Lab 6 Pt2 (15 points total)
The purpose of this lab is to apply practical experience to Chapter 6 concepts. There are several learning objectives to this assignment

- Using loops to solve problems (for, while, do while)
- Creating Classes and using objects in separate Java applications (main()) classes
- Reinforcement of methods

BankAccount (15pts)
Design a BankAccount class that stores a savings account's annual interest rate and balance based on the below UML. The class constructor should accept the amount of the savings account's starting balance. You will also need to create a no-arg constructor that sets monthlyIntRate to 0.035/12 (3.5% divided by 12 to create a monthly rate) and sets all other instance vars to 0. The class should have methods for subtracting the amount of a withdrawal, adding the amount of a deposit, and adding the amount of monthly interest to the balance. The monthly interest rate is the annual interest rate divided by twelve. To add the monthly interest to the balance, multiply the monthly interest rate by the balance, and add the result to the balance, ***only if the balance is >0***.

You will also need to complete the partial BankAccountDemo app class. This provides the user options to input values via console for input and output or dialog box for input and output. Test the class in a program that calculates the balance of a savings account at the end of a period of time. It should ask the user for the annual interest rate, the starting balance, and the number of months that have passed since the account was established. A loop should then iterate once for every month, performing the following:

Ask the user for the amount deposited into the account during the month. Use the class method to add this amount to the account balance.

Ask the user for the amount withdrawn from the account during the month. Use the class method to subtract this amount from the account balance.

Use the class method to calculate the monthly interest.

After the last iteration, the program should display the ending balance, the total amount of deposits, the total amount of withdrawals, and the total interest earned.

BankAccount
- balance: double **//updated based on monthly deposits, withdraw, and interest** - deposit: double **//tracks total amount deposited over the months** - withdraw: double **//tracks total amount withdrawn over the months** - interest: double **//tracks total interest over the months** - monthlyIntRate: double **//monthly vs annual (APR) rate. Users provide APR**

+ BankAccount() //set monthlyIntRate to 0.035/12 (monthly vs annual interest rate). Java sets all others to 0
+ BankAccount(inBalance:double, intRate:double) //divide intRate by 100, then by 12 to set monthlyIntRate
+ BankAccount(inBalance:String, intRate:String) // divide intRate by 100, then by 12 to set monthlyIntRate
+ makeDeposit(add:double): void
+ makeDeposit(add:String): void
+ makeWithdraw(sub:double): void
+ makeWithdraw(sub:String) : void
+ calcInterest(): void **//Do not include negative interest. If balance is greater than 0, calculate interest**
+ getBalance(): double
+ getDeposit(): double
+ getWithdraw(): double
+ getInterest(): double

Use the BankAccountDemo driver to test your BankAccount class. Pay close attention to the comments, there are several areas that students need to write code. Run the three Use Cases below.

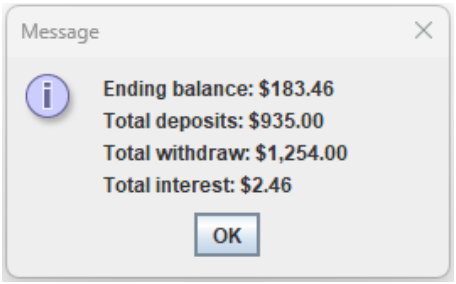
UseCase1 (Console)

```
Press 1 to use console or 2 to use GUI: 3  //ensure to test something other than
                                           //1 or 2 (ie 3) to show while pit works
Press 1 to use console or 2 to use GUI: 1
What is your starting balance: $500
What is your interest rate? (ie enter 3.5 for 3.5%): 3.5
How many months do you want to calculate? 3
==============================
Enter Month 1 deposits $400
Enter Month 1 withdraws $325
Enter Month 2 deposits $220
Enter Month 2 withdraws $710
Enter Month 3 deposits $315
Enter Month 3 withdraws $219

Ending balance: $183.46
Total deposits: $935.00
Total withdraw: $1,254.00
Total interest: $2.46
```

UseCase 2 (GUI)

Use the same numbers above for 3 months of inputs, but using option 2 (GUI). Make sure to get screen caps for each screen. Please put these in your .zip submission file.
The showDialogPane() should look something like:



UseCase3 (Console)

```
Press 1 to use console or 2 to use GUI: 1
What is your starting balance: $99
What is your interest rate? (ie enter 3.5 for 3.5%): 3.8
How many months do you want to calculate? 1
==============================
Enter Month 1 deposits $99
Enter Month 1 withdraws $345

Ending balance: $-147.00
Total deposits: $99.00
Total withdraw: $345.00
Total interest: $0.00
```

Submitting your work

For all labs you will need to provide a copy of all .java files. **No need to provide .class files. I cannot read these.** ***NOTE – For Replit, please update Main.java to another name such as TempProb.java, ProChall3.java, etc.*** In addition to your .java files, you will need to provide output files of your console. The name of the output file should match the class name and have the .txt extension such as TempProbOut.txt, ProChall3Output.txt. For GUIs such as JOptionPane, you will instead need to create screenshots. For Windows users, Snipping Tool is a great way to do this. Chromebook - Shift+Ctrl+Show Windows. Mac OS users, you can see how to take screenshots using the following url - https://support.apple.com/en-us/HT201361.