

第3次隨堂-隨堂-QZ3

學號：112111211

姓名：吳雨柔

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:
    def __init__(self, data):
        self.data = data
        self.link = None

class Stack: #堆疊
    def __init__(self):
        self.top = None

    def add_node(self, item):
        new_node = Node(item) #產生一個新節點
        new_node.link = self.top
        self.top = new_node #新節點成為top節點

    def __str__(self):
        elements = []
        current = self.top
        while current:
            elements.append(str(current.data))
            current = current.link
        return " -> ".join(elements)

    def pop(self):
        if self.is_empty(): #判斷鏈結堆疊「是否為空」
            raise Exception("STACK_EMPTY")
        node_to_remove = self.top #指標指向頂端的節點
        item = node_to_remove.data #取出頂端資料
        self.top = self.top.link #頂端指標top改指向第二個節點
        del node_to_remove
        return item

    def is_empty(self):
        return self.top is None

#堆疊範例
stack = Stack()
print("執行堆疊新增範例:")
stack.add_node(1)
stack.add_node(2)
stack.add_node(3)
print(stack) #輸出: 3 -> 2 -> 1

print("執行堆疊刪除範例:")
print(stack.pop()) #輸出: 3
print(stack) #輸出: 2 -> 1

class Queue: #佇列
```

```
def __init__(self):
    self.front = None
    self.rear = None

def is_empty(self):
    return self.front is None

def add_queue(self, item):
    new_node = Node(item)
    if self.is_empty():
        self.front = self.rear = new_node
    else:
        self.rear.link = new_node
        self.rear = new_node

def del_queue(self):
    if self.is_empty():
        raise Exception("Queue_Empty")
    else:
        node_to_remove = self.front #先利用一個指標指向前端節點
        item = node_to_remove.data #取出前端的資料
        self.front = self.front.link #前端的指標front至下一個節點
        if self.front is None:
            self.rear = None
        del node_to_remove
        return item

def __str__(self):
    elements = []
    current = self.front
    while current:
        elements.append(str(current.data))
        current = current.link
    return " -> ".join(elements)

#佇列範例
queue = Queue()
print("執行佇列新增操作:")
queue.add_queue(10)
queue.add_queue(20)
queue.add_queue(30)
print(queue) #輸出: 10 -> 20 -> 30

print("執行佇列刪除操作:")
print(queue.del_queue()) #輸出: 10
print(queue) #輸出: 20 -> 30
```

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