Computer Science Department

COMP1331 (Second Semester2022- 2023)

Final Project *Due Date: Monday (03/07/2023) on Ritaj*

In order to let ***ABC*** Bank to serve its customers, each group will have to build an ***Account management system*** for **ABC** Bank. Each account will have an ID (*four digit integer*), Holder Name(*String*),Phone Number(long), Account type((char) s:saving, o:others) and a balance (*double*). When the program starts, it will load existing accounts information from a text file and when the program exits, it will save all changes made while running the program to that same file.

After the Account information is loaded from the file ***called account.txt (attached within the project message)***, the program will display the following main menu to the user:

Please Select an Operation (1-8):

1- Add Account

2- View All Accounts

3- Add Amount

4- Withdraw Amount

5- View Account Details

6- Modify Account

7- Close an Account

8- Exit

This menu will keep showing after each time a user selects an operation and that operation is performed. When the user selects operation 8(exit system), the program will load all changes (updates) to the file ***called Account.txt)*** and will then exit.
All the operations above should have full functionality and attached to one of the methods described below.

Your program should at least contain the following methods

distributed among 3 different classes Account,Bank and Driver:

void displayMainMenu(); //displays the main menu shown above.

void displayAccounts(); // displays all the Accounts details

boolean addAccount(Account a);
// Adds information for a new account, which should be entered by the user
// The account is added to the bank if the ID is not redundant
// Displays an error message for failed additions

boolean closeAccount(Account a);
// Removes information of an old account if it exists
// Displays a fail message if the account does not exist

Account viewAccountDetails(int choice);
// Searches for account details based on the given choice:
// 1: ID
// 2: Account holder name
// 3: Part of the name
// Returns the specified account or null if not found

void modifyAccountDetails(int choice);
// Modifies account details based on the given choice:
// 1: ID
// 2: Account holder name
// 3: Part of the name

void addAmount(double amount);
// Increases the balance variable for a certain account determined by ID

Boolean withdrawAmount(double amount);
// Decreases the balance variable if possible for a certain account determined by ID
// Prints a failure message if withdrawal is not possible
// If the account is of saving type, the account holder cannot withdraw more than 500

boolean uploadDataFile();

```
// Uploads account data from an account file
// Returns true if the operation succeeds; otherwise, displays a failure message

boolean updateDataFile();
// Saves account data updates to the accounts file
// Returns true if the operation succeeds; otherwise, displays a failure message
```

Notice that your program should handle at least 3 different exceptions.

Items that should be turned in by each group on Ritaj:

1. ***A soft copy of all your . java files.***
2. ***a complete run screenshots.***

3. you should arrange a discussion (presentation session with TA directly after submission).