

國立高雄科技大學

智慧商務系

資料結構

期末專題

程式重構

指導教授：謝文川 教授

學生：智慧商務系二乙 黃綉蓮

中華民國 111 年 12 月

目錄

一、研究動機及目的.....	2
二、程式重構與實作.....	2
三、運用排序法新增功能.....	3

圖目錄

圖 1 原始程式碼	2
圖 2 重構後程式碼	3
圖 3 Todolist Systemn. 搜尋功能畫面	3
圖 4 快速排序法程式碼	4

一、研究動機及目的

回頭看自己一年級在程式設計課堂做的專題，程式的構建雜亂無章，也發現能夠用更加簡潔的程式碼來實現功能，因此想運用這次在課堂中學習的思維及機會，將之前凌亂的程式重新構建，使程式更加容易閱讀及執行。

另外，在這次的資料結構中學習到排序法，能夠應用於以往在撰寫前端網頁時想實現的搜尋結果排序功能，因此也在其他專題中增加這項功能。

二、程式重構與實作

在一年級程式設計的期末專題中發現之前撰寫功能的方式過於繁瑣，有許多功能可用一行程式就實現，卻用了很多步驟才完成。且功能都寫在一起沒有好好地拆分區塊，程式的架構不清楚。

```
1  #%%
2  import requests
3  import json
4  import pandas as pd
5  import pymysql
6  from sqlalchemy import create_engine
7  import matplotlib.pyplot as plt
8
9  font={'family':'DFKai-SB',
10 'weight':'bold',
11 'size':12}
12 plt.rc('font',**font)
13 plt.rc('axes',unicode_minus=False)
14
15 engine=create_engine('mysql+pymysql://C109156234:mis@192.168.23.137:3306/tour?charset=utf8')
16
17 response=requests.get("https://data.gov.tw/qc_download/dq_download_json.php?nid=8116&nd5_url=682b84de87067b3770f0831e93d9e3c4")
18 s=json.loads(response.content)
19 df=pd.DataFrame.from_dict(s)
20 df.to_sql(name='totaldata',con=engine,if_exists='replace',index=False) #將抓取的資料放進資料庫
21
22 ##取出2015-2019每年總旅次##
23 year=[]
24 totaltrip=[]
25 for i in range(int(df.loc[0,'年別']),int(df.loc[len(a)-1,'年別')+1,1):
26     year.append(str(i))
27     tt=df[df['年別'].isin([str(i)])] #以年份抓取每年的資料
28     tt.to_sql(name='total%s'%str(i),con=engine,if_exists='replace',index=False) #將每年的資料放進資料庫
29     t1=tt[['年別','合計']]
30     ttrip=0
31     for j in t1['合計']:
32         ttrip+=int(j)
33     totaltrip.append(ttrip)
34 tripperyear=pd.DataFrame.from_dict({'年別':year,'總旅次':totaltrip})
35 tripperyear.to_sql(name='totalperyear',con=engine,if_exists='replace',index=False)
36 tripperyear=tripperyear.set_index('年別')
37 tripperyear.plot(title='歷年國內觀光總旅次',figsize=[15,10]) #繪製折線圖
38
39 ##每年各縣市旅次、每年各景點遊客人數##
40 conn=pymysql.connect(host='192.168.23.137',port=3306,user='C109156234',passwd='mis',db='tour',charset='utf8')
41 cursor=conn.cursor(pymysql.cursors.DictCursor) #開回游標是準備的資料
42 for u in range(int(df.loc[0,'年別']),int(df.loc[len(a)-1,'年別')+1,1):
43     cursor.execute("SELECT * FROM 'total%s'%"%(str(u)))
44     rows=cursor.fetchall()
45
46 perdata=pd.DataFrame.from_dict(rows)
47 for i in range(len(perdata)):
48     perdata.loc[i,'合計']=int(perdata.loc[i,'合計'])
49
50 c1=perdata.縣市別.unique() #取出縣市別欄所有出現的數
51 reval=[]
52 for i in c1:
53     region=perdata[perdata["縣市別"]==i]
54     c2=0
55     for i in region['合計']:
56         c2+=i
57     reval.append(c2)
58 regiontrip=pd.DataFrame.from_dict({'縣市別':c1,'總旅次':reval})
59 regiontrip.to_sql(name='regiontrip',con=engine,if_exists='replace',index=False)
60 regiontrip.plot.bar(title='各縣市旅次',figsize=[15,10]) #繪製折線圖
61
62 attraction=perdata[['區分','合計']]
63 new_att=attraction.sort_values(['合計'],ascending=False) #區域排序
64 h10_att=new_att.head(10) #取前十筆資料
65 print("%s最多遊客人數景點--取前10名\n\n"%(str(u)),h10_att)
66 print("=====\n")
67 conn.commit()
68 cursor.close()
69 conn.close()
70
```

圖 1 原始程式碼

重構後的專題運用物件的方式實現程式碼，並將所有功能都用一個方法拆分，程式碼變得更加簡潔。

```
final_refactor.py > travel > _init_
1 import requests
2 import json
3 import pandas as pd
4
5 class travel:
6     def __init__(self) -> None:
7         response=requests.get("https://quality.data.gov.tw/dq_download_json.php?nid=8116&md5_ur1=6c2ab89ff50cab9b9348154384613944")
8         df=json.loads(response.content)
9
10        self.df=pd.DataFrame.from_dict(df)
11
12    # 每年總旅次
13    def getTotalTripsPerYear(self):
14        return self.df[['年別', '合計']].groupby('年別').sum()
15
16    #指定年度各縣市旅次
17    def getCityTotalTrips(self,year):
18        currentYearData=self.df[self.df['年別']==str(year)]
19        return currentYearData[['縣市別', '合計']].groupby('縣市別').sum()
20
21    #每年各景點旅遊人數
22    def getTouristSpot(self,year,spot:str=""):
23        currentYearData=self.df[self.df['年別']==str(year)]
24        returnData=currentYearData[['細分', '合計']]
25        if spot=="":
26            return returnData
27        else:
28            return returnData[returnData['細分']==spot]
29
30    obj = travel()
31    # print(obj.df)
32    print(obj.getTouristSpot(2012,"陽明山遊客中心"))
33    # print(obj.getTotalTripsPerYear())
```

圖 2 重構後程式碼

三、運用排序法新增功能

將快速排序法運用在個人代辦系統的 Rract 專題中，可讓搜尋出來的結果根據代辦清單的標題的第一個字母做排序。

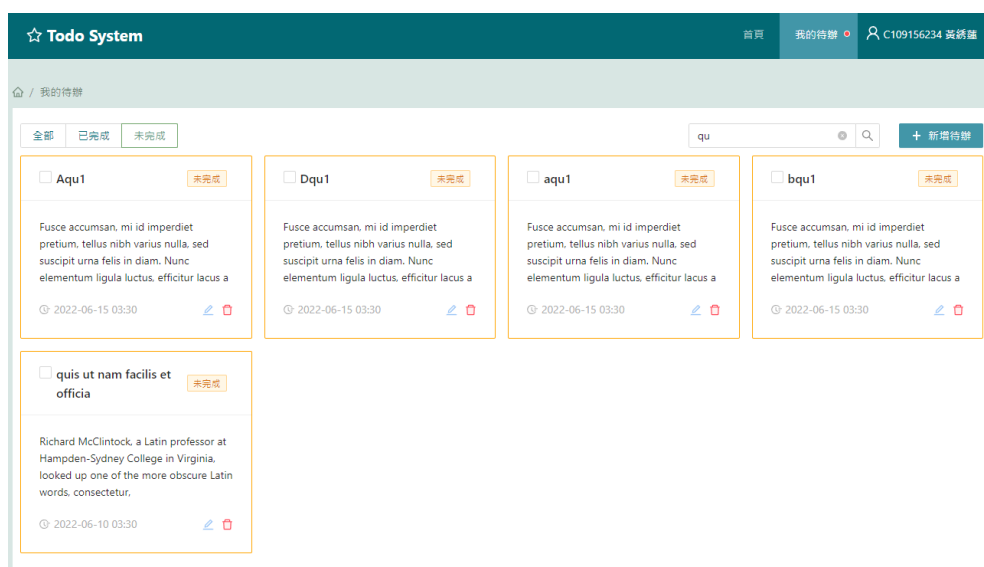


圖 3 Todolist Systemn. 搜尋功能畫面

```

const quickSortData=(arr,first,last)=>{
  if(arr.length===0){
    return;
  }
  let i=first
  let j=last;
  let pivot=arr[Math.floor((first+last)/2)].title.substr(0,1);
  while (i<=j){
    while (arr[i].title.substr(0,1) < pivot){i++;}
    while (arr[j].title.substr(0,1) > pivot){j--;}

    if (i<=j){
      let temp=arr[i];
      arr[i]=arr[j];
      arr[j]=temp;
      i++;
      j--;
    }

    if (first<j) {quickSortData(arr,first,j);}
    if (i<last) {quickSortData(arr,i,last);}
  }
}

```

```

const getTodos=()=>{
  const reg = new RegExp(searchValue,"g");
  if (searchValue=="") {
    switch (queryType) {
      case "completed":
        return todos.filter(todo=>todo.completed);
      case "uncompleted":
        return todos.filter(todo=>!todo.completed);
      case "all":
        return todos;
      default:
        return todos;
    }
  }
  else{
    switch (queryType) {
      case "completed":
        const completedSearch = todos.filter(todo=>todo.completed);
        let filterCompletedSearch = completedSearch.filter(todo=>todo.title.match(reg));
        quickSortData(filterCompletedSearch,0,filterCompletedSearch.length-1)
        return filterCompletedSearch;
      case "uncompleted":
        const uncompletedSearch = todos.filter(todo=>!todo.completed);
        let filterUncompleteSearch = uncompletedSearch.filter(todo=>todo.title.match(reg));
        quickSortData(filterUncompleteSearch,0,filterUncompleteSearch.length-1);
        return filterUncompleteSearch;
      case "all":
        let filterSearch = todos.filter(todo=>todo.title.match(reg));
        quickSortData(filterSearch,0,filterSearch.length-1)
        return filterSearch;
      default:
        return filterSearch;
    }
  }
};

```

圖 4 快速排序法程式碼