

【2018 JAVA 物件導向程式設計 Homework 10】

● 注意事項

1. 請使用 JAVA 語言，配合 IntelliJ IDEA 寫本次作業並進行測試，並安裝、使用 JAVA SE Development Kit (JDK) 8 函式庫。
2. 請依據作業規定設定 IntelliJ IDEA 專案名稱與 package name，若未依照規定將根據狀況扣分。
3. 嚴禁抄襲其他同學作業，參與者(抄襲與被抄襲)皆以學期成績不及格處理。
4. 請對你的程式碼有深入瞭解，demo 時助教會問。
5. 對題目有問題可以寄信問助教群(java_ta@net.nsysu.edu.tw)或是到實驗室(EC5018)詢問，但不幫忙 debug。
6. 逾期以零分計算，不接受補交，有任何因素導致無法如期繳交，請事先告知。
7. Demo 時間會另外通知。

● 作業規定與上傳

1. IntelliJ IDEA 專案名稱：**StudentID_HW10**

2. 作業請繳交專案之 **tar** 或 **zip archive** 並上傳至網路大學。

請於 **2018 年 5 月 27 日 (週日) 23:59** 前上傳完畢，逾期以零分計算，不接受補交，有任何因素導致無法如期繳交，有問題請事先告知，再次強調，Demo 時間會另外通知。

Account
- name:String - balance: int
+ Account(String name, int balance) + getBalance() : int + setBalance(int newbalance) : void

QueueMachine
- queue : LinkedList<Transaction>
+ isEmpty() : boolean + get() : Transaction + add(Transaction newTran) : void

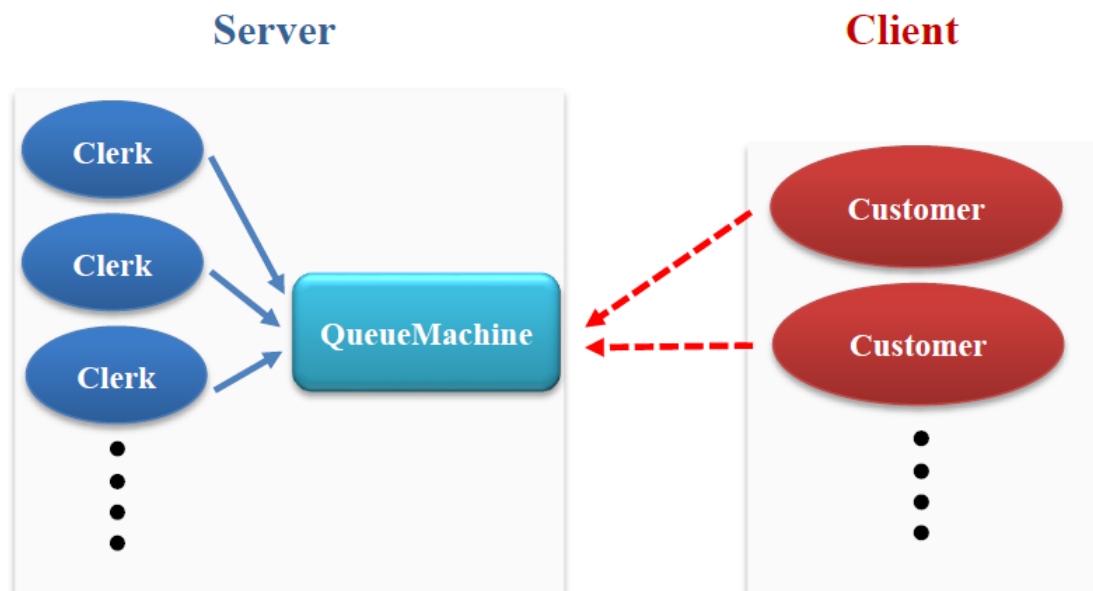
Transaction
- num : int - amount : int - tranType : String - account : Account
+ Transaction(int num, Account account, int amount, String tranType) + getTranType() : String + getAccount() : Account + getAmount() : int + getNum() : int

除了上面的 **UML** 所示的屬性及方法外，皆可自行增加所需要的屬性與方法以及 **class**

● 作業說明

模擬銀行營業據點的工作環境

在銀行內，往往有一定數量的銀行職員為隨機到達的客戶提供服務，每個客戶到達後向排隊機獲取一個服務號碼，銀行職員透過號碼依序服務客戶



● Server功能說明

1. 依照使用者的輸入產生對應數量的 Thread，每個 Thread 為一個員工 (Clerk)，Clerk 向 QueueMachine 索取交易資料來處理
2. 額外以一條 Thread 執行 QueueMachine，QueueMachine 處理 Client 傳來的交易資料並轉換成 Transaction 類別的形式，等待被處理

3. 在 Clerk 的執行緒中，setbalance 的 method 前 sleep 300 ms

4. Port number

Master student: M0X3040ABC -> 6XABC

College student: B0X3040ABC -> 5XABC

Ex. M053040001: port number is 65001

Account

用來儲存 customer 的帳戶餘額

Transaction

用來儲存每一筆交易資料

QueueMachine

將 Client 傳來的資料轉換成 Transaction 類別放進 queue 中等待被處理

● Client 功能說明

1. 依照使用者的輸入產生對應數量的 Thread，每個 Thread 為一個客戶
2. 每一個客戶的帳戶金額預設為 5000，並且隨機產生 5 筆交易資料
交易資料
 tranType：隨機為存款或者提款
 amount：交易金額隨機介於 100 ~ 300 之間
每產生完一筆資料，休眠 100 ~ 300 ms
3. 由於 customer 之間沒有關聯因此可以使用 Multiprocess 或者 Multithread 來實作

● Sample Output

這裡顯示當有兩個 Clerk 一個 Customer 的結果

```
Transaction[run: 1, account = Account[name: Java, balance = 5000], amount = 269, tranType = withdraw]
Transaction[run: 2, account = Account[name: Java, balance = 5000], amount = 125, tranType = deposit]
Transaction[run: 3, account = Account[name: Java, balance = 5000], amount = 272, tranType = deposit]
Transaction[run: 4, account = Account[name: Java, balance = 5000], amount = 234, tranType = withdraw]
Clerk A 取得交易 Transaction[run: 1, account = Account[name: Java, balance = 5000], amount = 269, tranType = withdraw]
Clerk B 取得交易 Transaction[run: 2, account = Account[name: Java, balance = 5000], amount = 125, tranType = deposit]
Transaction[run: 5, account = Account[name: Java, balance = 5000], amount = 165, tranType = withdraw]
Clerk B 完成 2 號交易處理，Java 帳戶餘額: 5125
Clerk B 取得交易 Transaction[run: 3, account = Account[name: Java, balance = 5125], amount = 272, tranType = deposit]
Clerk A 完成 1 號交易處理，Java 帳戶餘額: 4856
Clerk A 取得交易 Transaction[run: 4, account = Account[name: Java, balance = 4856], amount = 234, tranType = withdraw]
Clerk B 完成 3 號交易處理，Java 帳戶餘額: 5128
Clerk B 取得交易 Transaction[run: 5, account = Account[name: Java, balance = 5128], amount = 165, tranType = withdraw]
Clerk A 完成 4 號交易處理，Java 帳戶餘額: 4894
Clerk B 完成 5 號交易處理，Java 帳戶餘額: 4729
```