CSE 101 Programming Assignment 3

Due:

Tuesday, 7-December-2021 by 23:59

Deliverables:

The following Java file should be submitted to MS Teams by the due date and time specified above. Submissions received after the deadline will be subject to the late policy described in the syllabus.

- ATM_{StudentNumber}.java

Specifications:

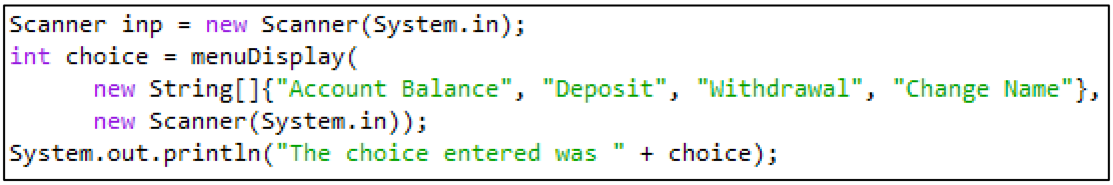
Overview: You will continue your program this week to maintain the account balance for a user of an ATM. Do not forget your headers with @author and @version information. This program will be expanded in future weeks, so be sure you understand the concepts covered in this program.

Requirements: Write a program that will simulate the user interface for an ATM. After allowing the user to login using an account number and PIN, it will allow the user to select one of four (4) options:

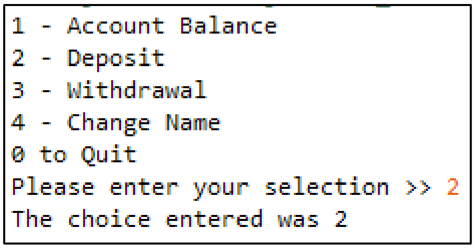
1. Display Balance
    a. Should display to the user the amount remaining in his/her account.
2. Deposit Money
    a. Adds to the balance the amount deposited
    b. Should verify the amount is valid (as described below)
3. Withdraw Money
    a. Removes the amount from the balance
    b. Should verify the amount is valid (as described below)
4. Change Name
    a. Allows the user to change his/her name on the account
    b. Will ask the user for the new name and surname he/she would like to use
5. Exit

To facilitate the execution of this program, you will add and modify (at minimum) the following methods:

1. menuDisplay(items, input)
    a. A new method to display the options based on the contents of the array
    b. Takes an array of strings and a Scanner object as parameters
    c. Displays to the output each of the names of the items in the Array preceded by a consecutive number ending with “0 to Quit”
    d. Returns an integer that represents the selection of the user

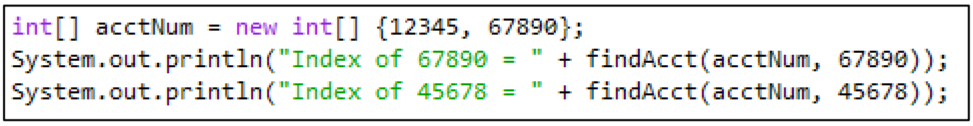


```
Scanner inp = new Scanner(System.in);
int choice = menuDisplay(
      new String[]{"Account Balance", "Deposit", "Withdrawal", "Change Name"},
      new Scanner(System.in));
System.out.println("The choice entered was " + choice);
```

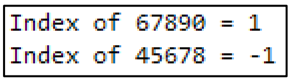


```
1 - Account Balance
2 - Deposit
3 - Withdrawal
4 - Change Name
0 to Quit
Please enter your selection >> 2
The choice entered was 2
```

2. atm(names, surnames, balances, index, input)
    - a. A method to run the atm program
    - b. It will take five parameters
        - i. String array for account holder names
        - ii. String array for account holder surnames
        - iii. Decimal array for account balances
        - iv. Index of the array for this user’s balance
        - v. Scanner object for asking the user to enter input
    - c. It will run according to the description above until the user types “0” or an invalid entry for the prompt.
    - d. NOTE: This method will use much of the main method from your previous assignment
    - e. Returns None
3. findAcct(acctNums, acctNum)
    - a. Will find the account for the given account number
    - b. Will return the index in the array of the account with that account number
    - c. Will return -1 if the account is not found



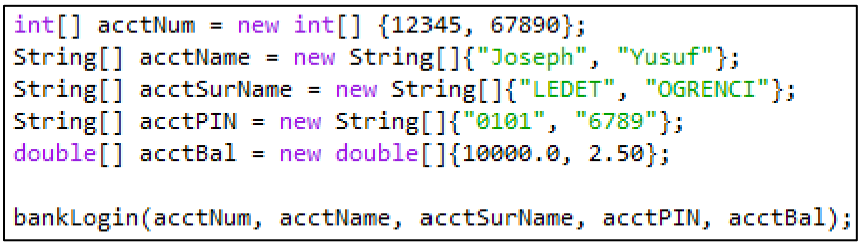
```
int[] acctNum = new int[] {12345, 67890};
System.out.println("Index of 67890 = " + findAcct(acctNum, 67890));
System.out.println("Index of 45678 = " + findAcct(acctNum, 45678));
```



```
Index of 67890 = 1
Index of 45678 = -1
```

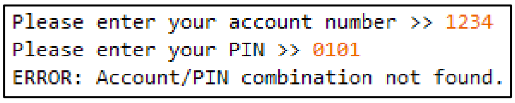
4. bankLogin(acctNums, acctNames, acctSurnames, acctPINs, acctBalances)
    - a. A method to login to the account
    - b. Will take four parameters
        - i. Array of integers for account numbers
        - ii. Array of strings for account user names
        - iii. Array of strings for account user surnames
        - iv. Array of strings for PINs (not integers as PINs can begin with 0)
        - v. Array of decimal numbers for account balances
    - c. Will give a user a prompt for their account number and a follow up for their PIN

   d. If the account is found and the PIN is correct, will then call atm method with the names array, balances array, index of the account, and the Scanner object created in this method
   e. Otherwise, will show an error message based on the examples below
   f. Returns None



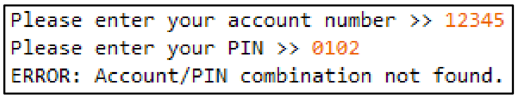
```
int[] acctNum = new int[] {12345, 67890};
String[] acctName = new String[]{"Joseph", "Yusuf"};
String[] acctSurName = new String[]{"LEDET", "OGRENCI"};
String[] acctPIN = new String[]{"0101", "6789"};
double[] acctBal = new double[]{10000.0, 2.50};

bankLogin(acctNum, acctName, acctSurName, acctPIN, acctBal);
```



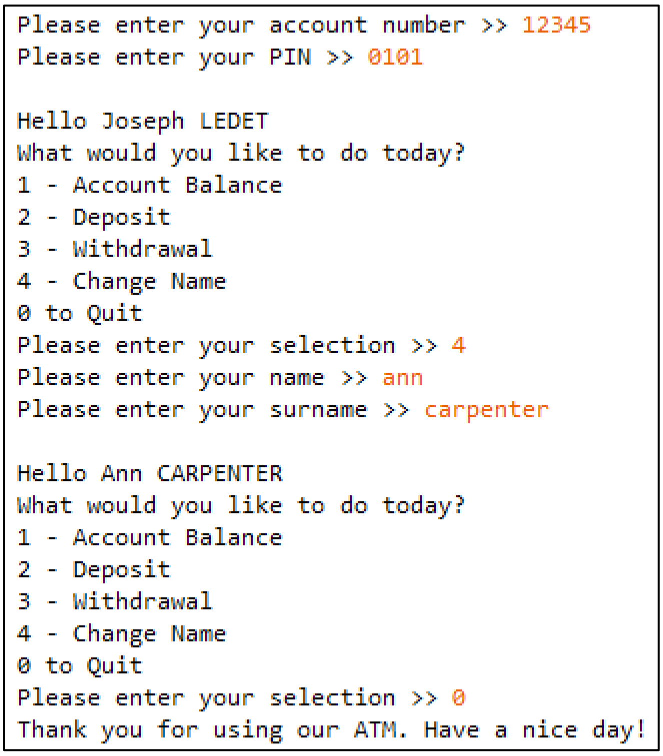
```
Please enter your account number >> 1234
Please enter your PIN >> 0101
ERROR: Account/PIN combination not found.
```

Login with incorrect Account



```
Please enter your account number >> 12345
Please enter your PIN >> 0102
ERROR: Account/PIN combination not found.
```

Login with incorrect PIN



```
Please enter your account number >> 12345
Please enter your PIN >> 0101

Hello Joseph LEDET
What would you like to do today?
1 - Account Balance
2 - Deposit
3 - Withdrawal
4 - Change Name
0 to Quit
Please enter your selection >> 4
Please enter your name >> ann
Please enter your surname >> carpenter

Hello Ann CARPENTER
What would you like to do today?
1 - Account Balance
2 - Deposit
3 - Withdrawal
4 - Change Name
0 to Quit
Please enter your selection >> 0
Thank you for using our ATM. Have a nice day!
```

Login with correct Account and PIN and showing the name change functionality.

5. isDepositValid(amount)
   a. No change from previous assignment
6. isWithdrawalValid(balance, amount)
   a. No change from previous assignment
7. moneyGiven(amount)
   a. No change from previous assignment
8. Any other methods you feel helpful can be implemented, however, these will be the only methods tested.

Design: Since your file does not require a `main` method, it should not execute. However, you should test your method using a main method in this or another file.

Code: As a user enters valid values for depositing and withdrawing money, update the balance array accordingly. Also, if the user changes his/her name, you should update the appropriate arrays. Note that when the user changes the names, the display shows the first character of the name capitalized and the entire surname in capital letters regardless of how the text was entered.

Test: You are responsible for testing your program. It is important to not rely solely on the examples presented in this Assignment description.

Grading:

MS Teams Submission: You can submit multiple times, however, we will only grade the last version that you submitted.

NOTE: If you use System.exit() in your code, you will automatically **receive 0 points** for this assignment.

Quiz: There will be a quiz based on this assignment given on 10-December at 08:40 on MS Teams and in lab. The result of this quiz will be used to determine your grade on this assignment. **Note**: if you do not submit the quiz, your score on this assignment **will be 0**.