Lab Exercise (Chapter 9: Part 2)
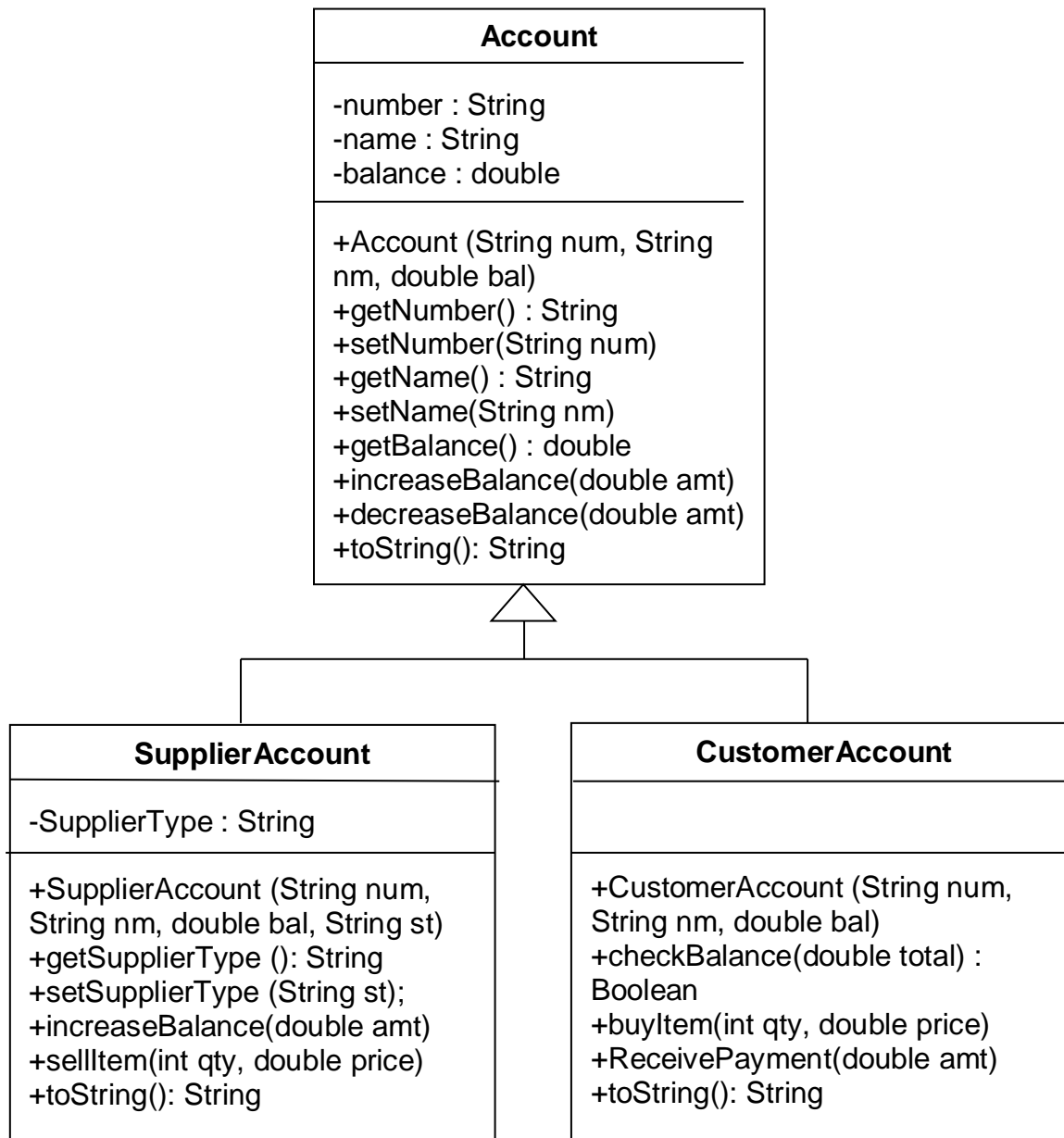
Exercise 1: (inheritance)
Description of the Problem

The following UML diagram shows an inheritance relationship between three classes:

Account

-number : String
-name : String
-balance : double

+Account (String num, String nm, double bal)
+getNumber() : String
+setNumber(String num)
+getName() : String
+setName(String nm)
+getBalance() : double
+increaseBalance(double amt)
+decreaseBalance(double amt)
+toString(): String

SupplierAccount

-SupplierType : String

+SupplierAccount (String num, String nm, double bal, String st)
+getSupplierType (): String
+setSupplierType (String st);
+increaseBalance(double amt)
+sellItem(int qty, double price)
+toString(): String

CustomerAccount

+CustomerAccount (String num, String nm, double bal)
+checkBalance(double total) : Boolean
+buyItem(int qty, double price)
+ReceivePayment(double amt)
+toString(): String

You should note the following points:

Account class:
- **increaseBalance**: add the balance by the amount if the amount is > 0
- **decreaseBalance**: deduct the balance by the amount if the amount is > 0
- **toString**: return the string of "account number xxx, name xxx, current balance is: xxxx" (tips: use the method **String.format**)

SupplierAccount class:
- variable **SupplierType** can have value "VIP" or "Normal"
- **increaseBalance**:
  - o if the supplier is of type "VIP", add the balance by the amount * 2
  - o else add the balance by amount

  (tips: use the method **str1.equals(str2)** to compare string)
- **sellItem**: increase the balance by the price * quantity
- **toString**: return the string of "Supplier: type xxx, account number xxx, name xxx, current balance is: xxxx"

CustomerAccount class:
- **checkBalance**:
  - o check the balance if the current balance is > total, return true, else return false.
- **BuyItem**: call the checkBalance method, if it's true (means balance > 0), reduce the balance by the price * quantity
- **ReceivePayment**: add the balance by the amount
- **toString**: return the string of "Customer: account number xxx, name xxx, current balance is: xxxx"

Build a main application **AccountTest** and test out your classes with below coding:

```
Account acc1 = new Account("A001", "Pensonic", 0);
SupplierAccount Sacc1 = new SupplierAccount("S001", "Jones",0,"VIP");
SupplierAccount Sacc2 = new SupplierAccount("S002", "Albert",0,"Normal");
CustomerAccount Cacc1= new CustomerAccount("C001", "Daniel",1000);

Sacc1.sellItem(100, 5.80);
Sacc2.sellItem(100, 5.80);

Cacc1.buyItem(7, 250.30);
Cacc1.ReceivePayment(200);

System.out.println(acc1.toString());
System.out.println(Sacc1.toString());
System.out.println(Sacc2.toString());
System.out.println(Cacc1.toString());

System.out.println(acc1);
System.out.println(Sacc1);
System.out.println(Sacc2);
System.out.println(Cacc1);
```

Enhance your **AccountTest** program above by creating an array/ArrayList and add all the accounts you've created into the array. Print out all the account details by using a loop through your array and perform the specific operation (as per stated) if it's a SupplierAccount or CustomerAccount.