import javax.swing.\*;

public class Conta {

    // atributos

    private double saldo;

    // contrutor zerado

    public Conta(){

        this(0.0);

    }

    //contrutor com valores

    public Conta(double saldo){

        this.saldo = saldo;

    }

    // getters e setters

    public double getSaldo() {

        return saldo;

    }

    public void setSaldo(double saldo) {

        this.saldo = saldo;

    }

    //metodos

    public void creditar(double valor){

        // double valor = Double.parseDouble(JOptionPane.showInputDialog("Digite o valor a creditar: "));

        this.setSaldo(this.getSaldo()+ valor);

        JOptionPane.showMessageDialog(null,"Valor creditado com sucesso!!! ");

    }

    public void debitar(double valor){

        // valor = Double.parseDouble(JOptionPane.showInputDialog("Digite o valor a debitar: "));

        if (this.getSaldo()>= valor){

            this.setSaldo(this.getSaldo() - valor );

            JOptionPane.showMessageDialog(null,"Valor debitado com sucesso!!! ");

        }else{

            //JOptionPane.showMessageDialog(null,"Seu saldo é: " + String.valueOf(this.getSaldo()));

            JOptionPane.showMessageDialog(null,"Saldo suficiente para isso!!!");

        }

        }

    public void atualizarSaldo(){

        if (this.getSaldo()<= 0){

            double valorJuros = (this.getSaldo()\*0.08);

            JOptionPane.showMessageDialog(null,"Saldo negativo e juros: " + String.valueOf(valorJuros));

        }

    }

}

import javax.swing.JOptionPane;

public class ContaCorrente extends Conta {

    // atributos

    private double limiteEspecial;

    //contrutor zerado

    public ContaCorrente(){

        this(0.0,0.0);

    }

    //contrutor com valor

    public ContaCorrente(double saldo, double plimiteEspecial){

        super(saldo);

        this.limiteEspecial = plimiteEspecial;

    }

    // getters e setters

    public double getLimiteEspecial() {

        return limiteEspecial;

    }

    public void setLimiteEspecial(double limiteEspecial) {

        this.limiteEspecial = limiteEspecial;

    }

    public void debitar(double valor){

        // double valor = Double.parseDouble(JOptionPane.showInputDialog("Digite o valor a debitar: "));

        if (this.getSaldo()+ this.getLimiteEspecial()>= valor){

            this.setSaldo(this.getSaldo()- valor);

        } else{

            JOptionPane.showMessageDialog(null,"Seu saldo ficará negativo em : R$ " + String.valueOf((this.getSaldo()+ limiteEspecial)- valor));

        }

    }

}

public class ContaPoupanca extends Conta {

    //atributos

    private double reajusteMensal;

    // contrutores sem valor

    public ContaPoupanca(){

        this(0.0,0.0);

    }

    // contrutores com valores

    public ContaPoupanca(double saldo, double preajusteMensal){

        super(saldo);

        this.reajusteMensal = preajusteMensal;

    }

// getter e setter

    public double getReajusteMensal() {

        return reajusteMensal;

    }

    public void setReajusteMensal(double preajusteMensal) {

        this.reajusteMensal = preajusteMensal;

    }

public void atualizarSaldo(){

}

}

import javax.swing.JOptionPane;

public class Movimentacao {

    /\*\*

     \* @param args

     \*/

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        ContaCorrente cc1 = new ContaCorrente(0000.00, 50.00);

        ContaPoupanca cp1 = new ContaPoupanca(1000.00, 50.00);

        int opPro, opOperacao = 0;

        do{

            opPro =Integer.parseInt(JOptionPane.showInputDialog("Digite:\n1 - Conta Corrente \n2 - Conta Poupança \n0 - Sair"));

            if (opPro != 0)

                do{

                    opOperacao =Integer.parseInt(JOptionPane.showInputDialog("Digite:\n1 - Depositar \n2 - Sacar \n3 - Consultar \n4 - Atualizar \n0 - Sair"));

                    switch(opPro){ // Abertura da estrutura de switch-case

                    case 1: // Conta Corrente

                        if (opOperacao == 1){//depositar

                            double valor = Integer.parseInt(JOptionPane.showInputDialog("Digite Valor a Creditar:"));

                            cc1.creditar(valor);

                        }

                        if (opOperacao == 2){//debitar

                            double valor = Double.parseDouble(JOptionPane.showInputDialog("Digite Valor a Debitar:"));

                            cc1.debitar(valor);

                        }

                        if (opOperacao == 3){//Consultar

                            JOptionPane.showMessageDialog(null, "O seu saldo é: " + String.valueOf(cc1.getSaldo()));

                        }

                        if (opOperacao == 4){//atualizar valor

                            //JOptionPane.showMessageDialog(null, "O seu saldo é: " + String.valueOf(cc1.getSaldo()));

                            cc1.atualizarSaldo();

                        }

                        break;

                    case 2: // Conta Poupanca

                        if (opOperacao == 1){//depositar

                            double valor = Integer.parseInt(JOptionPane.showInputDialog("Digite Valor a Creditar:"));

                            cp1.creditar(valor);

                        }

                        if (opOperacao == 2){//debitar

                            double limiteEspecial = Double.parseDouble(JOptionPane.showInputDialog("Digite Valor a Debitar:"));

                            cp1.debitar(limiteEspecial);

                        }

                        if (opOperacao == 3){//Consultar

                            JOptionPane.showMessageDialog(null,"ok");

                        }

                        break;

                    case 0:

                        JOptionPane.showMessageDialog(null,"Obrigado por usar a nossa agencia");

                        break;

                    }

                }while(opOperacao != 0);

        }while(opPro != 0);

    }

}