

第3次隨堂-隨堂-QZ3

學號：112111215

姓名：莊博勳

作業撰寫時間：180 (mins · 包含程式撰寫時間)

最後撰寫文件日期：2024/11/24

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

- ☒ 說明內容
- ☒ 個人認為完成作業須具備觀念

說明程式與內容

開始寫說明，該說明需說明想法，並於之後再對上述想法的每一部分將程式進一步進行展現，若需引用程式區則使用下面方法，若為.cs檔內程式除了於敘述中需註明檔案名稱外，還需使用語法```語言種類 程式碼```，其中語言種類若是要用python則使用py，java則使用java，C/C++則使用cpp，下段程式碼為語言種類選擇csharp使用後結果：

```
public void mt_getResult(){  
    ...  
}
```

若要於內文中標示部分網頁檔，則使用以下標籤```html 程式碼```，下段程式碼則為使用後結果：

```
<%@ Page Language="C#" AutoEventWireup="true" ...>  
  
<!DOCTYPE html>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
<meta http-equiv="Content-Type" ...>  
    <title></title>  
</head>  
<body>  
    <form id="form1" runat="server">  
        <div>  
            </div>  
    </form>  
</body>  
</html>
```

更多markdown方法可參閱<https://ithelp.ithome.com.tw/articles/10203758>

請在撰寫"說明程式與內容"該塊內容，請把原該塊內上述敘述刪除，該塊上述內容只是用來指引該怎麼撰寫內容。

1. 請參閱投影片Topic5的第31至35頁，請用物件導向方式進行新增與刪除。(請參照題目pdf)

Ans:

```
class Node: #節點
    def __init__(self, item=None):
        self.data = item #儲存資料
        self.link = None #下一節點的鏈結

class Stack: #堆疊
    def __init__(self):
        self.top = None #堆疊頂端預設為空

    def add(self, item: int): #將資料加入堆疊頂端
        new_node = Node(item) #創新節點
        new_node.link = self.top #新節點的link指向堆疊頂端
        self.top = new_node #更新top為新創的節點

    def remove(self) -> int: #從堆疊移除頂端節點並取得其資料
        if self.top is None: #如果堆疊是空的
            raise Exception("STACK_EMPTY") #拋出異常
        x = self.top #取出頂端節點
        item = x.data #保存其資料
        self.top = self.top.link #更新top為下一節點
        del x #刪除頂端節點
        return item #取得被移除節點的資料

class Queue: #佇列
    def __init__(self):
        self.front = None #佇列前端預設為空
        self.rear = None #佇列後端預設為空

    def add(self, item): #將資料加入佇列後端
        new_node = Node(item) #創新節點
        if self.rear is None: #如果佇列是空的,前後端都指向新節點
            self.front = new_node
            self.rear = new_node
        else: #否則只將rear的link指向新節點,並更新rear為新節點
            self.rear.link = new_node
            self.rear = new_node

    def remove(self): #從佇列移除前端節點並取得其資料
        if self.front is None: #如果佇列是空的
            raise Exception("Queue_Empty") #拋出異常
        x = self.front #取出前端節點
        item = x.data #保存其資料
        self.front = self.front.link #更新front為下一節點
        if self.front is None: #如果佇列變成空的
            self.rear = None #將rear設為None
        del x #刪除前端節點
        return item #取得被移除節點的資料s
```

個人認為完成作業須具備觀念

開始寫說明，需要說明本次練習需學會那些觀念 (需寫成文章，需最少50字，並且文內不得有你、我、他三種文字)且必須提供完整與練習相關過程的notion筆記連結