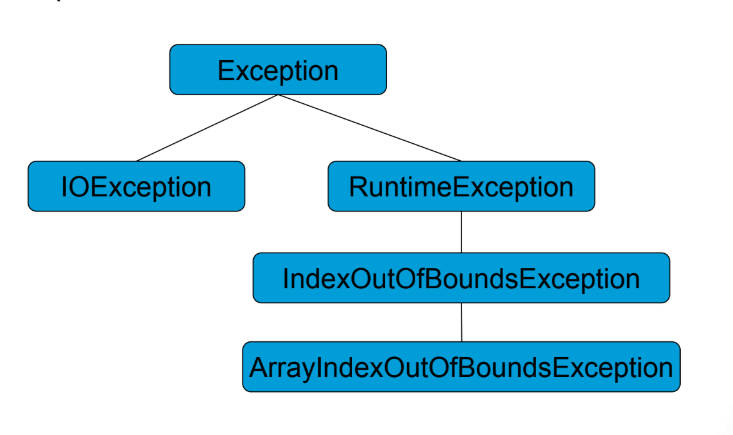




一個try可以接多個catch處理，代表可以攔截一段程式碼中可能會出現的數個錯誤，但請看下面：

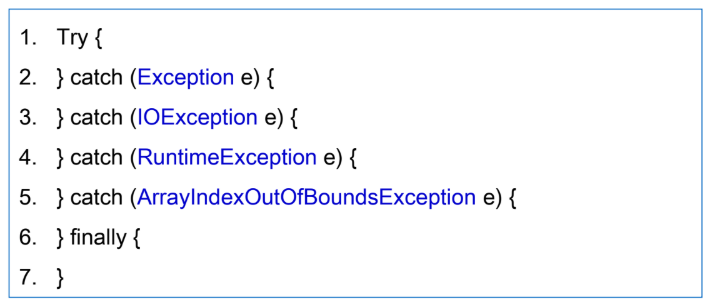


這樣的try/catch結構看似沒有問題，是要截取兩種可能的錯誤，但請注意！catch的例外種類範圍必須要由小>大，特定範圍>一般範圍來宣告！以上例來說搭配下圖：

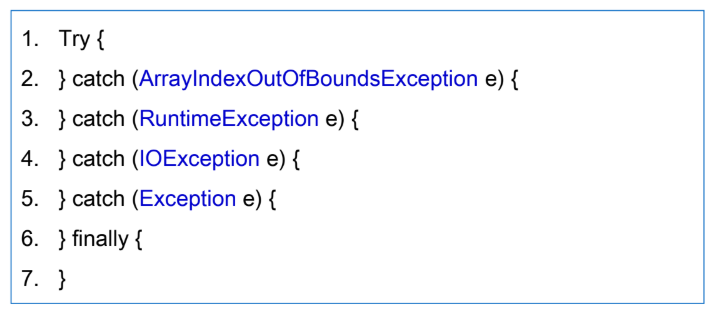


例外處理中，所有的根源類別皆是Exception類，然後往下有許多分支，例外處理在特定範圍有catch到之後就不會再往上catch了，所以不能寫反

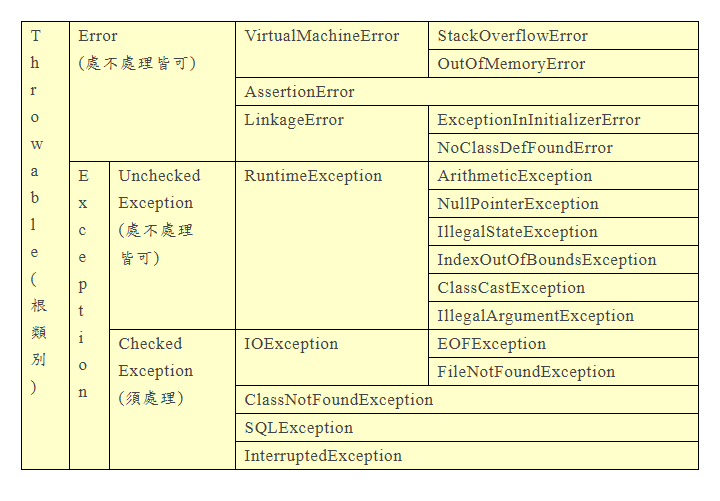
* 寫反的例子



* 正確的例子應該這樣子寫



→注意：例外種類涵蓋範圍愈大的要放在愈後面



<http://tianchyi1955.pixnet.net/blog/post/190707898>

* 處理例外的常用方法

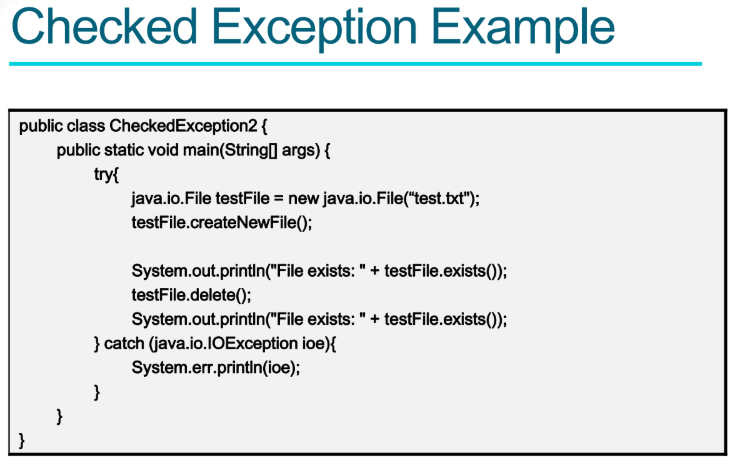
→e.printStackTrace

→e.getMessage

→System.err.print(e)

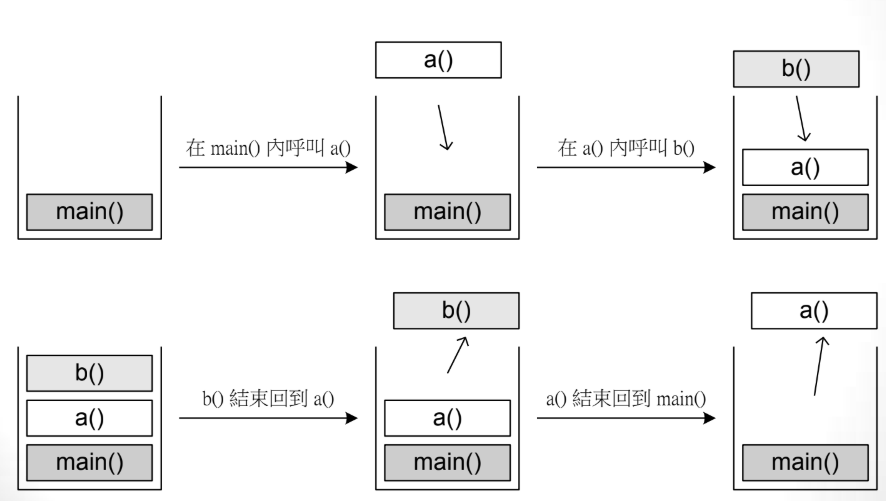
以上為印出或取得例外敘述的方法，在取log訊息時相當有用，不過如果需要實際處理時就要另外撰寫合適的程式碼

* 有強制要加上try/catch，不然編譯不會過的例子為Check Exception系列，意指撰寫時一定要加的必要寫法



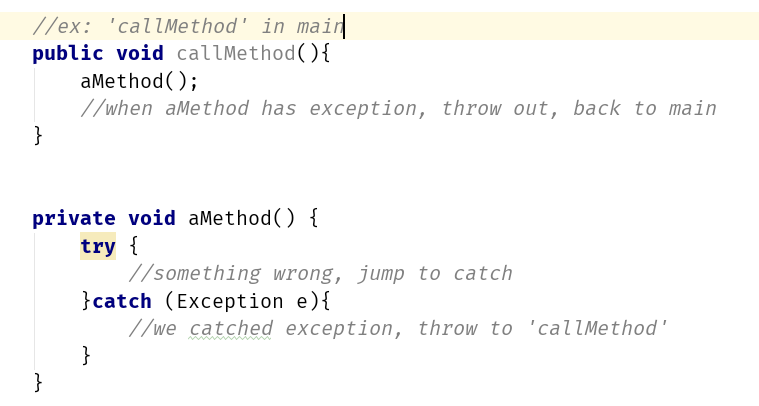
例外的傳遞的過程

* 方法的傳遞過程



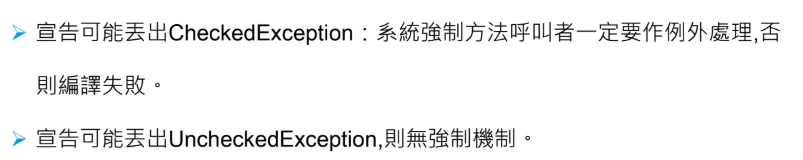
Main();

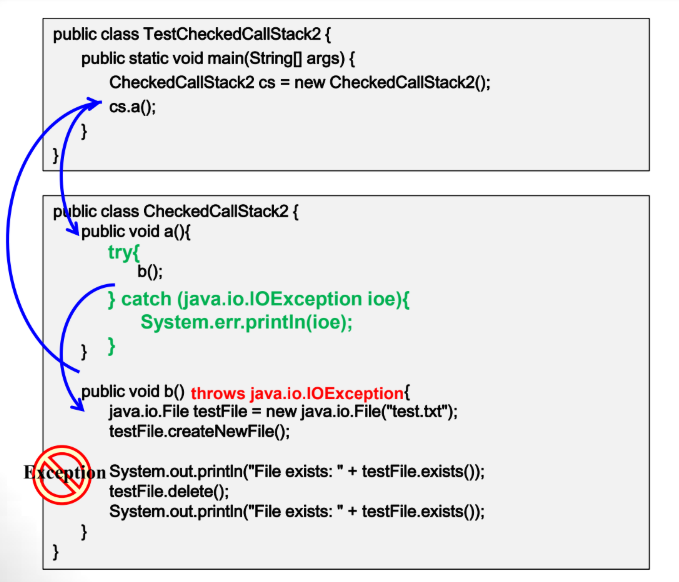
→例外的傳遞雷同

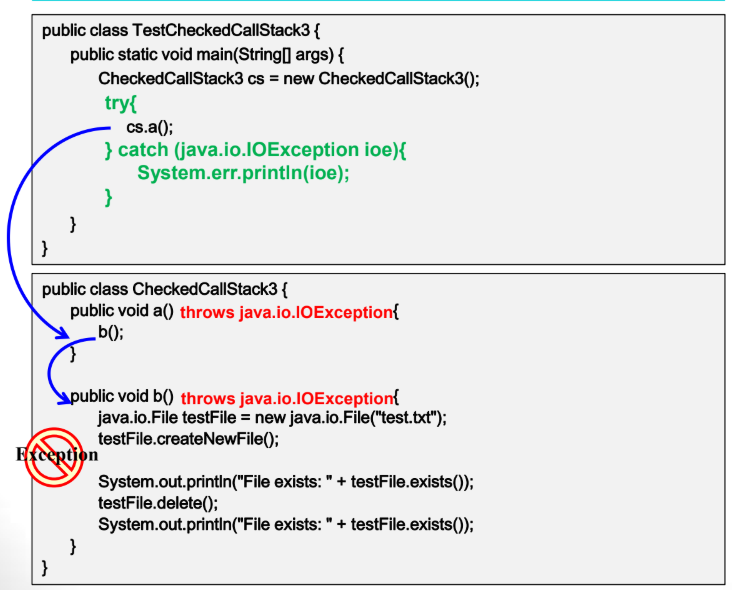


1. Throws的處理技巧

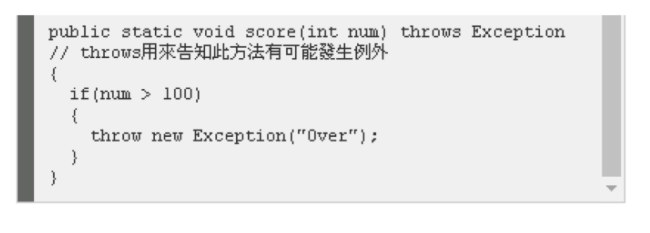
→方法中沒有寫try/catch時，在名稱後面註記回拋例外給上一層

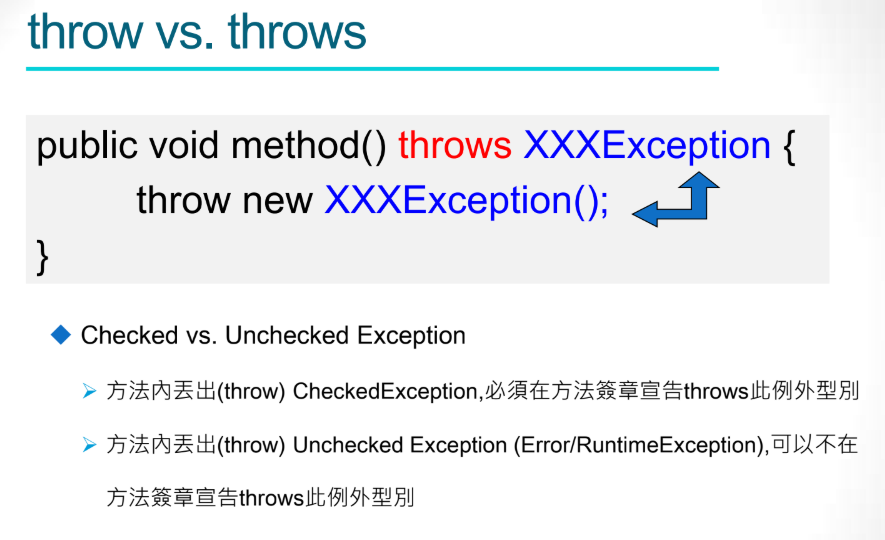




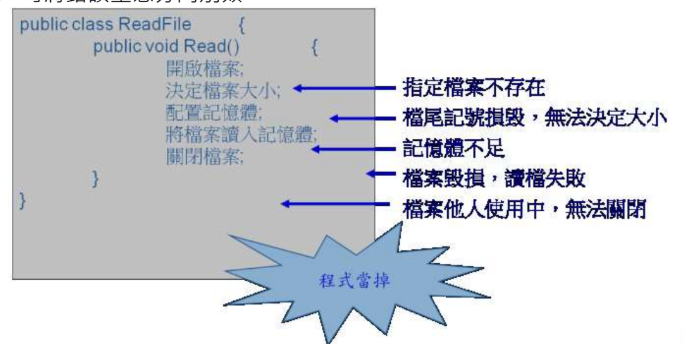


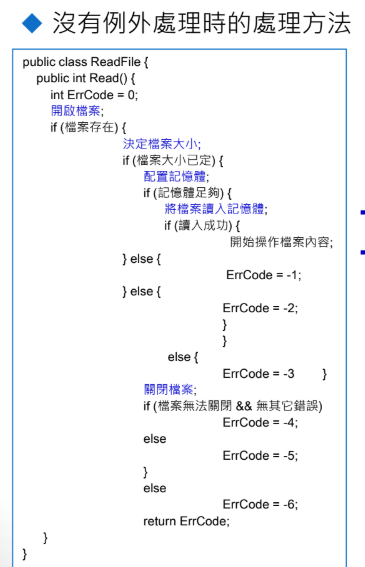
* 丟出例外物件給上一層處理



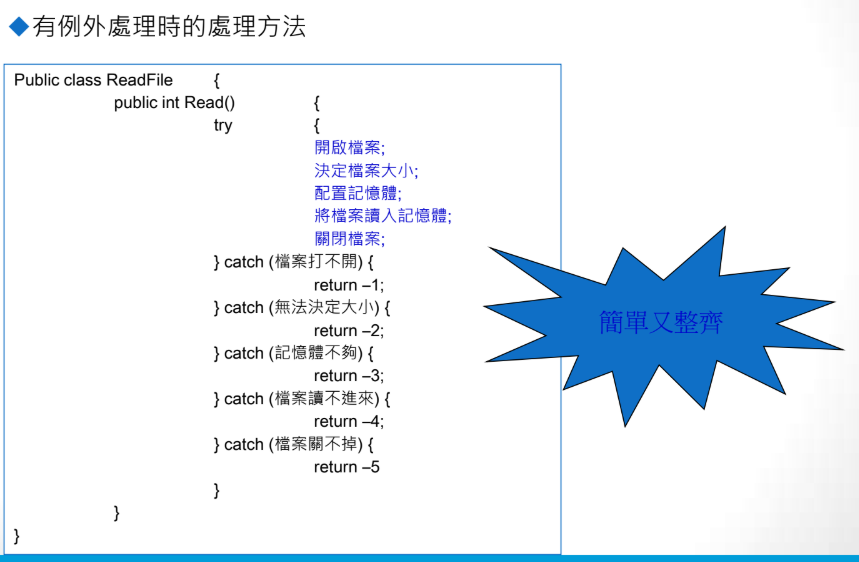


* 基本上只要出現例外錯誤，開發人員都是要處理的，不論如何回拋到多上面，最後大部份還是要寫出解決的方法，差別只在於在哪一層做處理而已
* 如果一個程式中沒有處理例外錯誤的話…

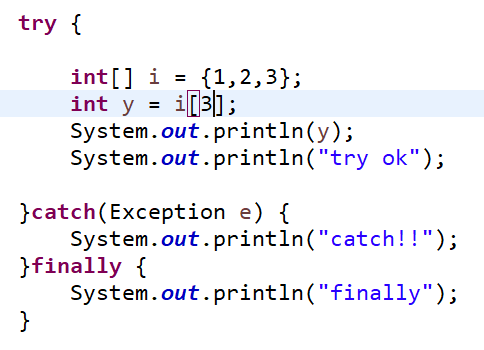




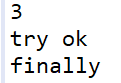
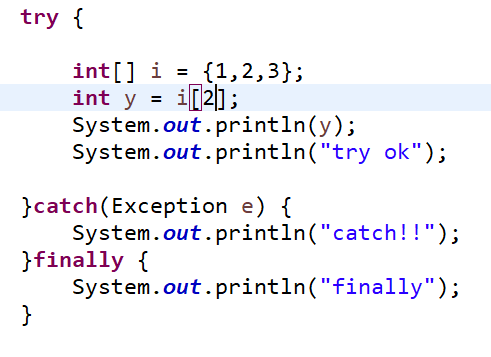
使用if/else或其它條件式區分錯誤情形，也是可以勉強處理到，但整段程式碼會相當冗長，維護起來也相較不易！



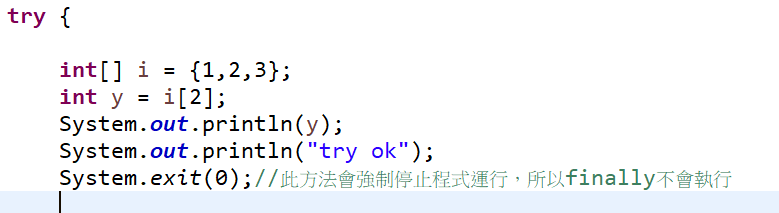
Ex：請問執行結果

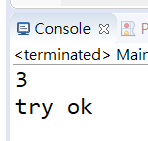


Ex: 請問執行結果

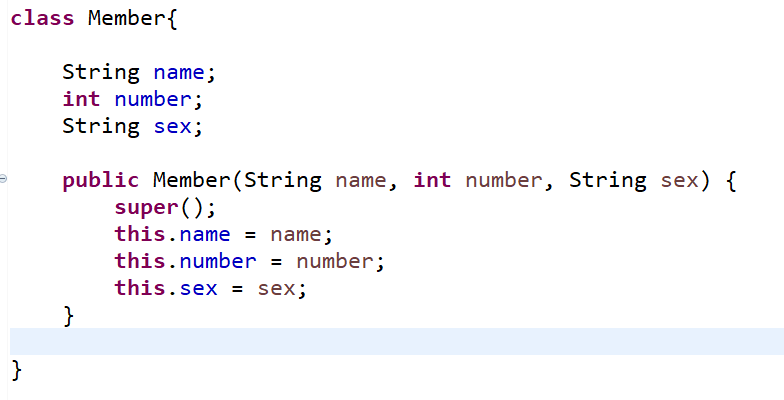


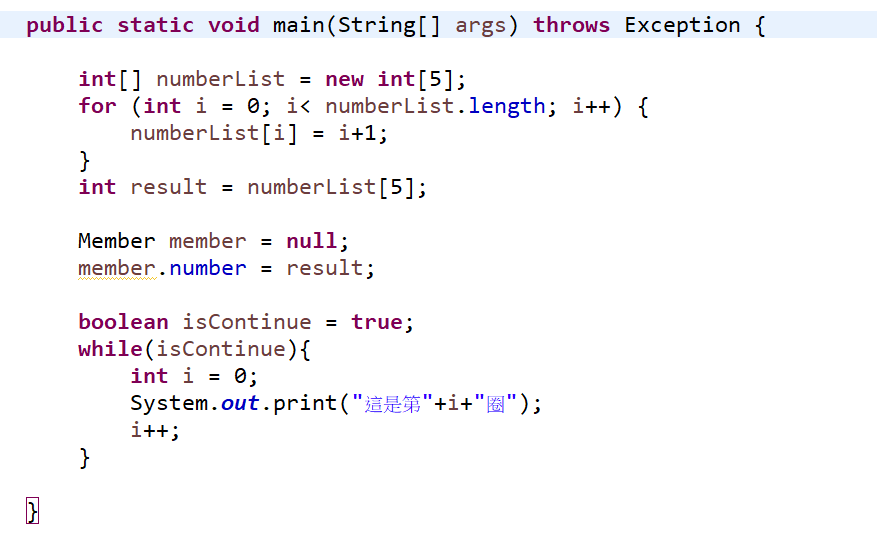
* 強制程式停止運行





練習：請找出以下程式有問題的地方，並寫出try/catch做後續處理，不能讓程式有crash現象





\*練習：試寫出一段錯誤的runtime exception程式碼，會產生crash的情形，試說明其錯誤原因

\*練習：承上，使用try/catch修復問題