Java Program 5

import java.util.Scanner;

class Account {

protected String customerName;

protected int accountNumber;

protected double balance;

public Account(String customerName, int accountNumber, double balance) {

this.customerName = customerName;

this.accountNumber = accountNumber;

this.balance = balance;

}

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

System.out.println("Deposited: " + amount);

} else {

System.out.println("Invalid deposit amount");

}

}

public void displayBalance() {

System.out.println("Balance: " + balance);

}

}

class SavAcct extends Account {

private double interestRate;

public SavAcct(String customerName, int accountNumber, double balance, double interestRate) {

super(customerName, accountNumber, balance);

this.interestRate = interestRate;

}

public void computeAndDepositInterest() {

double interest = balance \* (interestRate / 100);

balance += interest;

System.out.println("Interest added: " + interest);

}

public void withdraw(double amount) {

if (amount <= balance) {

balance -= amount;

System.out.println("Withdrawn: " + amount);

} else {

System.out.println("Insufficient balance for withdrawal");

}

}

}

class CurAcct extends Account {

private double minimumBalance;

private double serviceCharge;

public CurAcct(String customerName, int accountNumber, double balance, double minimumBalance, double serviceCharge) {

super(customerName, accountNumber, balance);

this.minimumBalance = minimumBalance;

this.serviceCharge = serviceCharge;

}

public void withdraw(double amount) {

if (amount <= balance) {

balance -= amount;

System.out.println("Withdrawn: " + amount);

if (balance < minimumBalance) {

balance -= serviceCharge;

System.out.println("Service charge imposed: " + serviceCharge);

}

} else {

System.out.println("Insufficient balance for withdrawal");

}

}

}

public class Bank {

public static void main(String[] args) {

System.out.println("Aparna Sankar, 1BM23CS047");

Scanner sc = new Scanner(System.in);

// Create a savings account

SavAcct savAcc = new SavAcct("Alice", 12345, 1000, 5);

// Create a current account

CurAcct curAcc = new CurAcct("Bob", 67890, 2000, 500, 50);

System.out.println("Choose Account Type:\n1. Savings Account\n2. Current Account");

int choice = sc.nextInt();

switch (choice) {

case 1:

System.out.println("Savings Account Selected");

savAcc.deposit(500);

savAcc.computeAndDepositInterest();

savAcc.withdraw(300);

savAcc.displayBalance();

break;

case 2:

System.out.println("Current Account Selected");

curAcc.deposit(500);

curAcc.withdraw(1800);

curAcc.displayBalance();

break;

default:

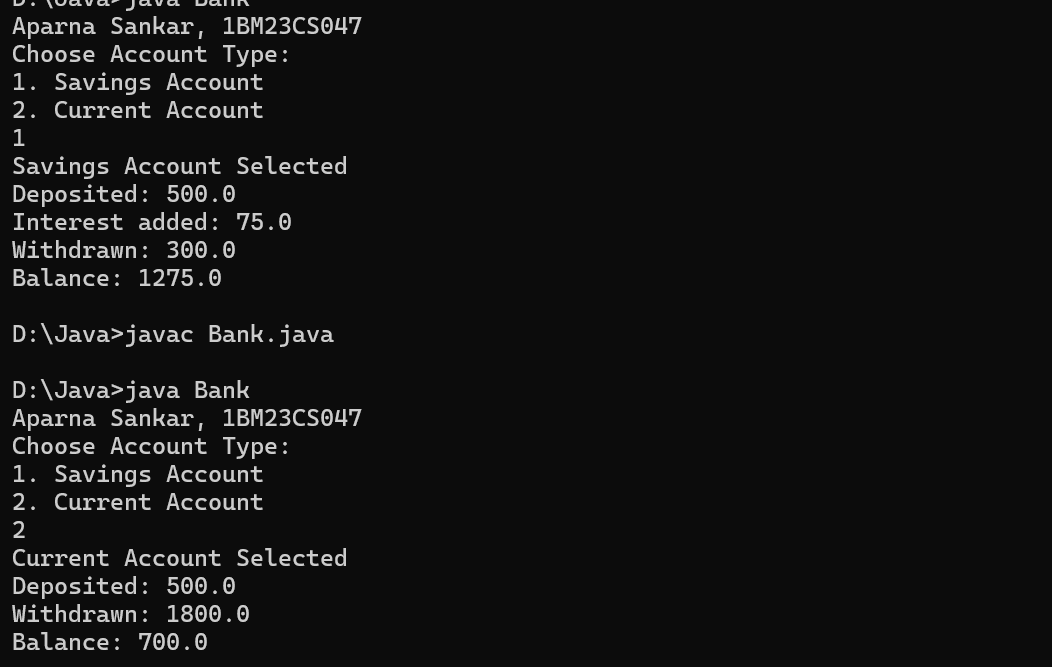
System.out.println("Invalid choice");

}

sc.close();

}

}



import java.util.Scanner;

class Account {

String customerName;

String accountNumber;

double balance;

Account(String customerName, String accountNumber, double initialBalance) {

this.customerName = customerName;

this.accountNumber = accountNumber;

this.balance = initialBalance;

}

void deposit(double amount) {

balance += amount;

System.out.println("Deposited: " + amount);

}

void displayBalance() {

System.out.println("Current Balance: " + balance);

}

void withdraw(double amount) {

if (amount > balance) {

System.out.println("Insufficient balance!");

} else {

balance -= amount;

System.out.println("Withdrew: " + amount);

}

}

}

class SavingsAccount extends Account {

double interestRate;

SavingsAccount(String customerName, String accountNumber, double initialBalance, double interestRate) {

super(customerName, accountNumber, initialBalance);

this.interestRate = interestRate;

}

void computeAndDepositInterest() {

double interest = balance \* (interestRate / 100);

deposit(interest);

System.out.println("Interest deposited: " + interest);

}

}

class CurrentAccount extends Account {

double minimumBalance;

double serviceCharge;

CurrentAccount(String customerName, String accountNumber, double initialBalance, double minimumBalance, double serviceCharge) {

super(customerName, accountNumber, initialBalance);

this.minimumBalance = minimumBalance;

this.serviceCharge = serviceCharge;

}

@Override

void withdraw(double amount) {

super.withdraw(amount);

if (balance < minimumBalance) {

balance -= serviceCharge;

System.out.println("Minimum balance not maintained. Service charge applied: " + serviceCharge);

}

}

}

public class Bank {

public static void main(String[] args) {

System.out.println("Aparna Sankar, 1BM23CS047");

Scanner sc = new Scanner(System.in);

Account account;

System.out.print("Enter customer name: ");

String name = sc.nextLine();

System.out.print("Enter account number: ");

String accNum = sc.nextLine();

System.out.print("Choose account type (1) Savings, 2) Current): ");

int accType = sc.nextInt();

if (accType == 1) {

System.out.print("Enter initial balance: ");

double balance = sc.nextDouble();

System.out.print("Enter interest rate: ");

double interestRate = sc.nextDouble();

account = new SavingsAccount(name, accNum, balance, interestRate);

} else {

System.out.print("Enter initial balance: ");

double balance = sc.nextDouble();

System.out.print("Enter minimum balance: ");

double minBalance = sc.nextDouble();

System.out.print("Enter service charge: ");

double serviceCharge = sc.nextDouble();

account = new CurrentAccount(name, accNum, balance, minBalance, serviceCharge);

}

int choice;

do {

System.out.println("\nMenu:");

System.out.println("1. Deposit");

System.out.println("2. Display Balance");

System.out.println("3. Withdraw");

System.out.println("4. Compute Interest (Savings Only)");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

choice = sc.nextInt();

switch (choice) {

case 1:

System.out.print("Enter amount to deposit: ");

double depositAmount = sc.nextDouble();

account.deposit(depositAmount);

break;

case 2:

account.displayBalance();

break;

case 3:

System.out.print("Enter amount to withdraw: ");

double withdrawAmount = sc.nextDouble();

account.withdraw(withdrawAmount);

break;

case 4:

if (account instanceof SavingsAccount) {

((SavingsAccount) account).computeAndDepositInterest();

} else {

System.out.println("Interest computation is not applicable for current accounts.");

}

break;

case 5:

System.out.println("Thank you for banking with us!");

break;

default:

System.out.println("Invalid choice! Try again.");

}

} while (choice != 5);

sc.close();

}

}

