Project 1

CST 338

In this project, you will develop four classes called `Bank`, `Account`, `Customer`, and `Transaction` to store account information, customer information, and transaction information for a bank. Based on the sample input data and sample run, you should identify operations of the project.

For the project, you can use either `Array` or `ArrayList`. If you use arrays, you can assume that your `Bank` class will have a maximum 30 accounts, 30 customers, and 100 transactions. Note that an `Account` object can have only one `Customer`, and each `Customer` can have a maximum of **two** `Accounts` (= one checking and one savings account). A `Transaction` object should include the account number, transaction type (= deposit, withdrawal, closing an account), and transaction date/time information.

Sample Input File

The following file represents a sample bank data file, `test1.txt` which is used in the sample runs below. Note that the first line (= 4) indicates the number of customers in the bank. The customer's information includes the name, address, zip code, and social security number (SSN or *sosh*). Note that each piece of customer data is delimited by the comma symbol (,).

After the customer data, there is a number, representing the number of accounts (= 5 in the example). Each account information has SSN, account number, account type (1: checking, 2: savings), and current balance. Again, the comma is used as a delimiter.

```
4
Tom Smith,123 University Center,93955,777-77-7777
Alice Smith,123 University Center,93955,888-88-8888
Joe Otter,2440 Ocean Avenue,93900,999-99-9999
Monica Smith,123 University Center,93955,123-45-7777
5
777-77-7777,1000,1,10.0
888-88-8888,2000,1,50.25
777-77-7777,3000,2,100.0
999-99-9999,5000,1,100.25
888-88-8888,6000,2,500.25
```

Sample Demo Program – 1

The following presents a sample demo program called `BankDemo1.java`.

```java
public class BankDemo1 {
  public static void main(String[] args) {
    Bank csumbBank = new Bank("CSUMB");
    System.out.println("========== READ DATA ==========");
    csumbBank.readData("resources/test1.txt");
    System.out.println("========== DONE ==========");
    System.out.println("\n========== BANK INFORMATION ==========");
    csumbBank.bankInfo();
    System.out.println("\n========== ACCOUNT INFORMATION ==========");
    csumbBank.accountInfo(1000);
    csumbBank.deposit(1000, 150.25);
    System.out.println("\n========== ACCOUNT INFORMATION ==========");
    csumbBank.accountInfo(1000);
    csumbBank.withdraw(1000, 100);
    System.out.println("\n========== ACCOUNT INFORMATION ==========");
    csumbBank.accountInfo(1000);
    System.out.println("\n========== ACCOUNT CLOSE ==========");
    if (csumbBank.closeAccount(1000)) {
      System.out.println("Account closed.");
    }
    System.out.println("\n========== TRANSACTION INFO ==========");
    csumbBank.transaction(1000);
    System.out.println("\n========== TRANSACTION INFO ==========");
    csumbBank.transaction(2000);
  }
}
```

The following presents a sample execution result. For the sample run, we assume that the deposit of the account number `1000` happened on `September 22, 2020`. The exact time is `15:23:25` (=HH:MM:SS format). When you run the demo program, your program should get the date and time of the `deposit()` method execution time. Similarly, we assume that the `withdraw()` and `closeAccount()` methods were invoked at `2020-09-22` (=YYY-MM-DD), at `15:23:26` and `15:23:28`, respectively.

```
========== READ DATA ==========
========== DONE ==========

========== BANK INFORMATION ==========
Bank Name: CSUMB
Number of Customers: 4
    Tom Smith: 777-77-7777
    Alice Smith: 888-88-8888
    Joe Otter: 999-99-9999
    Monica Smith: 123-45-7777
Number of Accounts: 5
    1000: $10.00
    2000: $50.25
    3000: $100.00
    5000: $100.25
    6000: $500.25
Total Balance: $760.75

========== ACCOUNT INFORMATION ==========
  - Number: 1000
  - Checking
  - Balance: $10.00
  - Customer: Tom Smith

========== ACCOUNT INFORMATION ==========
  - Number: 1000
  - Checking
  - Balance: $160.25
  - Customer: Tom Smith

========== ACCOUNT INFORMATION ==========
  - Number: 1000
  - Checking
  - Balance: $60.25
  - Customer: Tom Smith

========== ACCOUNT CLOSE ==========
Account closed.

========== TRANSACTION INFO ==========
  - Account Number: 1000, Deposit ($150.25), 2020-09-22, 15:23:25
  - Account Number: 1000, Withdraw ($100.00), 2020-09-22, 15:23:26
  - Account Number: 1000, Account closed, 2020-09-22, 15:23:28

========== TRANSACTION INFO ==========
  - No transaction for account 2000
```

Sample demo program – 2

This is another sample program called `BankDemo2.java`.

```java
public class BankDemo2 {
  public static void main(String[] args) {
    Bank csumbBank = new Bank("CSUMB");
    csumbBank.readData("resources/test1.txt");
    System.out.println("========== NEW CUSTOMERS ==========");
    csumbBank.newCustomer("Bob Smith", "123 University Center",
                93955, "123-45-6789");
    csumbBank.newCustomer("Unknown Smith", "123 University Center",
                93955, "777-77-7777");
    System.out.println("\n========== NEW ACCOUNTS ==========");
    csumbBank.newAccount("777-77-7777", 4000, 1, 100.50);
    csumbBank.newAccount("123-45-7777", 2000, 1, 100.50);
    csumbBank.newAccount("123-45-7777", 4000, 1, 100.50);
    System.out.println("\n========== ACCOUNT INFO ==========");
    csumbBank.accountInfo(7000);
    System.out.println("\n==== CUSTOMER WITH LAST FOUR DIGIT SSN ====");
    csumbBank.customerInfoWithSSN(7979);
    System.out.println("\n==== CUSTOMER WITH SSN 7777 ====");
    csumbBank.customerInfoWithSSN(7777);
    System.out.println("\n==== CUSTOMER WITH SSN 6789 ====");
    csumbBank.customerInfoWithSSN(6789);
    // We try to close the account 2000.
    csumbBank.closeAccount(2000);
    System.out.println("\n==== REMOVE CUSTOMER: 555-55-5555 ====");
    csumbBank.removeCustomer("555-55-5555");
    System.out.println("\n==== REMOVE CUSTOMER: 777-77-7777 ====");
    csumbBank.removeCustomer("777-77-7777");
  }
}
```

A sample run of your program should look like below.

```
========== NEW CUSTOMERS ==========
Bob Smith is added.
Unknown Smith is NOT added - Duplicated SSN.

========== NEW ACCOUNTS ==========
Account creation failed - Tom Smith (777-77-7777) already has a checking
account
Account creation failed - Account 2000 already exists
Account creation - Number: 4000, Customer: Monica Smith

========== ACCOUNT INFO ==========
Account (7000) does not exist.

==== CUSTOMER WITH LAST FOUR DIGIT SSN ====
No customer with 7979

==== CUSTOMER WITH SSN 7777 ====
Name: Tom Smith
      123 University Center, 93955
      SSN: 777-77-7777
      Checking (1000), $10.00
      Savings (3000), $100.00
Name: Monica Smith
      123 University Center, 93955
      SSN: 123-45-7777
      Checking (4000), $100.50

==== CUSTOMER WITH SSN 6789 ====
Name: Bob Smith
      123 University Center, 93955
      SSN: 123-45-6789
      No account

==== REMOVE CUSTOMER: 555-55-5555 ====
Customer remove failed. SSN does not exist.

==== REMOVE CUSTOMER: 777-77-7777 ====
Customer removed - SSN: 777-77-7777, Customer: Tom Smith
  Account closed - Number: 1000, Balance: 10.00
  Account closed - Number: 3000, Balance: 100.00
```

Your program will be graded based on:

1. Compilation without error.
2. Correct output result.
3. Good programming structure.
4. Comments. (Title, Abstract, Author, and Date are mandatory.)
5. Meaningful and related variable names.

How to turn in?

Turn in your Java files (`Bank.java`, `Account.java`, `Customer.java`, and `Transaction.java`) on iLearn. Submit your files individually.