Program 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

Develop a Java Program to create a class Bank that maintains two kinds of account for its customers, one Called Savings account and the other current account. The savings account operates compound between interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a sofwice change is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes (we acct and sav-acct to make them more specific to their order to achieve the following tasks:

```
a) Accept deposit from customer and update the balance
b) Dioplay the balance
c) Compute and deposit interest
d) Permit withdrawal and update the balance
check for the minimum balance, impose penalty if
necessary and update the bollance
imposet java util . Scannes;
abstract class Account 1
  storing custometiName, accountNumber;
  double balance;
  Account (String austomer Name, String accountNumber,
           double initial Balance) {
     this . Customer Name = customer Name;
     this. accountNumber = accountNumber;
   4. this balance = initial Balance;
   abstract void deposit (double amount);
 abstract void displayBalance();
abstract void withdraw (double amount);
class SanAcct extends Account
   double interest Rate;
   SavAcct (String customer Name, String account Number,
             double initial Balance) {
       Super (CustomeriName, account Number, initial Balance);
       this interestRate = interestRate;
   void deposit (double amount)
        balance += amount;
    void display Balance() (
       system out println ("savings Balance:
```

if (arount z=balance) balance = = amount void withdraw (double Account account if (type equal 5 ("savings")) [System out println ("Initial balance and interest rate:"); void dis. compute And Deposit Interest () ? balance + = balance * interestRate /100; account = new SavAcct (name, number, &conner -nextDouble). Boarren next Double()); System out . println ("Initial bolance :"); class Confect extends Account (Static final double MIN-BALANCE - 1000, SERVICE - CHARGE account = new Cuntect (name, number, scanner rext Double) current (string customername, string account Number, while (true) [System out println ("Ant-Deposit 2 Display Balance double initialBalance) [super (customer Name, account Number, initial Bolance) 3. Withdraw 4. Interest 5 Exit); int choice = Scannes .nextInt(); void deposit (double amount) [Switch (choice) Case 1: account deposit (Manner next Double ()); balance + = amount; System. out frintly ("currount Balance;" + balance); void display Balance () [Case a: account display Balance (); (ase 3: account withdraw (scanner next couble()); void withdraw (double amount) { (amount <= balance Case 4: if (account instance) SanAct) (balance = amount; ([savetet) account). Computer And Deposit Interest (); if (balance < MIN_BALANCE) balance break; class Bank f Case 5 : public static void main (string [] angs)[neturn; Sanner Blannett = new Slanner (Eystem.in); System out println ("Enter account type (savings) Enter account type (savings/cunsent): cushent):); String type = Branner next Line (); string name - beanter hest System out println ("Enter account name": "); Enter account name; Staing name = sconner. nextline (); System out . I sintly ("Enter account Number:"); Enter account Number: storing number = scanner nextline(); 1235DFG

```
import java.util.Scanner;
abstract class Account {
  String customerName, accountNumber;
  double balance;
  Account(String customerName, String accountNumber, double initialBalance) {
    this.customerName = customerName;
    this.accountNumber = accountNumber;
    this.balance = initialBalance;
  }
  abstract void deposit(double amount);
  abstract void displayBalance();
  abstract void withdraw(double amount);
}
class SavAcct extends Account {
  double interestRate;
  SavAcct(String customerName, String accountNumber, double initialBalance, double
interestRate) {
    super(customerName, accountNumber, initialBalance);
    this.interestRate = interestRate;
  }
  void deposit(double amount) {
    balance += amount;
  }
  void displayBalance() {
```

```
System.out.println("Savings Balance: " + balance);
  }
  void withdraw(double amount) {
    if (amount <= balance) balance -= amount;
  }
  void computeAndDepositInterest() {
    balance += balance * interestRate / 100;
  }
}
class CurAcct extends Account {
  static final double MIN BALANCE = 1000, SERVICE CHARGE = 50;
  CurAcct(String customerName, String accountNumber, double initialBalance) {
    super(customerName, accountNumber, initialBalance);
  }
  void deposit(double amount) {
    balance += amount;
  }
  void displayBalance() {
    System.out.println("Current Balance: " + balance);
  }
  void withdraw(double amount) {
    if (amount <= balance) {
       balance -= amount;
```

```
if (balance < MIN_BALANCE) balance -= SERVICE_CHARGE;
    }
}
class Bank {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter account type (savings/current): ");
    String type = scanner.nextLine();
    System.out.println("Enter customer name: ");
    String name = scanner.nextLine();
    System.out.println("Enter account number: ");
    String number = scanner.nextLine();
    Account account;
    if (type.equals("savings")) {
       System.out.println("Initial balance and interest rate: ");
       account = new SavAcct(name, number, scanner.nextDouble(), scanner.nextDouble());
    } else {
       System.out.println("Initial balance: ");
       account = new CurAcct(name, number, scanner.nextDouble());
    }
    while (true) {
       System.out.println("\n1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit");
       int choice = scanner.nextInt();
       switch (choice) {
```

```
case 1: account.deposit(scanner.nextDouble());
          break;
          case 2: account.displayBalance();
          break;
          case 3: account.withdraw(scanner.nextDouble());
          break;
          case 4: if (account instanceof SavAcct) ((SavAcct)
account).computeAndDepositInterest();
          break;
          case 5:
          return;
  }
}
D:\24BMSCE>javac Bank.java
D:\24BMSCE>java Bank
Enter account type (savings/current):
savings
Enter customer name:
anu rai
Enter account number:
123786645087301
Initial balance and interest rate:
5000
50
1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
200
1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
2
Savings Balance: 5200.0
1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
3
100
1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
Savings Balance: 7650.0
1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
```