

第3次隨堂-隨堂-QZ3

學號：112111205

姓名：鄭宇辰

作業撰寫時間：200 (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

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: # 節點類別
    def __init__(self, data=None):
        self.data = data # 儲存節點的資料
        self.next = None # 指向下一個節點的鏈結

class Stack: # 堆疊類別
    def __init__(self):
        self.top = None # 堆疊頂端指標

    def push(self, data):
        new_node = Node(data) # 建立新節點
        new_node.next = self.top # 新節點的鏈結指向目前的頂端節點
        self.top = new_node # 更新堆疊頂端
        print(f"已將 {data} 推入堆疊。")

    def pop(self):
        if self.is_empty():
            raise Exception("堆疊為空。") # 堆疊為空時拋出異常
        data = self.top.data # 取得頂端節點的資料
        self.top = self.top.next # 移除頂端節點，更新頂端為下一節點
        print(f"已從堆疊彈出 {data}。")
        return data

    def peek(self):
        if self.is_empty():
            raise Exception("堆疊為空。") # 堆疊為空時拋出異常
        return self.top.data # 返回頂端節點的資料

    def is_empty(self):
        return self.top is None # 當頂端為 None 時，堆疊為空
```

```
class Queue: # 佇列類別
    def __init__(self):
        self.front = None # 佇列前端指標
        self.rear = None # 佇列尾端指標

    def enqueue(self, data):
        new_node = Node(data) # 建立新節點
        if self.rear is None: # 如果佇列為空
            self.front = new_node # 新節點同時成為前端與尾端
            self.rear = new_node
        else:
            self.rear.next = new_node # 尾端的鏈結指向新節點
            self.rear = new_node # 更新尾端為新節點
        print(f"已將 {data} 加入佇列。")

    def dequeue(self):
        if self.is_empty():
            raise Exception("佇列為空。") # 佇列為空時拋出異常
        data = self.front.data # 取得前端節點的資料
        self.front = self.front.next # 移除前端節點，更新前端為下一節點
        if self.front is None: # 如果前端變為空
            self.rear = None # 同時更新尾端為 None
        print(f"已從佇列取出 {data}。")
        return data

    def peek(self):
        if self.is_empty():
            raise Exception("佇列為空。") # 佇列為空時拋出異常
        return self.front.data # 返回前端節點的資料

    def is_empty(self):
        return self.front is None # 當前端為 None 時，佇列為空

# 測試堆疊
print("堆疊測試：")
stack = Stack()
stack.push(10)
stack.push(20)
stack.push(30)
print("堆疊頂端元素:", stack.peek())
stack.pop()
print("堆疊是否為空?", stack.is_empty())
print("")

# 測試佇列
print("佇列測試：")
queue = Queue()
queue.enqueue(1)
queue.enqueue(2)
queue.enqueue(3)
print("佇列前端元素:", queue.peek())
queue.dequeue()
print("佇列是否為空?", queue.is_empty())
```

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

1. 物件導向設計: 學會如何使用類別和物件來管理程式碼，讓堆疊和佇列這些資料結構變得更清晰。
2. 堆疊與佇列的概念: 了解堆疊和佇列的基本運作方式，並學會如何進行操作，如新增、刪除和查看元素。
3. 鏈結串列的應用: 學會使用鏈結串列來實現堆疊和佇列，並理解節點如何連接在一起來存儲資料。