Lab Assignment 7

Due (2 weeks)

"BankAccount and **SavingsAccount" app**

(100 points)

Objectives

- Be able to declare a new class
- Be able to write a constructor
- Be able to instantiate an object
- Be able to use calls to instance methods to access and change the state of an object
- Be able to derive a class from an existing class
- Be able to define a class hierarchy in which methods are overridden and fields are hidden
- Be able to use derived-class objects
- Implement a copy constructor

PROJECT OVERVIEW

Design an abstract class named **BankAccount** to hold the following data for a bank account:

• Balance
• Number of deposits this month
• Number of withdrawals
• Annual interest rate
• Monthly service charges
• Pin code

The class should have the following methods:
Constructor: The constructor should accept arguments for the balance and annual interest rate.

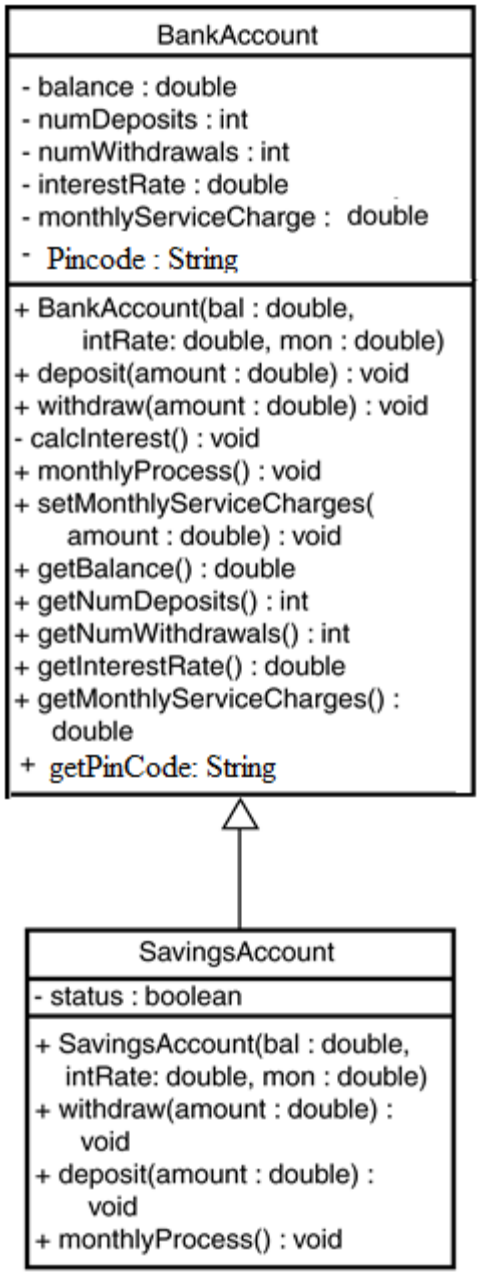
deposit: A method that accepts an argument for the amount of the deposit.
The method should add the argument to the account balance. It should also *increment the variable* holding the number of deposits.

withdraw: A method that accepts an argument for the amount of the withdrawal. The method should subtract the argument from the balance. It should also *increment the variable* holding the number of withdrawals.

calcInterest: A method that updates the balance by calculating the monthly interest earned by the account, and adding this interest to the balance. This is performed by the following formulas:

Monthly Interest Rate = [Annual Interest Rate / 12)
*Monthly Interest = Balance * Monthly Interest Rate*
Balance = Balance + Monthly Interest

monthlyProcess: A method that subtracts the monthly service charges from the balance, calls the calcInterest method, and then sets the variables that hold the number of withdrawals, number of deposits, and monthly service charges to zero.



Next, design a **SavingsAccount** class that extends the **BankAccount** class. The SavinqsAccount class should have a status field to represent an active or inactive account. If the balance of a savings account falls below $25, it becomes inactive. The status field could be a Boolean variable.) No more withdrawals may be made until the balance is raised above $25, at which time the account becomes active again. The savings account class should have the following methods:

withdraw: A method that determines whether *the account is inactive* before a withdrawal is made. (No withdrawal will be allowed if the account is not active.) A withdrawal is then made by calling the superclass version of the method.

deposit: A method that determines whether *the account is inactive* before a deposit is made. If the account is inactive and the deposit brings the balance above $25, the account becomes active again. A deposit is then made by calling the superclass version of the method.

monthlyProcess: Before the superclass method is called, this method checks the number of withdrawals. If the number of withdrawals for the month is more than 4, a service charge of $1 for each withdrawal above 4 is added to the superclass field that holds the monthly service charges. (Don't forget to check the account balance after the service charge is taken. If the balance falls below $25, the account becomes inactive.)

OUTPUT

Please enter the balance amount: 100

Please enter the interest rate: 0.03

Please enter a monthly service charge: 2.50

Please enter your pin code: ****

Please enter a correct pin code: ****

What action do you wish to take?

List the balance, number of deposits and withdrawals: Enter 1:

Deposit money: Enter 2:

Withdraw money: Enter 3:

Exit: Enter 4:

Enter 1, 2, 3, or 4: 1

Balance: $100.00

Number of deposits: 0

Number of withdrawals: 0

What action do you wish to take?

List the balance, number of deposits and withdrawals: Enter 1:

Deposit money: Enter 2:

Withdraw money: Enter 3:

Exit: Enter 4:

Enter 1, 2, 3, or 4: 2

Enter the amount that you want to deposit: 25

What action do you wish to take?

List the balance, number of deposits and withdrawals: Enter 1:

Deposit money: Enter 2:

Withdraw money: Enter 3:

Exit: Enter 4:

Enter 1, 2, 3, or 4: 1

Balance: $125.00

Number of deposits: 1

Number of withdrawals: 0

What action do you wish to take?

List the balance, number of deposits and withdrawals: Enter 1:

Deposit money: Enter 2:

Withdraw money: Enter 3:

Exit: Enter 4:

Enter 1, 2, 3, or 4: 3

Enter the amount that you want to withdraw: 100

What action do you wish to take?

List the balance, number of deposits and withdrawals: Enter 1:

Deposit money: Enter 2:

Withdraw money: Enter 3:

Exit: Enter 4:

Enter 1, 2, 3, or 4: 1

Balance: $25.00

Number of deposits: 1

Number of withdrawals: 1

What action do you wish to take?

List the balance, number of deposits and withdrawals: Enter 1:

Deposit money: Enter 2:

Withdraw money: Enter 3:

Exit: Enter 4:

Enter 1, 2, 3, or 4: 3

Enter the amount that you want to Withdraw: 10

Sorry you can't withdraw 10$ from your balance, your balance is below or equal 25$.

What action do you wish to take?

List the balance, number of deposits and withdrawals: Enter 1:

Deposit money: Enter 2:

Withdraw money: Enter 3:

Exit: Enter 4:

Enter 1, 2, 3, or 4: 4

Implementation Phase

Write Java code to implement your design.

Assignment Submission

- Your project folder should be named la7cs1110_yourlastnameFirstInitial_mmddyy, Replace "yourlastnameFirstInitial" and "mmddyy" appropriately.
- Generate a .zip file that contains all of your files in the above java project folder.
- Submit the .zip file via E-learning.
- Any submitted file to the Instructors' emails will NOT be graded.