Homework 4

CST 338

Task 1 (10 points)

Develop a class called `VendingMachine` that simulates an imaginary vending machine at CSUMB. In the program, a user can buy a bottle of water ($1.50), a coffee ($2.00), a bag of chips ($1.00), and a chocolate bar ($2.50) from the machine.

The user can buy several items if they are available in the machine. For payment, a user can use only cash (including coins). Additionally, an administrator of the machine can reset or refill the machine.

Demo Program (link)

```
public class VendingMachineDemo {
  public static void main(String[] args) {
    VendingMachine machine1 = new VendingMachine(100);
    VendingMachine machine2 = new VendingMachine(200, "Library");
    System.out.println("===== Welcome to the CSUMB Vending Machine =====");
    System.out.println(machine1);
    System.out.println("");
    System.out.println(machine2);
    System.out.println("\n=== Compare Two Machines ===");
    System.out.println(machine1.equals(machine2));
    machine1.setLocation("BIT104");
    machine1.setName(50);
    machine1.setName(100);
    System.out.println("\n=== Machine Reset ===");

    /*
    for machine1() we assume that a machine administrator
    resets the contents of a machine
    */
    machine1.reset(4, 5, 0, 6);
    machine1.addItems(1, 2, 3, 4); // A system admin can add items to the machine
    System.out.println(machine1);
    System.out.println("");
    machine1.displayMenu();

    System.out.println("\n=== Buy an Item ===");
    machine1.buyItem();
    System.out.println("\n=== Buy another Item ===");
    if (machine1.buyItem(1, 4) == false) {
        System.exit(1);
    }
```

```
    System.out.println("\n=== Return Two Items ===");
    machine1.returned(1, 2);
    machine1.returned(2, 3);
    System.out.println("\n=== Buy another Item ===");
    machine1.buyItem(3, 5);
    System.out.println("\n=== Pay for items selected. ===");
    if (machine1.payment()) {
        System.out.println("\n=== Valid payment. ===");
        machine1.displayReceipt();
    } else {
        System.out.println("\n=== Invalid payment. Try one more time. ===");
        if (machine1.payment()) {
            System.out.println("\n=== Print Receipt ===");
            machine1.displayReceipt();
        } else {
            System.exit(1);
        }
    }
    machine1.addItems(5, 5, 5, 5);
    System.out.println("\n=== Machine Status ===");
    machine1.status();
    System.out.println("\n===== Thank you! =====");
  }
}
```

Sample Run of the Demo Program: The following presents a sample result of the demo program. Read the result very carefully to identify the operations of the program.

```
===== Welcome to the CSUMB Vending Machine =====
Serial Number: 100
Location: UNKNOWN
Contents:
  Water: 0
  Coffee: 0
  Sun Chips: 0
  Chocolate Bar: 0

Serial Number: 200
Location: Library
Contents:
  Water: 0
  Coffee: 0
  Sun Chips: 0
  Chocolate Bar: 0

=== Compare Two Machines ===
false
```

```
=== Machine Reset ===
Serial Number: 100
Location: BIT104
Contents:
  Water: 5
  Coffee: 7
  Sun Chips: 3
  Chocolate Bar: 10

===== Vending Machine Menu =====
  1. Water.............$1.50
  2. Regular Coffee...$2.00
  3. Sun Chips........$1.00
  4. Chocolate Bar....$2.50

=== Buy an Item ===
Select an item number: 2
How many do you want to buy? 3
You selected Regular Coffee. Quantity: 3

=== Buy another Item ===
Select an item number: 1
How many do you want to buy? 4
You selected Water. Quantity: 4

=== Return Two Items ===
You selected Water. Quantity: 2
You selected Regular Coffee. Quantity: 3

=== Buy another Item ===
Select an item number: 3
How many do you want to buy? 5
You selected Sun Chips. Quantity: 5
Selection Failed. We don't have enough Sun Chips.

=== Pay for items selected. ===
Enter money amount: $2.00
Insufficient money. $2.00 returned

=== Invalid payment. Try one more time. ===
Enter money amount: $5.00
Sufficient money. $1.70 returned

=== Print Receipt ===
Water: $1.50 X 2 = $3.00
Tax (10.0%): $0.30
Total: $3.30
```

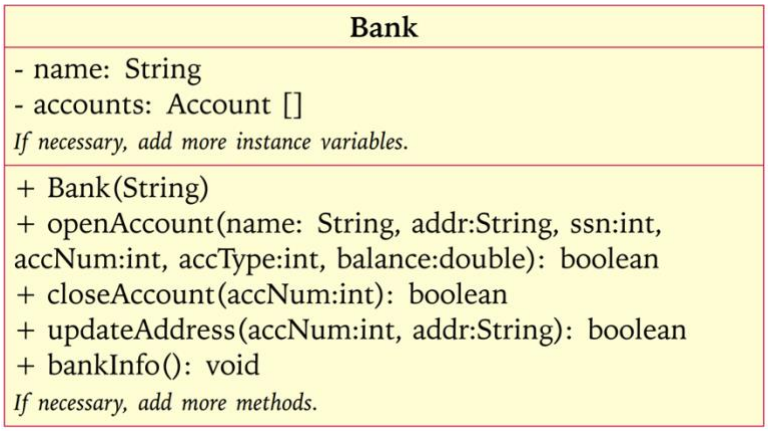
```
=== Machine Status ===
Serial Number: 100
Location: BIT104
Sold Items:
  Water: 2
  Coffee: 0
  Sun Chips: 0
  Chocolate Bar: 0
Current Contents:
  Water: 8
  Coffee: 12
  Sun Chips: 8
  Chocolate Bar: 15
Total Earning: $3.30

===== Thank you! =====
```

Task 2 (10 points)

Develop three classes called `Bank`, `Account`, and `Customer` to store account information and its customer information for a bank.

Bank Class



Your `Bank` class should have a maximum of five accounts. For the `openAccount()` method, your class should add a new account with the account number (`accNum`) and the corresponding information. Additionally, your method should put the customer information with the SSN number in the `Account` object. If the account number is already taken by another object or the bank already has too many accounts (= five),

your method should return false. Furthermore, if a customer with the SSN already has an account in the bank, your method should return false as well.

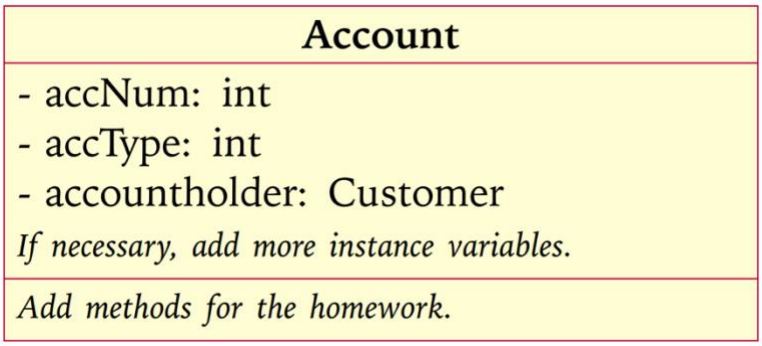
For the `closeAccount()` method, your class should delete the account with the account number (`accNum`) from the bank. If the account number doesn't exist in the bank, your method should return false.

For the `accountInfo()` method, your class should display the account information with the corresponding customer information. If the account number doesn't exist in the bank, your method should return false.

For the `updateBalance()` method, your class should change the balance with the new amount for the account number. If the account number doesn't exist in the bank or the balance is a negative number, your method should return false.

For the `updateAddress()` method, your class should update the address of the customer with the account number. If the account number doesn't exist in the bank, your method should return false.

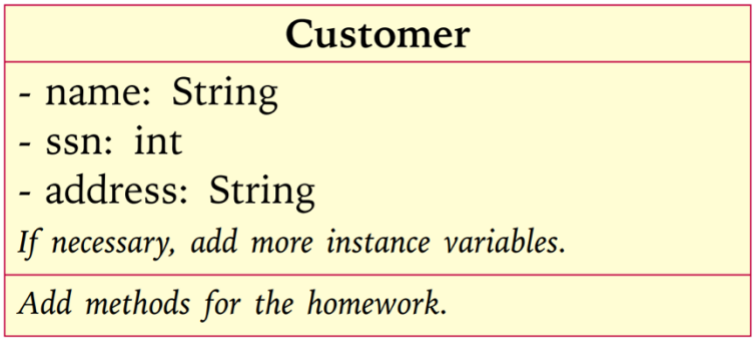
Account Class



Note:

- For `accType`, "1" is checking and "2" is savings.
- `accountholder` holds the `Customer` object for the account.
- One `Account` object can have only one account holder (= `Customer`).

Customer Class



Sample demo program (link)

The following presents a sample demo program called `BankDemo.java`. For the homework, your program should display messages similar to the sample run.

```
public class BankDemo {
  public static void main(String[] args) {
    Bank csumbBank = new Bank("CSUMB");
    System.out.println("\n========== Three New Accounts ==========");
    csumbBank.openAccount("Tom Smith", "123 University Center 93955",
                          77777, 1000, 1, 10.0);
    csumbBank.openAccount("Alice Smith", "123 University Center 93955",
                          88888, 2000, 1, 50.25);
    csumbBank.openAccount("Joe Otter", "2440 Ocean Avenue 93900",
                          99999, 3000, 2, 100.25);
    System.out.println("\n========== Bank Info ==========");
    csumbBank.bankInfo();
    System.out.println("\n========== Close Account ==========");
    System.out.println(csumbBank.closeAccount(1000));
    System.out.println(csumbBank.closeAccount(7000));
    System.out.println("========== Account Info ==========");
    csumbBank.accountInfo(2000);
    System.out.println(csumbBank.accountInfo(7000));
  }
}
```

A sample run of your program should look like:

```
========== Three New Accounts ==========

========== Bank Info ==========
Bank Name: CSUMB
Number of Accounts: 3
  1000: $10.00 - Tom Smith: 77777
  2000: $50.25 - Alice Smith: 88888
```

```
  3000: $100.25 - Joe Otter: 99999
Bank Total Balance: $160.50

========== Close Account ==========
true
false
========== Account Info ==========
Account Info:Account Number: 2000
                    Checking account
                    Balance: $50.25

Customer:    Alice Smith
             123 University Center 93955
             SSN: 8888

false
```

Your program will be graded based on:

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