Week 5

import java.util.Scanner;

class Account {

String customerName;

int accountNumber;

String accountType;

double balance;

Account(String customerName, int accountNumber, String accountType) {

this.customerName = customerName;

this.accountNumber = accountNumber;

this.accountType = accountType;

this.balance = 0.0;

}

void deposit(double amount) {

balance += amount;

System.out.println("Amount deposited: " + amount);

}

void displayBalance() {

System.out.println("Customer name: " + customerName);

System.out.println("Account number: " + accountNumber);

System.out.println("Type of Account: " + accountType + " account");

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

}

}

class SavAcct extends Account {

SavAcct(String customerName, int accountNumber) {

super(customerName, accountNumber, "Savings");

}

void computeInterest(double rate, int time) {

balance += balance \* Math.pow(1 + rate / 100, time) - balance;

}

void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Amount withdrawn: " + amount);

} else {

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

}

}

}

class CurAcct extends Account {

static final double MIN\_BALANCE = 500.0;

static final double PENALTY = 50.0;

CurAcct(String customerName, int accountNumber) {

super(customerName, accountNumber, "Current");

}

void withdraw(double amount) {

if (balance - amount >= MIN\_BALANCE) {

balance -= amount;

System.out.println("Amount withdrawn: " + amount);

} else {

System.out.println("Minimum balance requirement not met. Cannot withdraw.");

}

}

void checkMinimumBalance() {

if (balance < MIN\_BALANCE) {

balance -= PENALTY;

System.out.println("Penalty of " + PENALTY + " imposed due to insufficient balance.");

}

}

}

public class Bank {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Ruqaiyya Mahreen 1BM23EE044");

Account[] accounts = new Account[10];

int count = 0;

while (true) {

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

String customerName = sc.next();

System.out.println("Enter account Number: ");

int accountNumber = sc.nextInt();

System.out.println("Enter the type of account (1 for Savings, 2 for Current): ");

int type = sc.nextInt();

if (type == 1) {

accounts[count] = new SavAcct(customerName, accountNumber);

} else {

accounts[count] = new CurAcct(customerName, accountNumber);

}

count++;

while (true) {

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

System.out.println("1. Deposit\n2. Withdraw\n3. Compute interest for SavingsAccount\n4. Display account details\n5. Exit");

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

int choice = sc.nextInt();

if (choice == 5) break;

switch (choice) {

case 1:

System.out.println("Enter the deposit amount: ");

double depositAmount = sc.nextDouble();

accounts[count - 1].deposit(depositAmount);

break;

case 2:

System.out.println("Enter the withdrawal amount: ");

double withdrawAmount = sc.nextDouble();

if (type == 1) {

((SavAcct) accounts[count - 1]).withdraw(withdrawAmount);

} else {

((CurAcct) accounts[count - 1]).withdraw(withdrawAmount);

((CurAcct) accounts[count - 1]).checkMinimumBalance();

}

break;

case 3:

if (type == 1) {

System.out.println("Enter the interest rate and time in years: ");

double rate = sc.nextDouble();

int time = sc.nextInt();

((SavAcct) accounts[count - 1]).computeInterest(rate, time);

System.out.println("Interest computed and added to balance.");

} else {

System.out.println("No interest computation for Current Account.");

}

break;

case 4:

accounts[count - 1].displayBalance();

break;

}

}

System.out.println("Do you want to add another account? (yes/no): ");

String response = sc.next();

if (response.equalsIgnoreCase("no")) break;

}

sc.close();

}

}



