

第2次隨堂-隨堂-QZ2

學號：112111205

姓名：鄭宇辰

作業撰寫時間：60 (mins · 包含程式撰寫時間)

最後撰寫文件日期：2024/12/28

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. 請執行下面Git操作 (請參照題目pdf)

Ans:

```
# Import Stack class from previous code
class Stack:
    def __init__(self, max_size):
        self.items = []          # 用來儲存堆疊中的元素
        self.max_size = max_size # 最大容量設定

    def push(self, item):
        if not self.IsFull():
            self.items.append(item)
        else:
            raise Exception("Stack is full!")

    def pop(self):
        if not self.IsEmpty():
            return self.items.pop()
        else:
            raise Exception("Stack is empty!")

    def TopItem(self):
        if not self.IsEmpty():
            return self.items[-1]
        else:
            return None

    def IsEmpty(self):
        return len(self.items) == 0

    def IsFull(self):
        return len(self.items) == self.max_size

# 測試程式碼
if __name__ == "__main__":
    max_size = 5
    stack = Stack(max_size)
```

```
# 檢查是否空的
print("堆疊空了嗎?", stack.IsEmpty()) # True

# 推入元素
stack.push(10)
stack.push(20)
stack.push(30)

# 檢查堆疊狀態
print("頂部項目:", stack.TopItem()) # 30
print("堆疊空了嗎?", stack.IsEmpty()) # False
print("堆疊滿了嗎?", stack.IsFull()) # False

# 再推入兩個元素，堆疊將滿
stack.push(40)
stack.push(50)

# 檢查堆疊滿的情況
print("堆疊滿了嗎?", stack.IsFull()) # True

# pop 出來一個元素
print("彈出項目:", stack.pop()) # 50
print("彈出後的頂部項目:", stack.TopItem()) # 40

# 再次檢查堆疊狀態
print("堆疊空了嗎?", stack.IsEmpty()) # False
print("堆疊滿了嗎?", stack.IsFull()) # False

# 再清空堆疊
while not stack.IsEmpty():
    stack.pop()

print("清除後堆疊是空的?", stack.IsEmpty()) # True
```

個人認為完成作業須具備觀念

- 1.類別與方法：學會如何設計和實作類別 (Stack) 和方法 (push, pop, TopItem, IsEmpty, IsFull)，這些方法可以模擬堆疊的行為。
- 2.堆疊資料結構：瞭解堆疊是一種後進先出的結構，學會如何操作堆疊元素。
- 3.基本程式設計邏輯：掌握如何利用邏輯控制來檢查堆疊狀態，並在操作中有效地處理元素。