

Name: Ananya Ghosh

Registration No.: 20MIC0063

CSI2008 – Programming in Java

L53+L54

Challenging Task 2 – Classes and Objects

Banking

Write a menu-driven java program to reflect the functioning of a bank using basic classes and objects. The bank has many customers and each customer. There are three types of accounts available in the bank - Savings account, Current account and a loan account

The following facilities are offered by the bank:

- 1) **Creating a new account** – Get the customer name, Aadhaar card number, account type (savings, current or loan) , opening balance – min Rs.500 for savings and Rs.5000 for current account and the loan amount for the loan account. The loan account should have the loan amount as a negative amount and as amount is credited to it, the remaining loan amount is adjusted and when it becomes zero, the loan status is set to closed and no further credit should be allowed in the account.

Once all the details are received from then generate the account number (random number) and display it. The account number should have six digits followed the characters 'S','C' or 'L' to indicate the type of the account. Ex: 192384C

- 2) **Debit from Savings or Current Account** – Allow transfer of money from one account to another account either between the same person's accounts or to a different customer. Check the account balance before transfer and display appropriate messages
- 3) **Credit to the accounts** – No limit for savings and current account. If crediting to loan account check the left over loan amount and display appropriate messages
- 4) **Deposit Interest** – When this functionality is triggered, calculate interest at the rate of 4% for savings account and 7.5% for current account and the corresponding amount should be credited to all the corresponding accounts.
- 5) **Display** all the account details

CODE:

```
import java.util.Scanner;
```

```
public class BankingApplication {
```

```
    public static void main(String[] args) {
```

```
        BankAccount obj = new BankAccount("SL DevCode", "SL00001");
```

```
        obj.showMenu();
```

```
    }
```

```
}
```

```
class BankAccount{
```

```
    int balance;
```

```
    int previousTransaction;
```

```
    String customerName;
```

```
    String customerId;
```

```
    BankAccount(String cname , String cid) {
```

```
        customerName = cname;
```

```
        customerId = cid;
```

```
    }
```

```
    void deposit(int amount) {
```

```
        if(amount != 0) {  
            balance = balance + amount;  
            previousTransaction = amount;  
        }  
    }
```

```
void withdraw(int amount) {  
    if(amount != 0) {  
        balance = balance - amount;  
        previousTransaction = -amount;  
    }  
}
```

```
void getPreviousTransaction() {  
    if(previousTransaction > 0) {  
        System.out.println("Deposited: " + previousTransaction);  
    }  
    else if(previousTransaction < 0) {  
        System.out.println("Withdraw: " + Math.abs(previousTransaction));  
    }  
    else {  
        System.out.println("No Transaction Occured");  
    }  
}
```

```
void showMenu() {
```

```
char option = '\0';
```

```
Scanner scanner = new Scanner(System.in);
```

```
System.out.println("Welcome " +customerName);
```

```
System.out.println("Your ID is " +customerId);
```

```
System.out.println("\n");
```

```
System.out.println("A : Check Your Balance");
```

```
System.out.println("B : Deposit");
```

```
System.out.println("C : Withdraw");
```

```
System.out.println("D : Previous Transaction");
```

```
System.out.println("E : Exit The System");
```

```
do {
```

```
System.out.println("*****  
*****");
```

```
System.out.println("Enter Your Option");
```

```
System.out.println("*****  
*****");
```

```
option = scanner.next().charAt(0);
```

```
System.out.println("\n");
```

```
switch (option) {
```

```
case 'A':
```

```
System.out.println("-----");
```

```
System.out.println("Balance = "+balance);

System.out.println("-----");

System.out.println("\n");

break;
```

case 'B':

```
System.out.println("-----");

System.out.println("Enter an amount to deposit ");

System.out.println("-----");


int amount = scanner.nextInt();

deposit(amount);

System.out.println("\n");

break;
```

case 'C':

```
System.out.println("-----");

System.out.println("Enter an amount to withdraw ");

System.out.println("-----");


int amount2 = scanner.nextInt();

withdraw(amount2);

System.out.println("\n");

break;
```

case 'D':

```
System.out.println("-----");
```

```

        getPreviousTransaction();

        System.out.println("-----");

        System.out.println("\n");

        break;

    case 'E' :

        System.out.println("=====
=====");

        break;

    default:

        System.out.println("Invalid Option!! Please Enter Correct
Option...");

        break;

    }

}

while(option != 'E');

    System.out.println("Thank You for Using our Services.....!!");

}

}

```

OUTPUT

BankingApplication.java - Java - Visual Studio Code

```
1 import java.util.Scanner;
2
3 public class BankingApplication {
4
5     Run | Debug
6     public static void main(String[] args) {
```

PS C:\Users\anany\Desktop\VIT\semester 4\java> cd "c:\Users\anany\Desktop\VIT\semester 4\java\" ; if (\$?) { javac BankingApplication.java } ; if (\$?) { java BankingApplication }
Welcome SL DevCode
Your ID is SL00001

A : Check Your Balance
B : Deposit
C : Withdraw
D : Previous Transaction
E : Exit The System
=====

BankingApplication.java - Java - Visual Studio Code

```
1 import java.util.Scanner;
2
3 public class BankingApplication {
4
5     Run | Debug
6     public static void main(String[] args) {
```

E : Exit The System
=====

Enter Your Option
=====

A

Balance = 0
=====

Enter Your Option
=====

This screenshot shows the Visual Studio Code interface with the `BankingApplication.java` file open. The code defines a `BankingApplication` class with a `main` method that uses `Scanner` to take user input. The terminal output shows the program running and prompting the user to enter an amount to deposit. The user has entered `1000`, and the program has calculated the balance as `1000`. The Explorer sidebar on the left lists various Java files, including `BankingApplication.java` and several utility classes like `AreaofCircle.java`, `Array.java`, and `Factorial.java`. The status bar at the bottom indicates the file is at line 9, column 1, with 4 spaces, UTF-8 encoding, and CRLF line endings.

```
BankingApplication.java 1 X
1  import java.util.Scanner;
2
3  public class BankingApplication {
4
5      Run | Debug
      public static void main(String[] args) {
6
7          B
8          -----
9          Enter an amount to deposit
10         -----
11         1000
12
13         =====
14         Enter Your Option
15         =====
16         A
17
18         -----
19         Balance = 1000
20         -----
21
22         =====
23         Enter Your Option
24         =====
```

This screenshot shows the Visual Studio Code interface with the `BankingApplication.java` file open. The code is the same as in the first screenshot. The terminal output shows the program running and prompting the user to enter an amount to withdraw. The user has entered `200`, and the program has calculated the balance as `800`. The Explorer sidebar on the left lists various Java files, including `BankingApplication.java` and several utility classes like `AreaofCircle.java`, `Array.java`, and `Factorial.java`. The status bar at the bottom indicates the file is at line 9, column 1, with 4 spaces, UTF-8 encoding, and CRLF line endings.

```
BankingApplication.java 1 X
1  import java.util.Scanner;
2
3  public class BankingApplication {
4
5      Run | Debug
      public static void main(String[] args) {
6
7          C
8          -----
9          Enter an amount to withdraw
10         -----
11         200
12
13         =====
14         Enter Your Option
15         =====
16         A
17
18         -----
19         Balance = 800
20         -----
21
22         =====
23         Enter Your Option
24         =====
```


