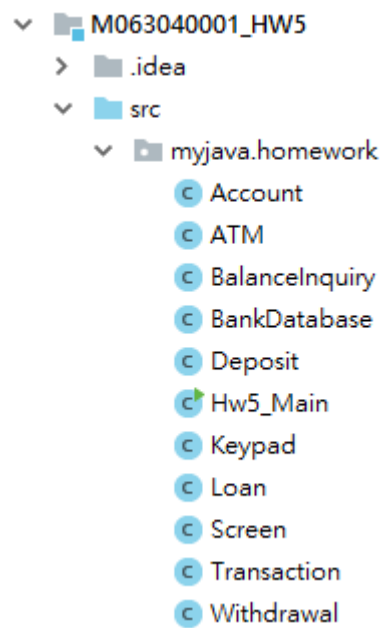# 2018 JAVA Homework 5

● Notice

1. Please use JAVA SE Development Kit(JDK) 8 and IntelliJ IDEA in this homework.
2. Do not copy homework of others. If it happened, you will get ZERO whether you are either the owner of the homework or the copycat.
3. You have to deeply understand what your program do because TA will ask you something about your program during the demo.
4. If you have any question, please send email to java_ta@net.nsysu.edu.tw or come to EC5018, but TA does not help to debug.
5. If you do not submit your assignment on time, you will not hand in the delayed homework and get ZERO as well. If you have trouble, please advise in advance by email. Moreover, time and place for demo will be announced later.

● Upload

1. Please compress your homework into zip or tar archive.
2. Naming rules: "StudentID _HW5.zip".
        For example: M063040001_HW5.zip

3. Upload your homework to National Sun Yat-sen Cyber University.

4. Deadline: 2018/04/15 (Sun.) 23:59.

5. Example of package explorer

```
∨  M063040001_HW5
   >  .idea
   ∨  src
      ∨  myjava.homework
            C Account
            C ATM
            C BalanceInquiry
            C BankDatabase
            C Deposit
            C Hw5_Main
            C Keypad
            C Loan
            C Screen
            C Transaction
            C Withdrawal
```

● Hint

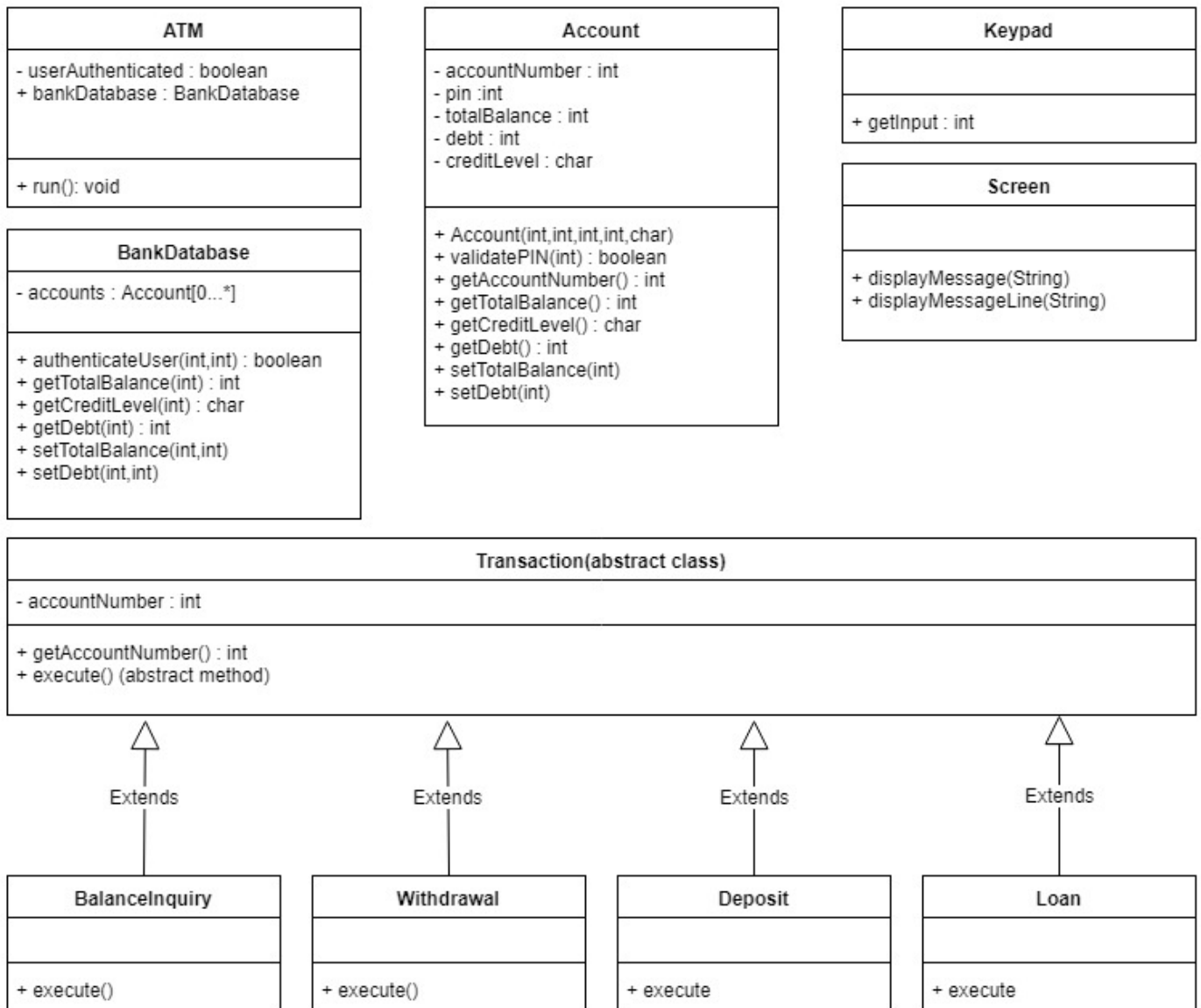Any element (+, -, #, _, etc.) in UML are meaningful.
Reference: Class diagram (Wikipedia).

所有例外 情況都必須 處理， 程式 不得在使用者任意 輸入 的狀況下意外
結束， 否則將斟酌扣分。

● Introduction

根據作業提供的 UML 實作一個 ATM 系統，並練習使用 inheritance, polymorphism, abstract class 及 abstract method.

● UML

ATM
- userAuthenticated : boolean + bankDatabase : BankDatabase
+ run(): void

Account
- accountNumber : int - pin :int - totalBalance : int - debt : int - creditLevel : char
+ Account(int,int,int,int,char) + validatePIN(int) : boolean + getAccountNumber() : int + getTotalBalance() : int + getCreditLevel() : char + getDebt() : int + setTotalBalance(int) + setDebt(int)

Keypad
+ getInput : int

Screen
+ displayMessage(String) + displayMessageLine(String)

BankDatabase
- accounts : Account[0...*]
+ authenticateUser(int,int) : boolean + getTotalBalance(int) : int + getCreditLevel(int) : char + getDebt(int) : int + setTotalBalance(int,int) + setDebt(int,int)

Transaction(abstract class)
- accountNumber : int
+ getAccountNumber() : int + execute() (abstract method)

Extends   Extends   Extends   Extends

BalanceInquiry
+ execute()

Withdrawal
+ execute()

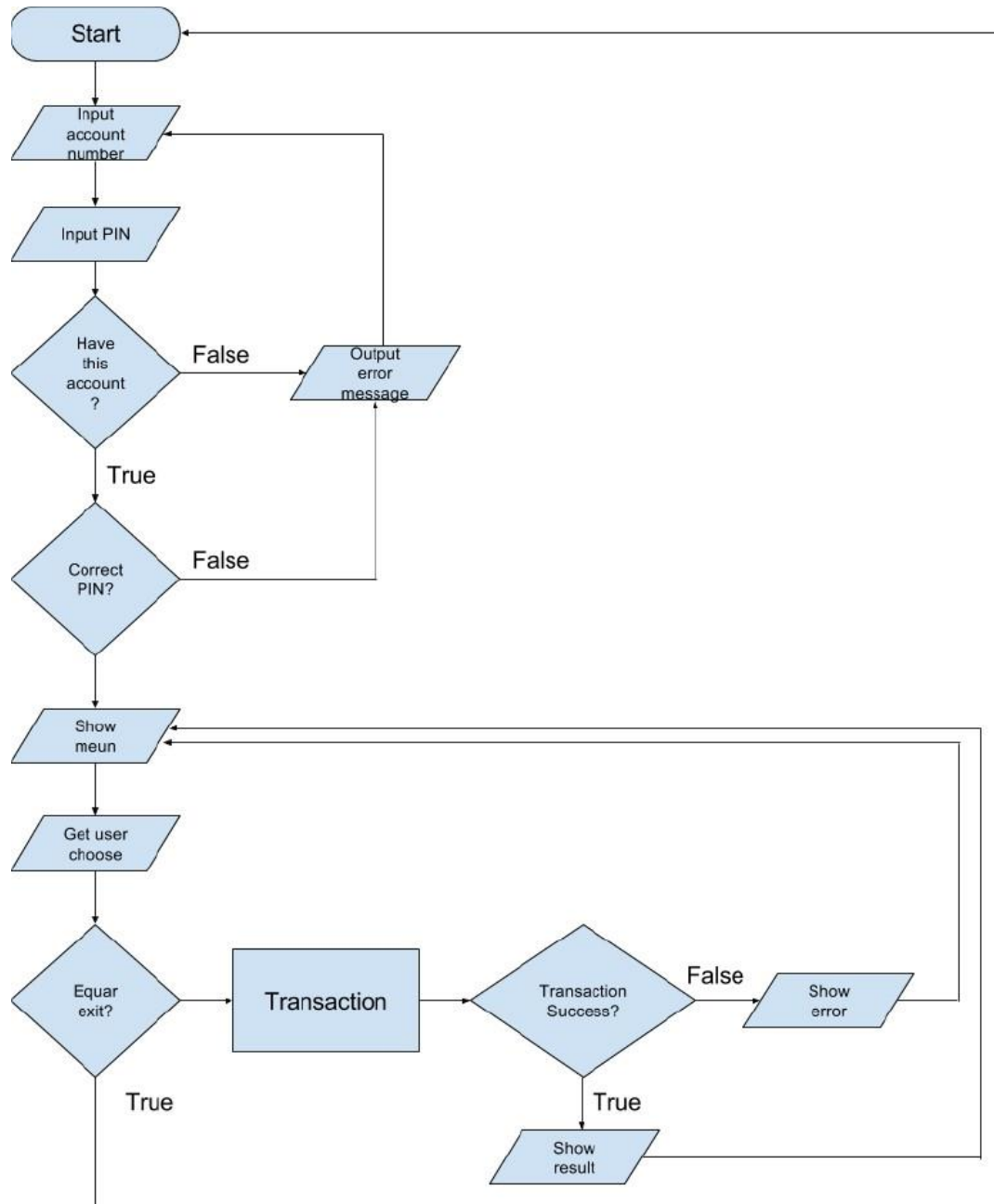| Deposit |
|---|
| |
| + execute |

Loan
+ execute

● Program Flow

　　ATM 中的 run 方法會使用無窮迴圈來重複歡迎者及認證，若成功認證，則使用者可以繼續進行交易完所需要的動作後選擇離開，ATM 會重設帳戶參數，再新開始上述流程，請使用無窮迴圈模擬 ATM，使用者可以離開系統但不會關閉 ATM。

程式流程圖如下：

程式中的交易主要分為兩個部份：

1. 帳戶儲蓄功能: 包含取得帳戶餘額、存錢及領錢等功能，模擬一般的帳戶儲蓄步驟。

2. 借貸功能: 主要為借貸功能的模擬，並根據帳戶的信用等級給予不同的借貸額度，信用等級分為：

    I.　A：可借貸 11000 元

    II.　B：可借貸 9000 元

    III.　C：可借貸 7000 元

    IV.　D：可借貸 5000 元

    超過借貸金額則不予借貸，可重複借貸直到借貸上限為止，不與帳戶儲蓄功能有互動。

● Hint

提供 11 個參考檔案，可直接複製使用每類別中，除了 UML 所示的屬性之外，可自行添加所需的變數或函式。

帳戶需自建，建議使用參考檔案 BankDatabase.java 中預設帳戶，以方便日後 DEMO 進行。

● Sample Output

登入及儲蓄功能:

```
Wellcone !
Please enter your account number : 111
Please enter your pin : 333
AccountNumber or Pin Error

Wellcone !
Please enter your account number : 111
Please enter your pin : 222

Main_menu :
1. View my balance
2. Withdraw
3. Deposit
4. Loan
5. Exit
Enter a choise : 1
Balance Information
Total Balence : 5000
```

```
Main_menu :
1. View my balance
2. Withdraw
3. Deposit
4. Loan
5. Exit
Enter a choise : 2
How much do you want to withdraw : 1000
Transaction Success

Main_menu :
1. View my balance
2. Withdraw
3. Deposit
4. Loan
5. Exit
Enter a choise : 1
Balance Information
Total Balence : 4000

Main_menu :
1. View my balance
2. Withdraw
3. Deposit
4. Loan
5. Exit
Enter a choise : 3
How much do you want to deposit : 2000
Transaction Success

Main_menu :
1. View my balance
2. Withdraw
3. Deposit
4. Loan
5. Exit
Enter a choise : 1
Balance Information
Total Balence : 6000
```

借貸功能:

```
Main_menu :
1. View my balance
2. Withdraw
3. Deposit
4. Loan
5. Exit
Enter a choise : 4
Your debt : 0
Your loan limit is 11000, How much do you want to loan : 10000
Transaction Success

Main_menu :
1. View my balance
2. Withdraw
3. Deposit
4. Loan
5. Exit
Enter a choise : 4
Your debt : 10000
Your loan limit is 1000, How much do you want to loan : 2000
Transaction Error, You have not much Loan Limit

Main_menu :
1. View my balance
2. Withdraw
3. Deposit
4. Loan
5. Exit
Enter a choise : 4
Your debt : 10000
Your loan limit is 1000, How much do you want to loan : 1000
Transaction Success
```

```
Main_menu :
1. View my balance
2. Withdraw
3. Deposit
4. Loan
5. Exit
Enter a choise : 4
Your debt : 11000
Sorry, You can not loan any money now.

Main_menu :
1. View my balance
2. Withdraw
3. Deposit
4. Loan
5. Exit
Enter a choise : 5

Wellcone !
Please enter your account number : |
```