Program-6 Write a Program to simulate a Bank Management System

import java.io.\*;

import java.util.ArrayList;

interface Operations {

    float deposit(float amount);

    float withdrawl(float amount);

    float balance();

}

class Current implements Operations {

    static int idCreater = 6;

    private int id;

    private float balance;

    Current(float amount) {

        this.balance = amount;

        idCreater++;

        this.id = idCreater;

    }

    int getId() {

        return this.id;

    }

    public float deposit(float amount) {

        this.balance += amount;

        return balance;

    }

    public float withdrawl(float amount) {

        if (balance >= amount) {

            balance -= amount;

            return balance;

        }

        return -1;

    }

    public float balance() {

        return this.balance;

    }

}

class Saving implements Operations {

    static int idCreater = 0;

    private int id;

    private float balance;

    private final float interest = 3.0f;

    Saving(float amount) {

        this.balance = amount;

        idCreater++;

        this.id = idCreater;

    }

    private float interest(float amount) {

        return (interest \* this.balance) / 100;

    }

    int getId() {

        return this.id;

    }

    public float deposit(float amount) {

        this.balance += amount;

        this.balance += interest(this.balance);

        return this.balance;

    }

    public float withdrawl(float amount) {

        if (this.balance >= amount) {

            this.balance -= amount;

            return this.balance;

        }

        return -1;

    }

    public float balance() {

        return this.balance;

    }

}

class AccountHandler {

    static int totalAccounts = 0;

    static ArrayList<Current> currentAccounts = new ArrayList<>();

    static ArrayList<Saving> savingAccounts = new ArrayList<>();

    int openCurrentAccount(float amount) {

        if(amount < 10000){

            System.out.println("Minimum amount is 10000/-");

            return -1;

        }

        if(totalAccounts > 10){

            System.out.println("Max Accounts Limit reached");

            return -1;

        }

        Current account = new Current(amount);

        currentAccounts.add(account);

        totalAccounts++;

        return account.getId();

    }

    int openSavingAccount(float amount) {

        if(amount < 1000){

            System.out.println("Minimum amount is 1000/-");

            return -1;

        }

        if(totalAccounts > 10){

            System.out.println("Max Accounts Limit reached");

            return -1;

        }

        Saving account = new Saving(amount);

        savingAccounts.add(account);

        totalAccounts++;

        return account.getId();

    }

    int closeCurrentAccount(int id) {

        for (Current account : currentAccounts) {

            if (account.getId() == id) {

                currentAccounts.remove(account);

                totalAccounts--;

                return 1;

            }

        }

        return -1;

    }

    int closeSavingAccount(int id) {

        for (Saving account : savingAccounts) {

            if (account.getId() == id) {

                savingAccounts.remove(account);

                totalAccounts--;

                return 1;

            }

        }

        return -1;

    }

    void deposit(int id, float amount){

        float res = -1;

        for(Current account: currentAccounts){

            if(account.getId() == id){

                res = account.deposit(amount);

            }

        }

        for(Saving account : savingAccounts){

            if(account.getId() == id){

                res = account.deposit(amount);

            }

        }

        if(res == -1) System.out.println("Transaction Failed");

        else{

            System.out.println("Transaction Completed");

            System.out.println("Your Balance: " + res);

        }

    }

    float withdrawl(int id, float amount){

        float res = -1;

        for(Current account: currentAccounts){

            if(account.getId() == id){

                res = account.withdrawl(amount);

            }

        }

        for(Saving account: savingAccounts){

            if(account.getId() == id){

                res = account.withdrawl(amount);

            }

        }

        if(res != -1){

            System.out.println("Transaction Completed");

            System.out.println("Your Balance: " + res);

        return res;

        }

    return res;

    }

    float getBalance(int id) {

        for (Current account : currentAccounts) {

            if (account.getId() == id)

                return account.balance();

        }

        for (Saving account : savingAccounts) {

            if (account.getId() == id)

                return account.balance();

        }

        return -1;

    }

}

class Bank {

    static BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

    static AccountHandler accountHandler = new AccountHandler();

    public static void main(String[] args) throws IOException {

        while(true){

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

            System.out.println("What to Do?");

            System.out.println("1: Open Account");

            System.out.println("2: Close Account");

            System.out.println("3: Access Account");

            System.out.println("4: Exit");

            System.out.print("Choice: ");

            int choice = Integer.parseInt(br.readLine());

            if(choice == 4) break;

            menuOperator(choice);

        }

    }

    static void menuOperator(int choice) throws IOException {

        if(choice == 1){

            System.out.println("Type of Account?");

            System.out.println("1: Saving");

            System.out.println("2: Current");

            int type = Integer.parseInt(br.readLine());

            if(type != 1 && type != 2){

                System.out.println("Input valid type");

                return;

            }

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

            float amount = Float.parseFloat(br.readLine());

            int id;

            if(type == 1) id = accountHandler.openSavingAccount(amount);

            else id = accountHandler.openCurrentAccount(amount);

            if(id != -1) System.out.println("Your Account number is: " + id);

        }

        else if(choice == 2){

            System.out.println("Enter the id of account to be deleted");

            int id = Integer.parseInt(br.readLine());

            if(accountHandler.closeCurrentAccount(id) == 1 || accountHandler.closeSavingAccount(id) == 1){

                System.out.println("Account Deleted Successfully");

            }

        else{

          System.out.println("Account Not Found");

        }

        }

        else if(choice == 3){

            System.out.println("Enter the id of your account");

            int id = Integer.parseInt(br.readLine());

        boolean flag = false;

        for(Current account: accountHandler.currentAccounts){

          if(account.getId() == id) flag = true;

            }

        for(Saving account: accountHandler.savingAccounts){

          if(account.getId() == id) flag = true;

            }

        if(flag == false){

          System.out.println("Account not Found");

          return;

        }

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

            System.out.println("2: Withdrawl");

            System.out.println("3: Check Balance");

            System.out.println("Choose: ");

            int opr = Integer.parseInt(br.readLine());

            if(opr == 1){

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

                float amount = Float.parseFloat(br.readLine());

                accountHandler.deposit(id, amount);

            }

            else if(opr == 2){

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

                float amount = Float.parseFloat(br.readLine());

            float res = accountHandler.withdrawl(id, amount);

                if(res == -1) System.out.println("Insufficient Balance");

            }

            else if(opr == 3){

                float res = accountHandler.getBalance(id);

                if(res != -1) System.out.println("Your balance: " + res);

                else System.out.println("Account Not Found");

            }

        }

        else{

            System.out.println("Input valid Choice");

        }

    }

}











