UNIVERSITY OF CALGARY

Team Assignment 5

Due February 28 at 9:50 AM *Starts Feb 17, 2018 9:00 AM* *Ends Feb 28, 2018 10:00 AM*

Topic: GUIs and Event Handling

Due Date: Wednesday, February 28 9:50 AM (no late submissions accepted)

Your first task is to ensure that the following classes from Assignment 4 are fully functional:

- BankAccount
- SavingsAccount
- Customer

Please use the provided test cases below to test the functionality of these classes before proceeding – note that there are additional test cases against which your code must run.

Next, create a setter method getCustomer() for the BankAccount class.

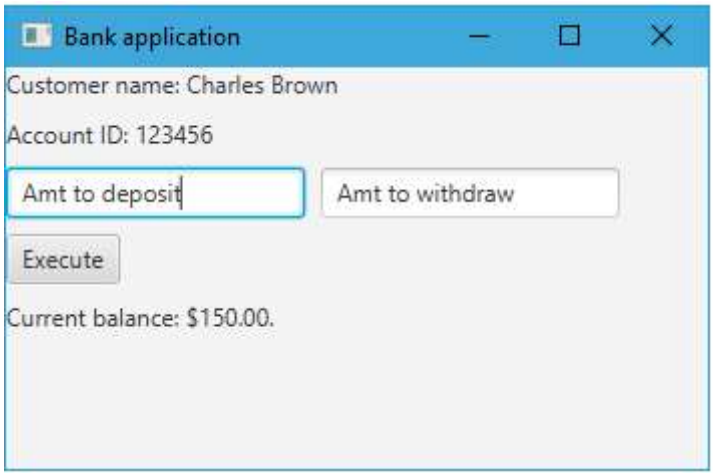
Now, create a new class BankApplication, which is a subclass of javafx.application.Application.

It should contain the following:

Instance Variables

- Customer (give the customer a name and an account ID)
- SavingsAccount (starting balance of $150.00, and the customer you have created as an instance variable)
- The following GUI controls:
    - Labels: customerNameLabel, customerIDLabel, balanceLabel
    - Text fields: depositTextField, withdrawTextField Button: executeButton

  They should be laid out as follows:



  Each label and button should be on its own line, but the text fields should be next to each other.

  (Hints: Recall that you can add one Group (such as a Hbox) to another Group (such as a VBox). TextFields can be initialized by calling the setPromptText() method)

Instance Methods -

```
public static void start(Stage primaryStage) throws Exception
```

You must also implement an event handler for the Execute button. It should handle events of type ActionEvent. This may be its own class, or an internal class.

The application must do the following:

- accept input in either (or both) text fields (You should check for valid input, and reset the text fields if the input is not valid).
- When the Execute button is clicked, the amount indicated is respectively deposited or withdrawn from the balance of the SavingsAccount, then both text fields are reset, and the balance updated. (Hint: you can read the text currently in a TextField by calling the getText() method)

Before handing in your code, use the following scenarios to manually test your code:

1. New instance of the application, deposit $100
2. New instance of the application, withdraw $100
3. New instance of the application, deposit $100, deposit $50
4. New instance of the application, withdraw $100, withdraw $50
5. New instance of the application, deposit $100, withdraw $50
6. New instance of the application, withdraw $43.72, deposit $43.72

0 % 0 of 2 topics complete

SavingsAccountTest.java

Assignment 5

Due February 28 at 9:50 AM *Ends Feb 28, 2018 10:00 AM*