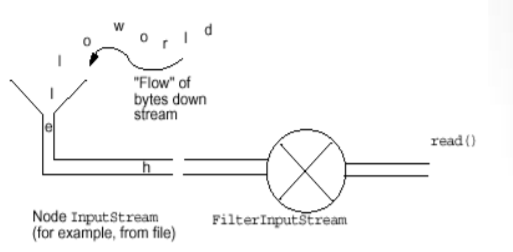
第六章 File I/O

1. 資料流(stream)的意義

<https://ethan-imagination.blogspot.com/2018/12/javase-gettingstarted-019.html>

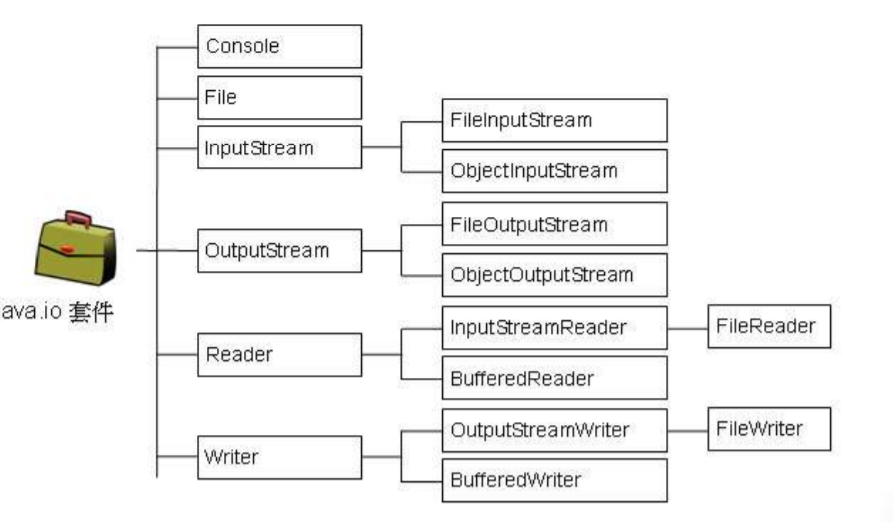
* Java中資料的讀取是以流的形式進行，有源頭和目的地，並且有固定方向



* 想要讀取文件內容，就好像往桶子裡插入一根水管，文件內容藉由這根管線將內容依序讀出，寫入時也是一樣，放入一根水管後開水，內容就尤如水流一樣寫進檔案內
* 輸入流還是輸出流？是以程式的角度來看的，用一根管道插入文件裡讀數據，視為輸入，寫入數據叫輸出
* Java在java.io套件中有定義了多個流的型態(類別或抽象類別都有)來實現輸入/輸出流內容，大致分成兩類：

|  |  |  |
| --- | --- | --- |
|  | 字節流 | 字符流 |
| 輸入流 | InputStream | Reader |
| 輸出流 | OutputStream | Writer |

1. 字節流(byte stream)：最原始的一種流，讀出來的數據就是0010101這種的表現形式，按照一個字8位元的字節來讀取
2. 字符流(character stream)：一個字符一個字符地往外讀取數據，一個字符為2個字節

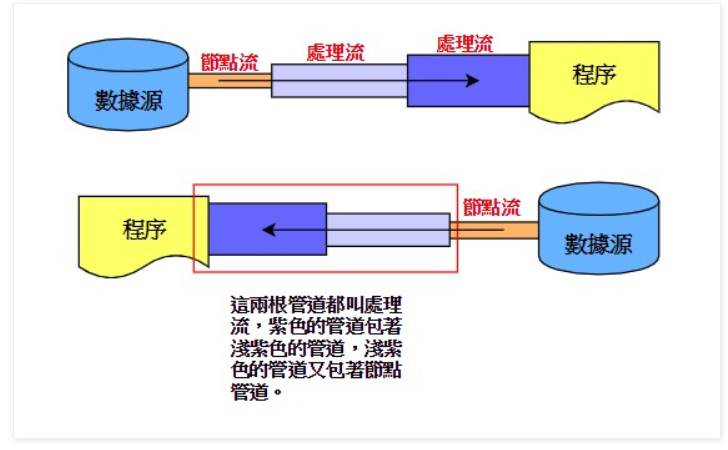


1. 節點流和處理流

* 節點流是從一個特定的數據源讀寫數據的管道

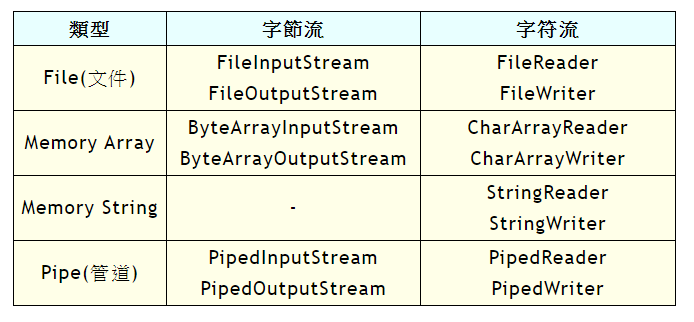


* 處理流是連接已存在的流中，提供更強大的數據處理



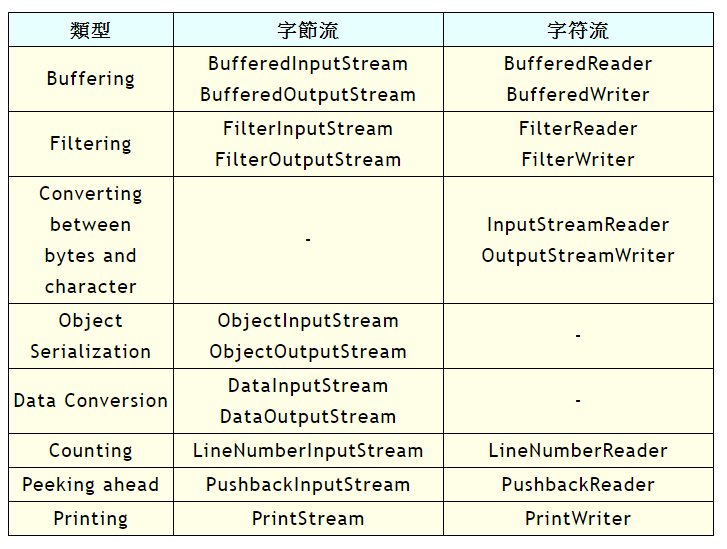
1. 節點流的類型

節點流就是有一根管道直接插到數據源上，直接讀取或是寫入數據源裡面的數據，最常見的有四種：  
FileInputStream、FileOutputStream、FileReader、FileWriter



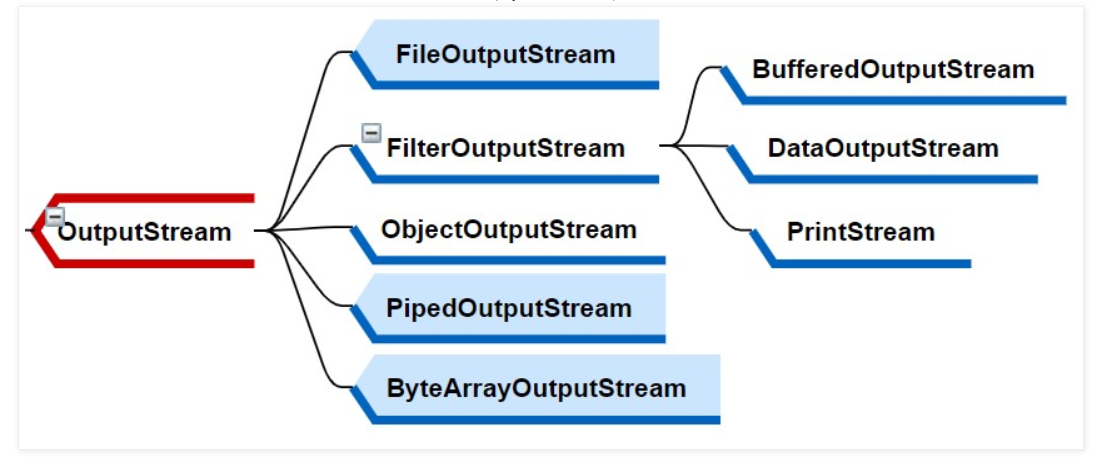
1. 處理流的類型

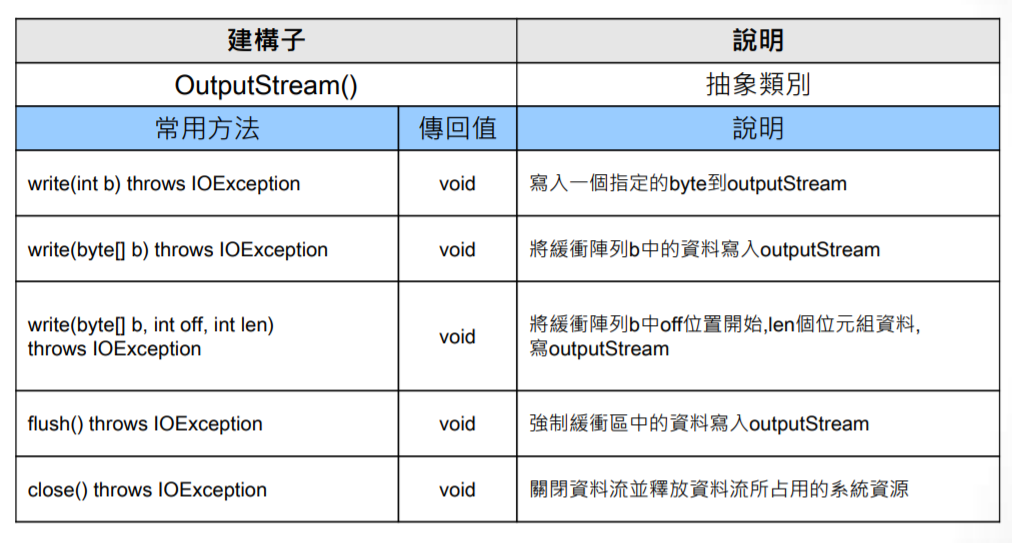
處理流是包在流上面的流，做為強化或加強緩衝效果的管道



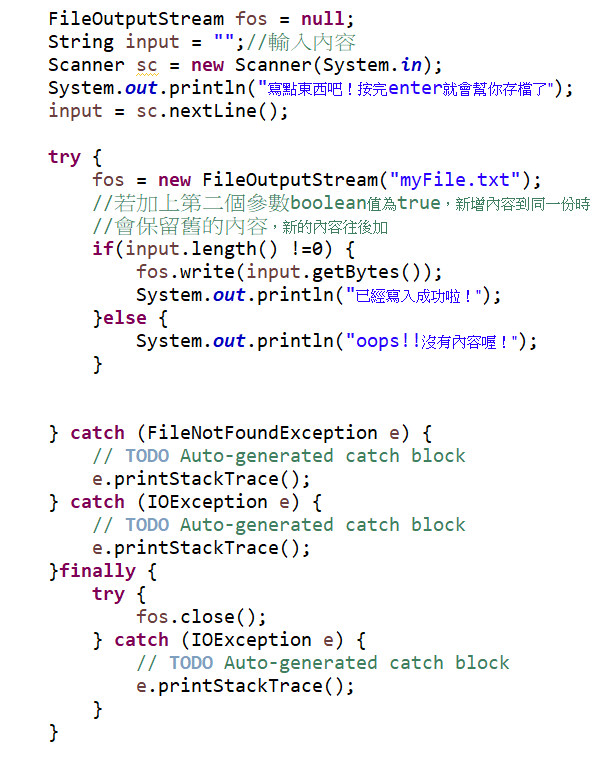
1. 輸出流(OutputStream)

* 作用在程式中用於輸出數據(到文件)，數據的字節為8位元，分為節點流(藍色背景)和處理流(白色背景)



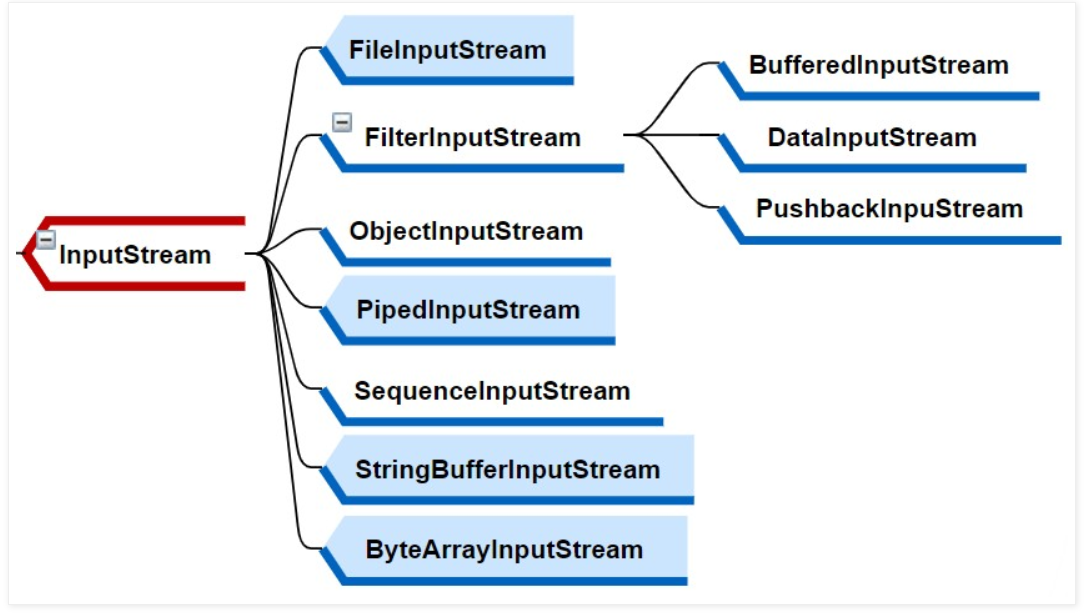


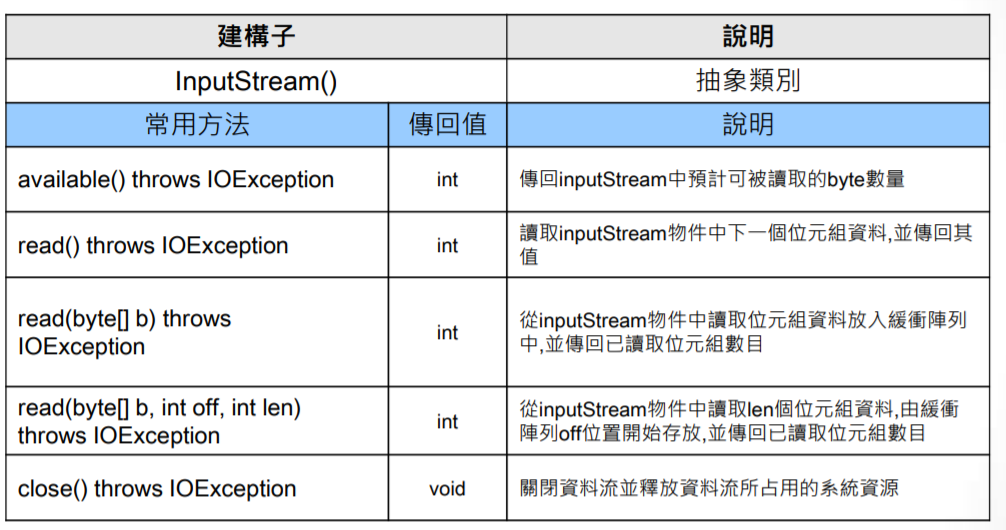
* FileOutputStream範例



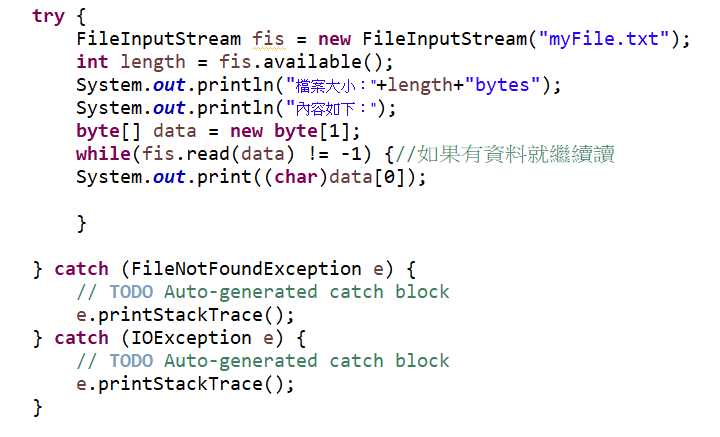
1. 輸入流(FileInputStream)

* 作用在程式中用於從數據源輸入數據，數據的字節為8位元，分為節點流(藍色背景)和處理流(白色背景)





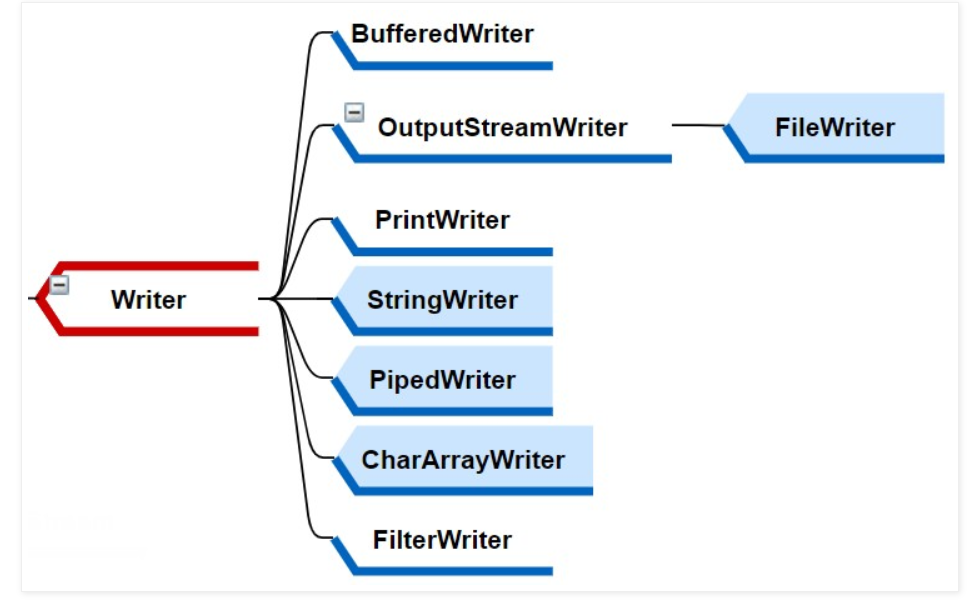
* FileInputStream範例

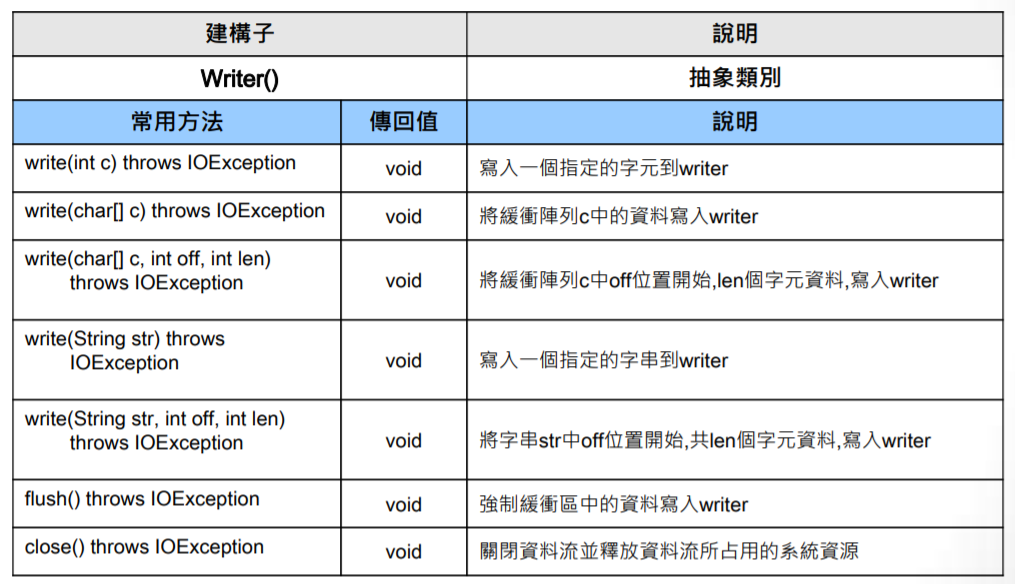


Ps：此種方法只能讀取位元組組成的內容(英文、數字等8位元資料)，中文的話會發現讀出一堆亂碼，是因為中文字是由兩個字節組成的(16字元)，所以會使用其它的方法來讀取。

1. 輸出流(Writer)

* 在程式中用於輸出數據(到文件)，數據的字節為16位元，分為節點流(藍色背景)和處理流(白色背景)



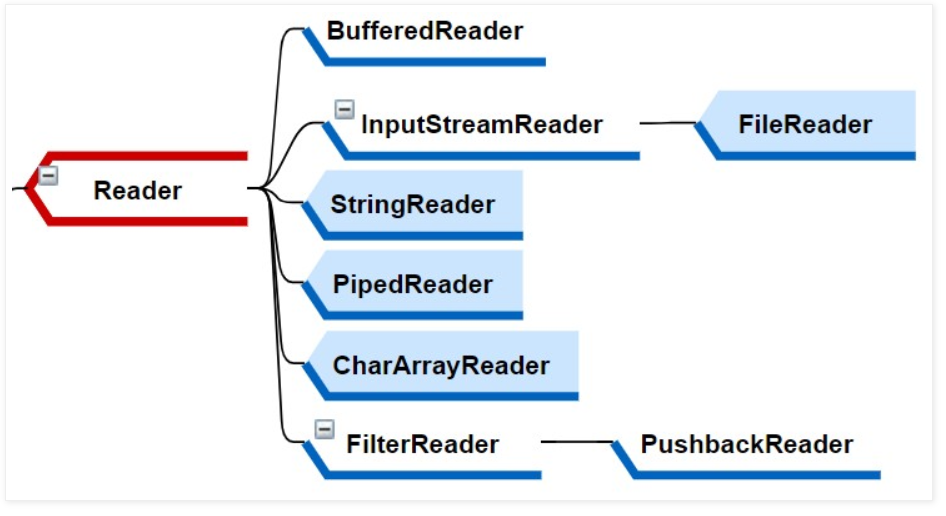


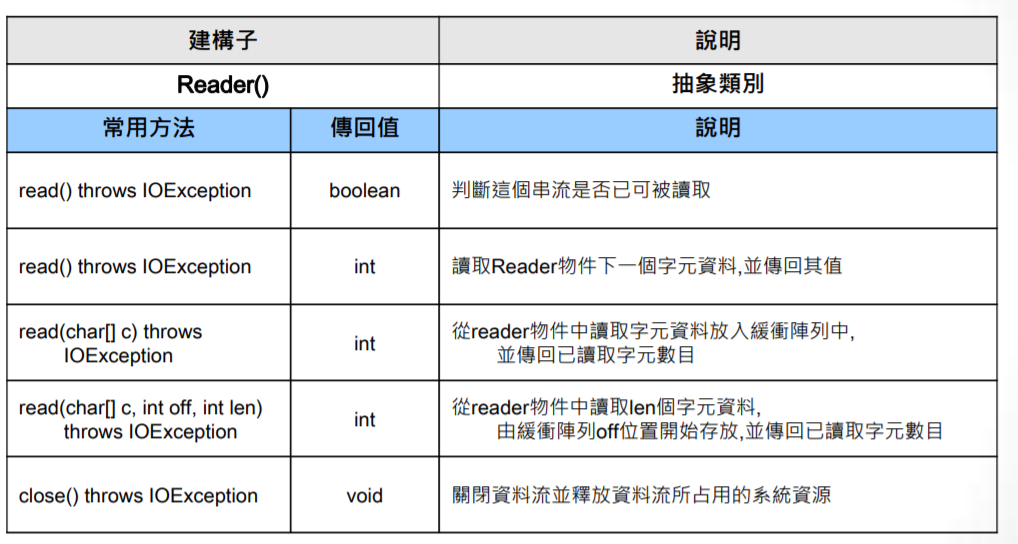
* FileWriter範例



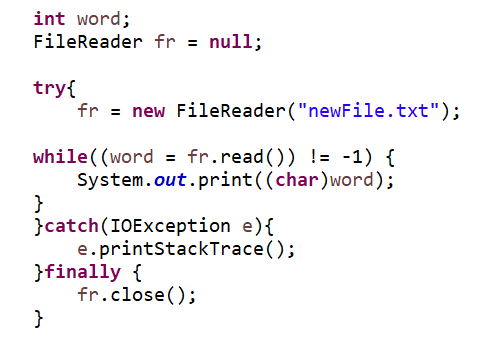
1. 輸入流(FileReader)

* 用於在程式中用於從數據源輸入數據，數據的字節為16位元，分為節點流(藍色背景)和處理流(白色背景)





* FileReader範例

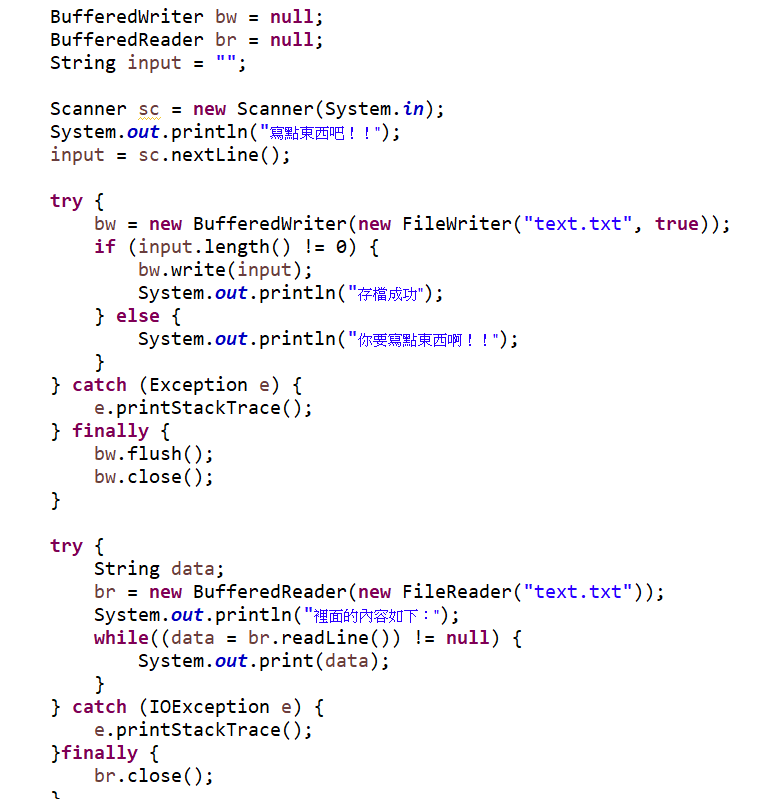


1. 緩衝流(Buffer)

* 緩衝流的目的是為了提高讀寫的效率，並套接在相應的節點流上，常用的有四種緩衝流：

1. BufferedReader
2. BufferedWriter
3. BufferedInputStream
4. BufferedOutputStream

* 帶有緩衝區的流，其緩衝區就是內存裡的一塊區域，讀寫數據時會先把數據放到這塊區域，裝滿之後再做處理，減少io對硬盤的訪問次數，增加效率並保護硬體護備。
* BufferedReader & BufferedWriter的範例



1. 轉換流(StreamReader/Writer)

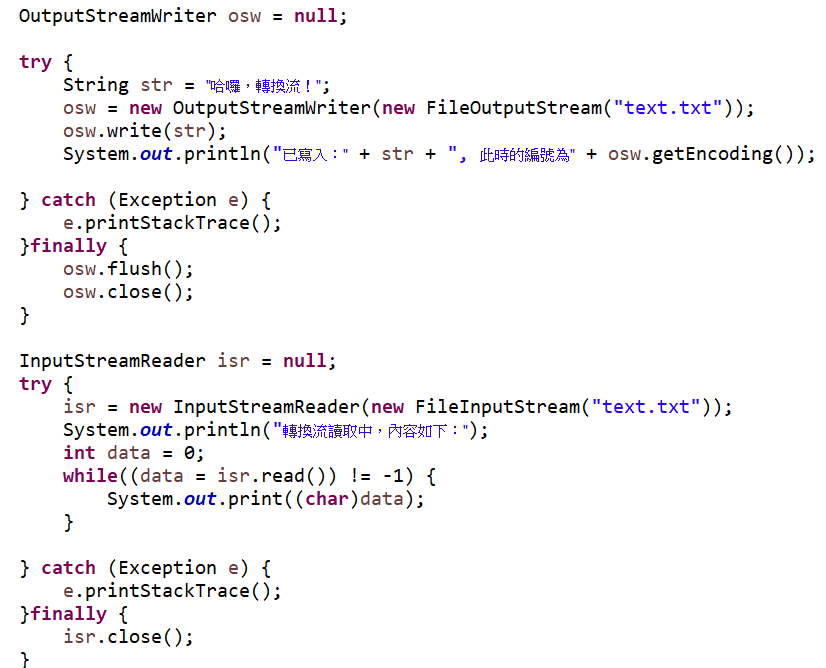
* 轉換流的意思就是它可以把字節流轉換成字符流，常見的有兩種：

1. InputStreamReader
2. OutputStreamWriter

* InputStream轉換成Reader, OutputStream轉換成Writer的技術
* 可以將內容以字符的方式通過管道寫入或讀出數據，並且指定編碼，如下圖所示：

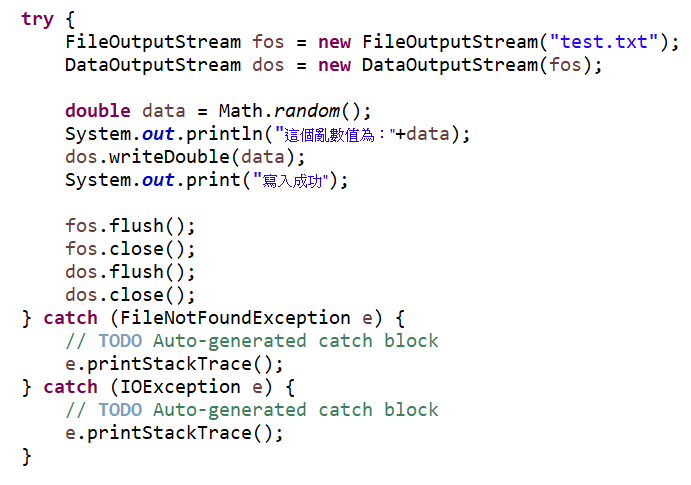


轉換流搭配stream的範例

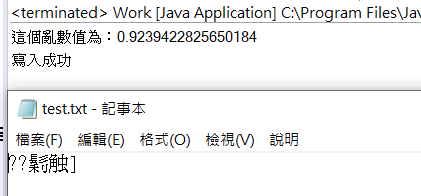


1. 數據流(DataInputStream/DataOutputStream)

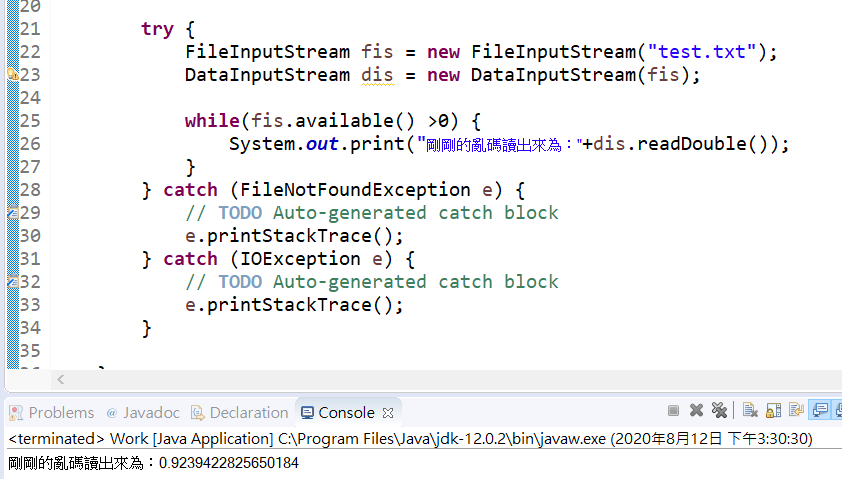
* 屬於處理流，需要分別接在InputStream 和 OutputStream的節點流上
* 提供可存取原始類型數據的方法(ex: int, double這種)
* DataOutputStream的範例



文件結果是亂碼



* DataInputStream範例

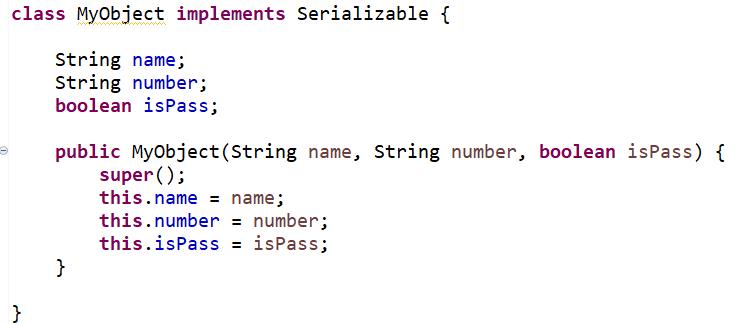


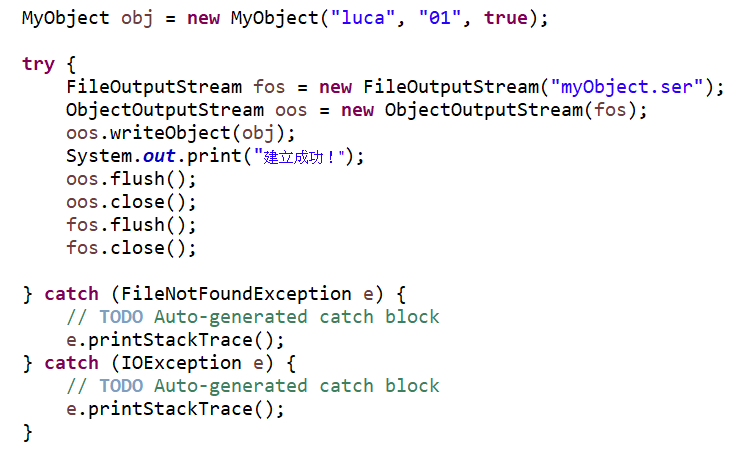
1. 物件流(Object)

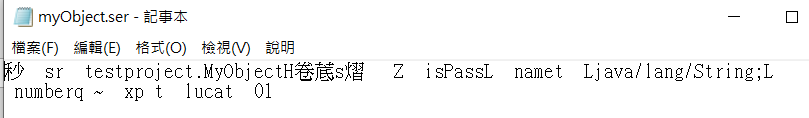
* 可直接將物件寫入或讀取的技巧
* 與永續化(persistence)的概念有關
* 使用序列化(Serializable)完成物件的存儲

1. 序列化(Serialization)與反序列化(Deserialization)

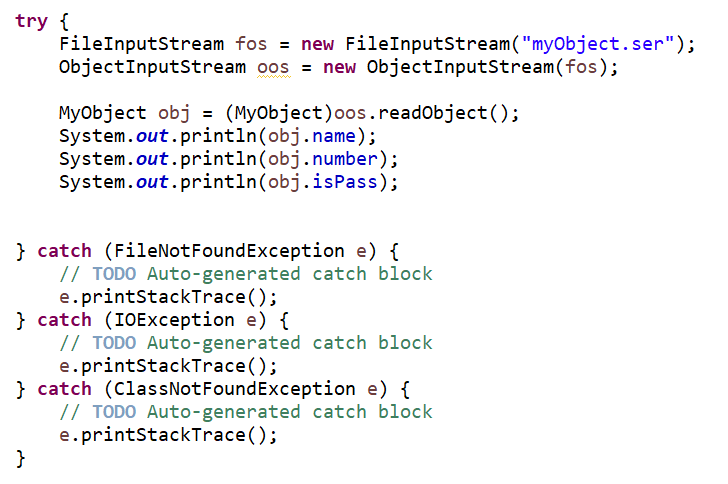
* 序列化功能是把物件變成資料流，將物件儲存成檔案並永久儲存在礎碟中，並且互相傳遞
* 反序列化就是將檔案還原成物件後使用
* 物件序列化基本用法

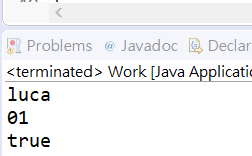






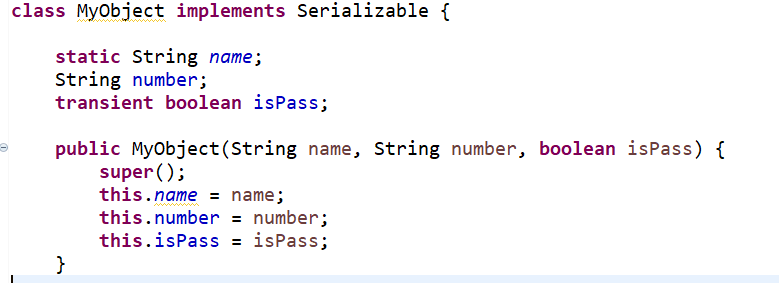
* 物件反序列化基本用法

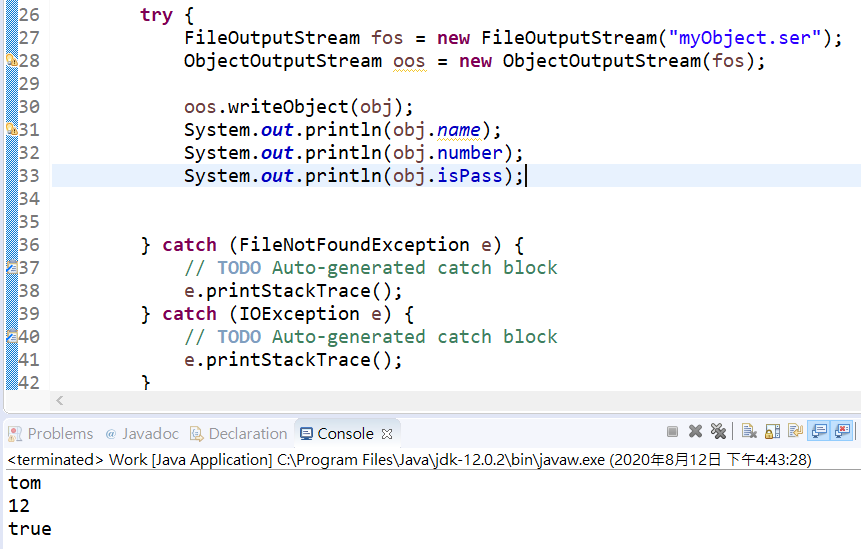


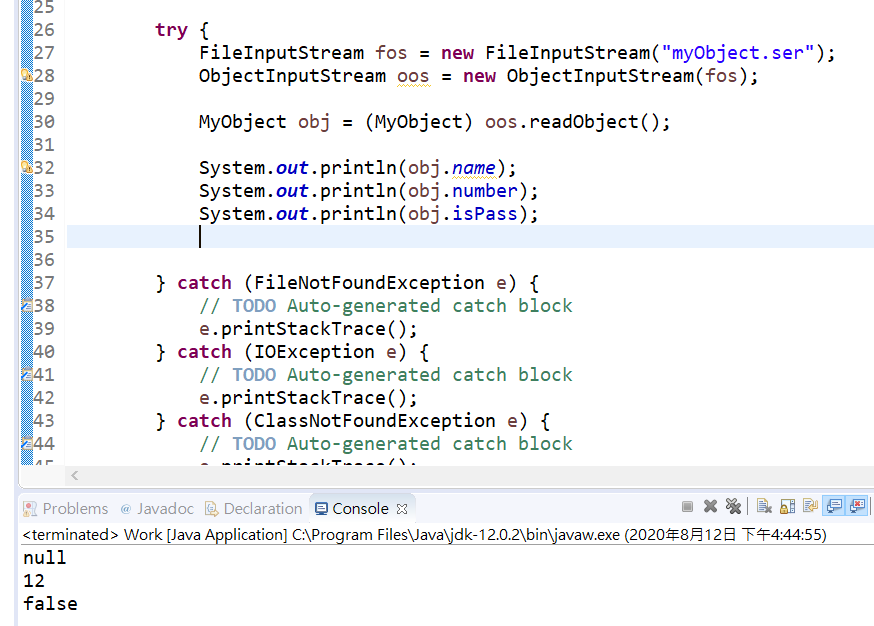
剛剛序列化後存起來的東西重現了！

1. 類別屬性與transient關鍵字

* 物件當中被冠上static或下transient關鍵字後，序列化過程即會被忽略







冠了static或transient的屬性在序列化時沒有被存儲下來，反序列化後看到的是null或是該屬性的預設值。

十一、其它補充

1. SourceTree 與 git

* Git是一個輕量級的版本控制系統，軟體開發有分成很多不同的階段，龐大的工作量使得一個專案的開發通常由許多人分工合作，而如何將每個人的成品整合成一個完整的程式碼就是很常要處理的問題



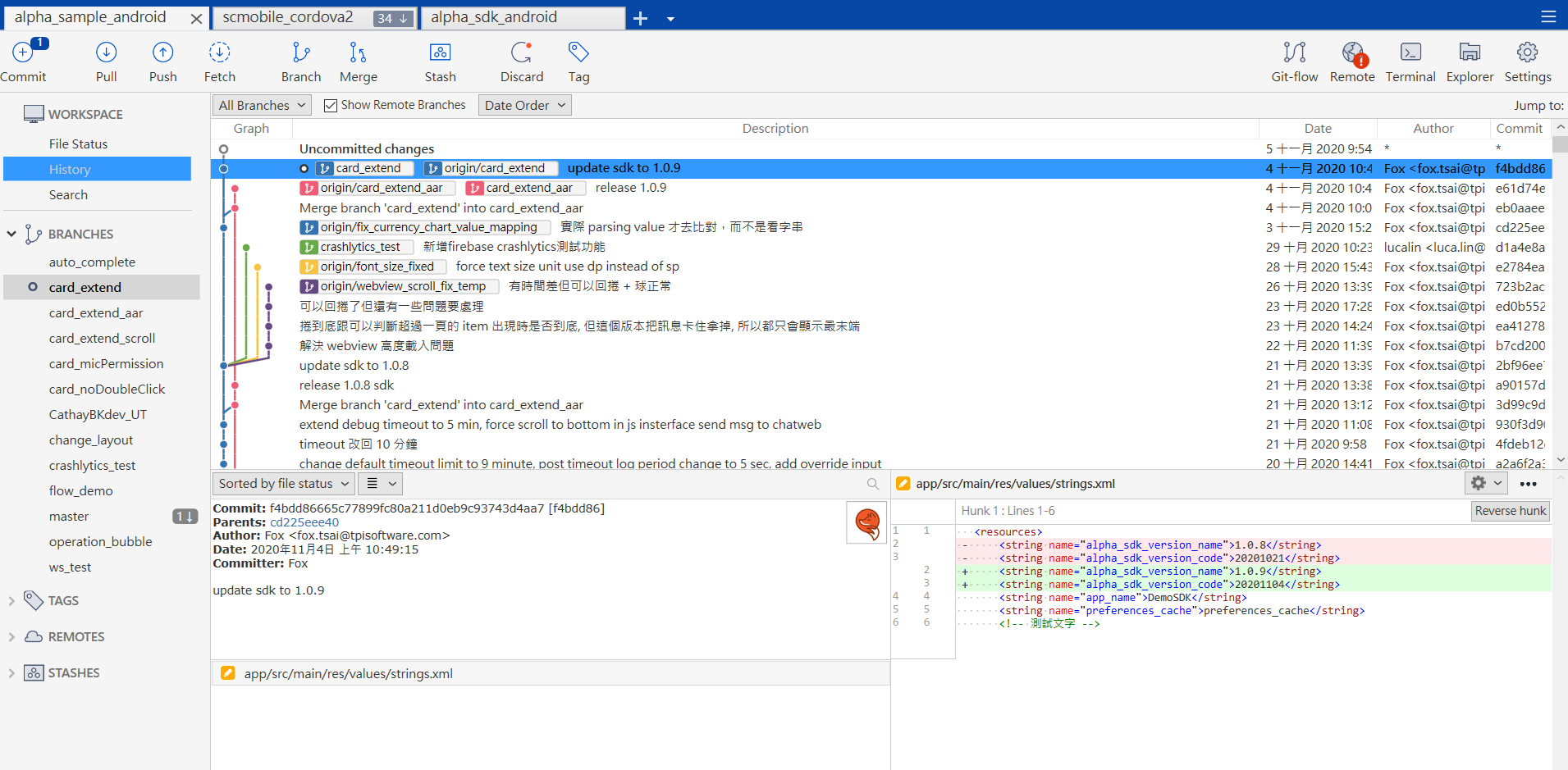
* 使用git技術的地方很多，Linux Kernel, Apache, Eclipse, Perl, PHP, Node.js, jQuery, 等等。還有諸如 Google, facebook, Microsoft, Twitter, Linkedin, NetFlix也都是使用git在管理專案版本
* 以git為核心的雲端平台也很多，最常用的如gitlab、bitbucket、github都是
* Git安裝與簡單操作的教學：<https://tw.alphacamp.co/blog/git-github-version-control-guide>
* 版本控制的好處：

1. 清楚記錄修改前與修改後的程式碼
2. 修改的人是誰
3. 修改的時間點為何
4. 修改的原因
5. 所有人皆可遠端操作
6. 可回溯至前面的版本
7. SourceTree

* sourceTree是一個圖形化介面的版本控制軟體，與git指令結合，提供用戶更簡單操作版本的工具

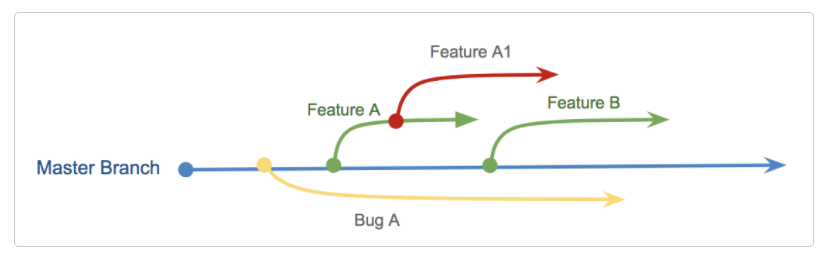


使用介面範例



1. 專案頁籤
2. 版本線操作
3. 分支清單
4. 圖形化專案進度線、時間點與作者
5. 記錄日誌
6. 程式碼修改記錄

* Branch與merge



* 以這張圖來看，專案的主要任務線為master(2020已改為main)，其餘即為任務分支(branch)，至於為何要開branch就因專案而異了，有時是為了測試某功能是否正常，但並沒有要把該程式碼寫進專案中。有時是附加項目，待測試成功後才會加進主線的。有時是純粹的poc，往後並沒有要併入主線。
* 將分支(branch)併入主線的行為就叫合併(merge)



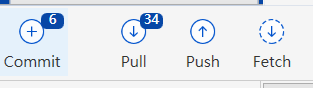
Merge痕跡

各個分支

主線

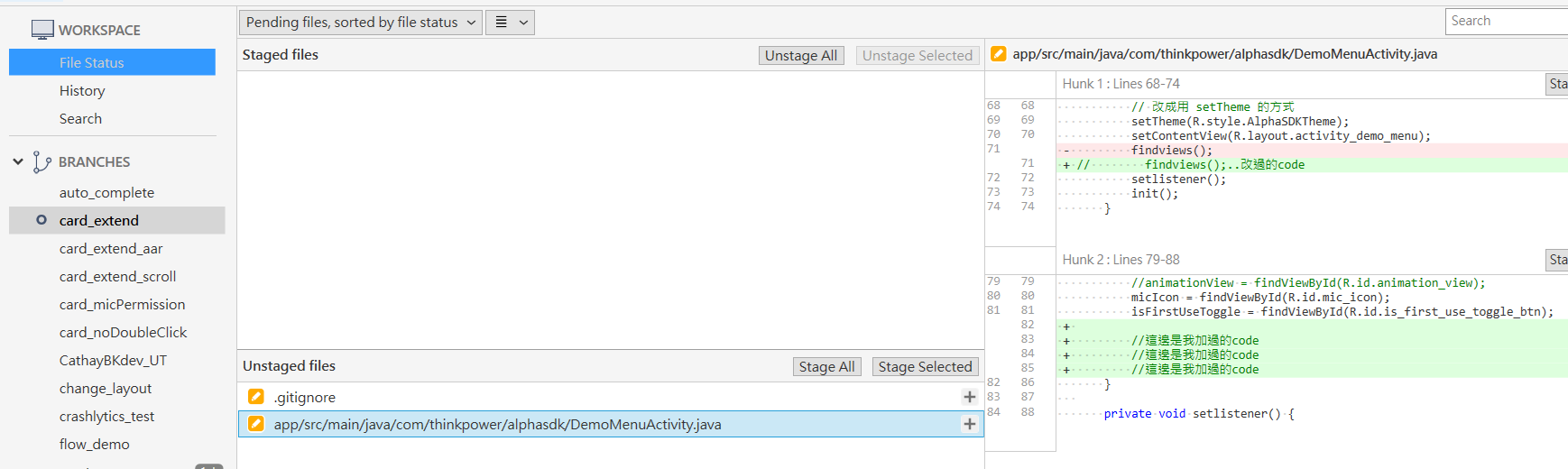
* Fetch

在專案進行中，可能不時會有其他同仁已寫好並上傳，這時fetch就是一個同步的操作，確保目前手上拿到的程式碼是更新過後的

有更新記錄次數的話會顯示在pull按鈕上

* Pull & push
* Pull為將自己手上的程式碼更新到別人寫過並上傳過後的程式碼，在pull前要先確定自己手上的碼並沒有更動過的記錄才行

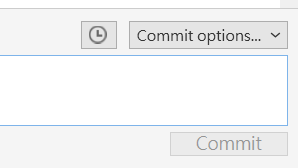
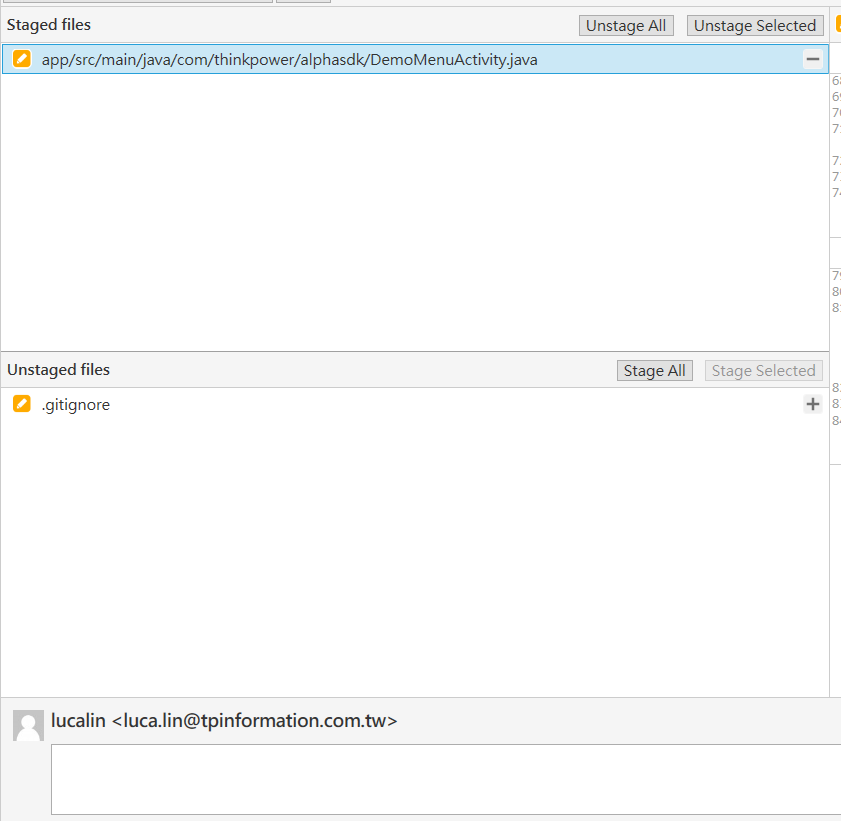
* Push為你要將寫好的程式碼上傳至專案線上的動作，在filestatus欄中可以看到你的更動記錄，並可選擇要上傳哪些新的程式碼



你的更動記錄

選擇要推哪些檔，或是全推

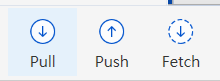
改過的所有檔案會在這



設定好之後即可點選commit

這裡可以寫註解

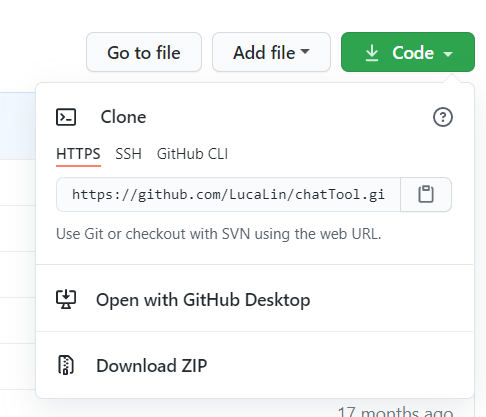
選擇要推上去的檔案



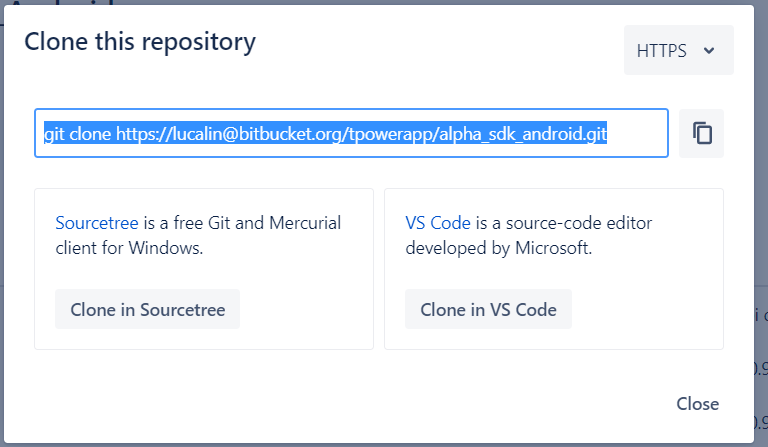
有commit記錄的話在push就可以看到等待推上任務線的通知，點選puch後才算是真正的推上程式碼！

* 從雲端clone專案到本地

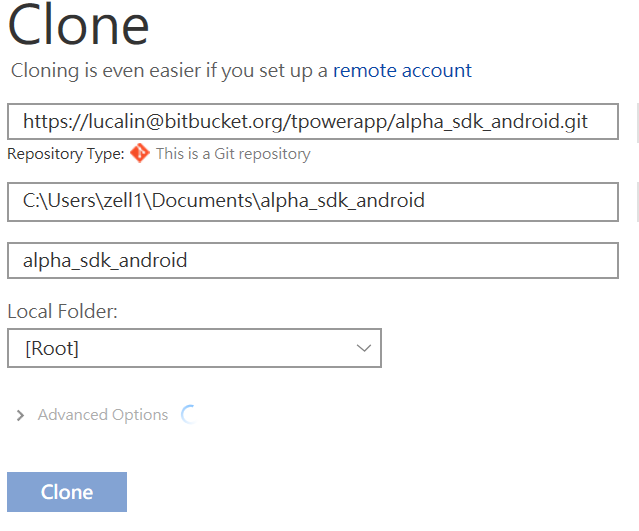
1. 從github



1. Bitbucket



* Clone url copy到surcetree



* 網址正確的話會顯示正確，選好置放路徑後按下clone，稍待3-5分即可，clone時間取決於專案的大小