



物流系統

黃培堯 | 106361606

張凱傑 | 106361609

鄭勝中 | 106361624(組長)

# 動機與目的

因為目前公司使用的物流系統使用了一些功能常常出現問題，

主要的缺點是：

1. 在查詢資料的時候速度太慢
2. 有時候進貨時會發生資料錯誤的情況發生

這些問題間接導致了事後都要人為直接進行資料更改，所以我們想要嘗試自己做一個簡易的物流系統。

主要就是模擬跟公司差不多的狀況，並且嘗試做出一些修正。

我們希望我們自行編寫的系統能建立廠商基本資料及其商品資訊，正常讀取搜尋資料、進出貨系統、盤點系統、多倉庫管理，及簡易的權限設置。

# 預計完成功能

1. 以廠商資料及商品資訊建立資料庫
2. 讀取並搜尋指令資料
3. 進出貨系統(存入、取出功能)
4. 盤點功能(類似整體資料庫當前狀態功能)
5. 多倉庫管理(多資料庫sync功能)
6. 此系統的功能使用權限設定

# 預計使用語言、平台、開發環境

預計使用Java，在PC上進行，使用Eclipse IDE開發

# 預計使用資料結構、演算法、其程式碼來源

線性資料結構：

1. SortList 利用： <https://www.geeksforgeeks.org/collections-sort-java-examples/>

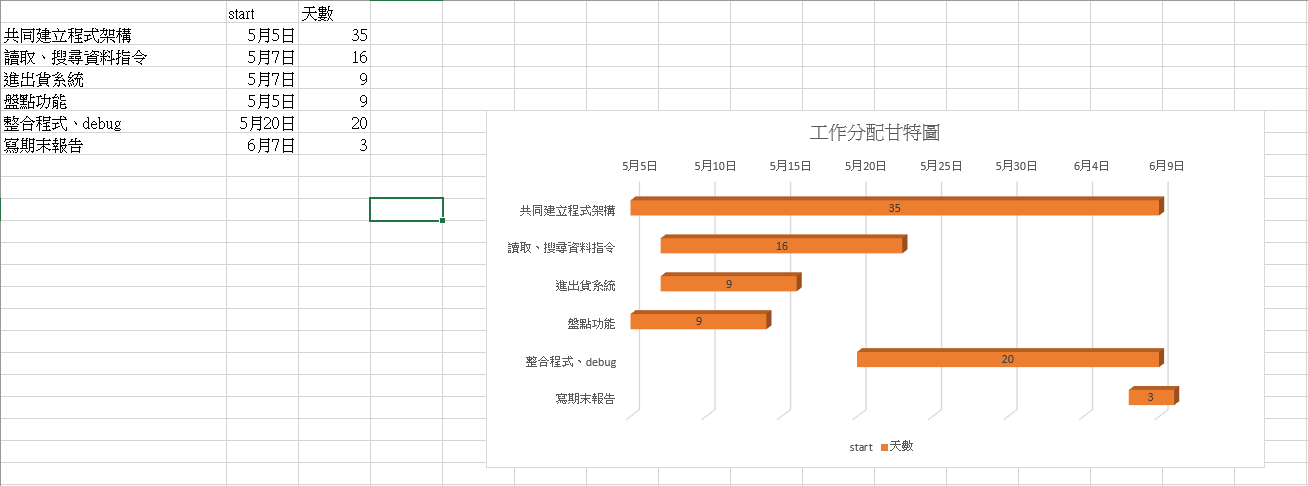
非線性資料結構：

1. HashMap利用：<https://codertw.com/android-%E9%96%8B%E7%99%BC/185337/#outline__1_0_1>

演算法：

1. Sort 內含在Linked List 裡面

# 工作排程甘特圖

實際分配狀況

# 實際完成功能

1. 以廠商資料及商品資訊建立資料庫 (O)
2. 讀取並搜尋指令資料 (O)
3. 進出貨系統(存入、取出功能) (O)
4. 盤點功能(類似整體資料庫當前狀態功能) (O)
5. 多倉庫管理(多資料庫sync功能) (X)

未完成原因 : 來不及趕工完成

1. 此系統的功能使用權限設定 (X)

未完成原因 : 來不及趕工完成

# 實際使用語言、平台、開發環境

使用Python3.7，在PC上進行，使用Spyder IDE開發

更改程式語言的原因 : 當初雖然說是想要使用java 後來發現有部分的組員沒有很熟悉，後來在多方的討論下我們最終改用大家比較熟悉的語言python來做此專案

# 實際使用資料結構、演算法、其程式碼來源

線性資料結構：

* DataFrame，參考資料：<https://oranwind.org/python-pandas-ji-chu-jiao-xue/>

更改原因 : 因為整個語言都更改了所以我們重新上網找有關python 的資料結構，後來發現此線性資料結構，用來輸出檔案很方便，所以我們就將它納入我們的程式中。

非線性資料結構：

* AVL Tree，參考資料：<https://gist.github.com/girish3/a8e3931154af4da89995>

更改原因 : 因為在老師上課時聽到此資料結構，發現我們的架構是用廠商編號 及 商品條碼 來輸入資料，剛好都是唯一碼，而且在讀取資料時也不是按照編號讀取所以非常適合用到我們的專題中。

演算法：

* Search Binary Tree，包含在AVL Tree裡面

更改原因 : 因為用到AVL tree ，然後裡面剛好有 Search Binary Tree 所以我們就直接作使用。

* SortList，包含在AVL Tree裡面

# 程式的 github 連結

我們這組後期利用github 來做管理，剛好github有一些我們的更改過程  
程式聯結 : <https://github.com/0obriano0/Final_Period_DataStructure?fbclid=IwAR3-sljGus6sAd1xxnkvE_DgayNStjw1Bd4xdOCr9efYgITOH7ZPZhnAhGc>

# 程式開啟方法

1. 安裝Anaconda Navigator

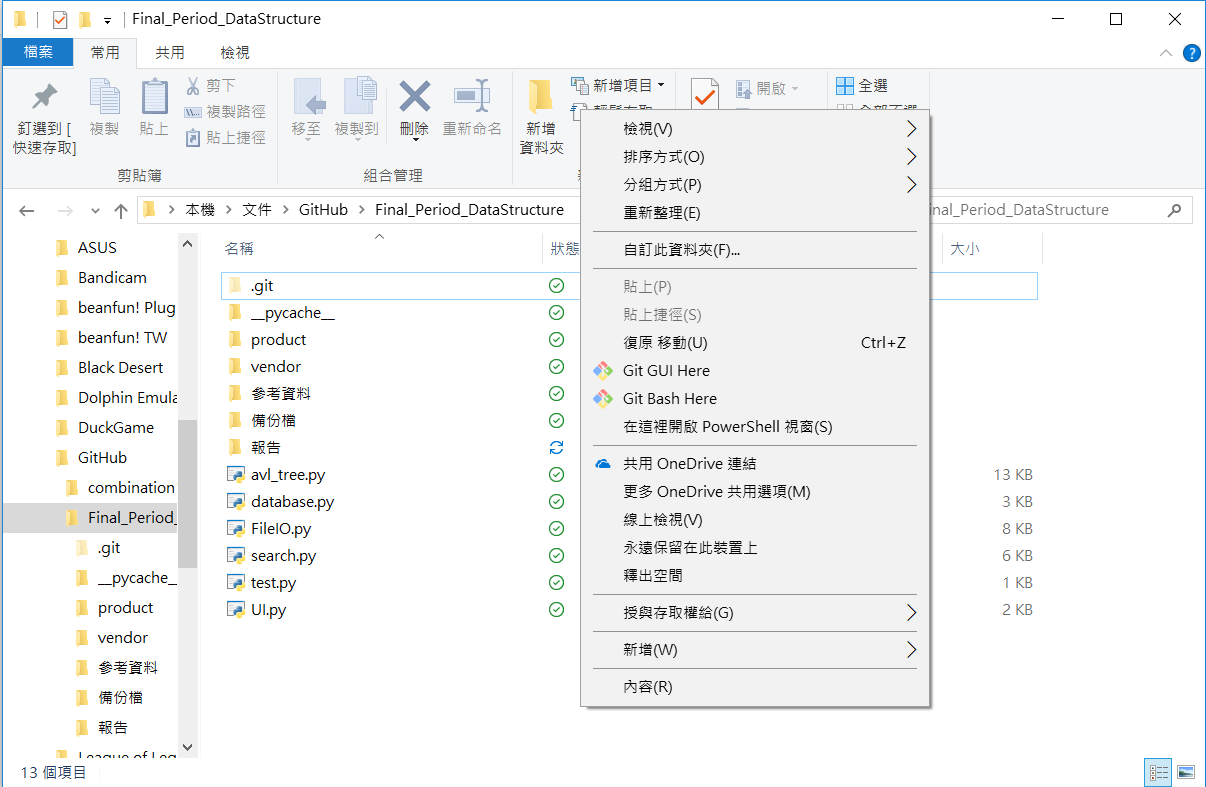
安裝時請勾選設定環境變數

安裝教學範例 :

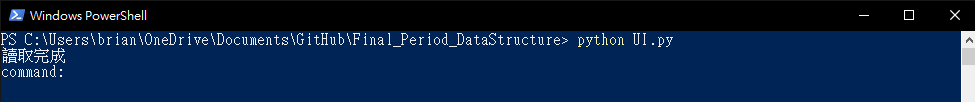
<http://xken831.pixnet.net/blog/post/463042472-%5Bpython%5D%5Banaconda%5D-anaconda-%E5%AE%89%E8%A3%9D%E6%96%B9%E5%BC%8F>

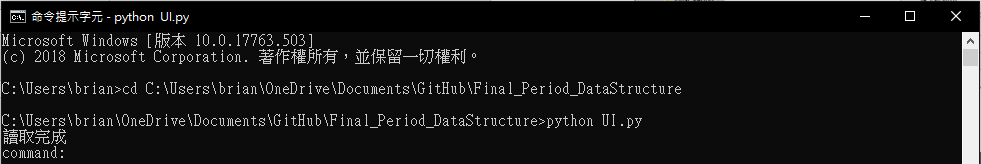
1. 在程式資料夾內空白處按住Shift+點擊滑鼠右鍵

選擇”在這裡開啟PowerShell視窗”



1. 在PowerShell或是 cmd視窗內輸入"python UI.py"即可運行程式





# 程式使用說明

* Select指令 : 可以在設定搜尋條件下去找到所期望回傳的資料屬性

|  |  |
| --- | --- |
| 範例 : select "RN","principle" from "vendor" where "name"="廠商一" | |
| 綠色引數 | 選擇要搜尋資料的是商品(product)或廠商(vendor) |
| 藍色引數 | 設定要搜尋資料的條件。多個條件設定請用 , 隔開，範例如下:  Select "SN" from "product" where "name"="1","number"="16"  程式會去搜尋name這個屬性為1 或 16 的商品出來 |
| 紅色引數 | 期望程式回傳的資料屬性 。多個期望屬性請用 , 隔開，範例如下:  Select "RN","principle" from "vendor" where "name"="廠商"  程式會去搜尋name是「廠商」的廠商資料，並回傳所有找到的資料的RN 跟principle回來。 |

下方是所有支援Select指令的引數名稱 :

|  |  |  |
| --- | --- | --- |
| 資料種類 (綠色引數) | vendor(廠商) | product(商品) |
| 引數名稱 (紅色引數、 藍色引數) | name : 名稱  RN : 公司註冊序號  principle : 負責人  address : 地址  product : 商品 | name : 商品名稱  SN : 序號 number : 流水號 warranty : 保固期 volume : 體積 weight : 重量 category : 種類  quantity : 數量 |

* post指令 : 將進貨的商品存入

post與 pop指令使用相同的格式，範例如下 :

post "vendor"="廠商", "name"="名稱","quantity"="數量","SN"="序號"

|  |  |
| --- | --- |
| 廠商 | 欲進貨貨物的來源廠商名稱 |
| 名稱 | 欲進貨貨物的貨物名稱 |
| 數量 | 欲進貨貨物的貨物數量 |
| 序號 | 欲進貨貨物的貨物序號 |

* pop指令 : 將欲提貨的商品提出

pop與post指令使用相同的格式，範例如下 :

pop "vendor"="廠商", "name"="名稱", "quantity"="數量", "SN"="序號"

|  |  |
| --- | --- |
| 廠商 | 欲提貨貨物的來源廠商名稱 |
| 名稱 | 欲提貨貨物的貨物名稱 |
| 數量 | 欲提貨貨物的貨物數量 |
| 序號 | 欲提貨貨物的貨物序號 |

# 心得與感想

* 張凱傑 :

這次原本組長想要我們使用java製作一個物流管理系統，計畫都打好了結果發現就只有組長一個人會用java，其實我有在打計畫前就提醒他，但他執意要這麼做，最後索性使用python來完成後續作業，真是可哀可嘆呀

* 黃培堯:

透過這次的專題，我算是對class有更多的了解了。雖然說在頂層與底層之間作溝通會常常搞混要把資料給誰，但是其實組員間溝通好的話，編成速度其實可以很清楚明瞭且迅速的。在最後全部的檔案都導入以後成功的在頂層把程式跑起來的那一瞬間真的是很有成就感。

* 鄭勝中:

這次在製作程式我負責了大部分的 debug 、 檔案存取系統 及 AVL Tree 修改，還負責整個架構的建構。第一次嘗試將檔案分成三份給三個人作編寫，其中我為了講解一個class 花了我蠻多力氣的，我們最後還是有將程式基本功能完成，雖然中間有小小得爭吵，但是還是有感到有成就感的。