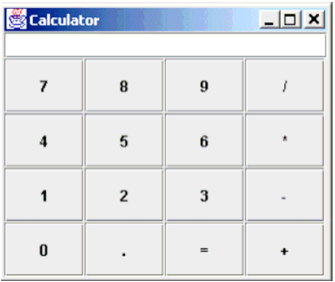
**Nome: Pedro Henique Padilha Portella da Cruz**

**1) Implemente a seguinte calculadora.**

****

CLASSE MAIN

package calculator;

//importando todas as funções das bibliotecas swing e awt

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Calculator {

private JFrame janela; //cria componente JFrame

private JPanel painelPrincipal; //cria componente JPainel

private Label label1, invisivel01, invisivel02; // cria componentes label

private TextField display; // cria componentes TextField

private double memoria = 0;

private boolean ponto = false, fazendoOperacao = false, apagarParaOProximoNumero = false;

private String operacao = "";

public static void main(String[] args) {

new Calculator().montaTela(); //define método montatela em Classe

//fatec, responsável pela interface gráfica

}

private void montaTela() {//chama os métodos que formarão a tela

preparaJanela();

preparaPainelPrincipal();

preparaLabel();

preparaText();

preparaBotao07();

preparaBotao08();

preparaBotao09();

preparaBotaoDividir();

preparaBotao04();

preparaBotao05();

preparaBotao06();

preparaBotaoMultiplicar();

preparaBotao01();

preparaBotao02();

preparaBotao03();

preparaBotaoSubtrair();

preparaBotao00();

preparaBotaoPonto();

preparaBotaoIgual();

preparaBotaoSomar();

//preparaBotaoSair();

mostraJanela();

}

private void preparaJanela() {

janela = new JFrame("Calculator");

// como é necessário utilizar o Jframe no método mostraJanela é necessário que

// janela seja um atributo ao invés de uma variável local

janela.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); //indica ao nosso frame que a aplicação deve ser terminada quando o usuário fechar a janela

}

private void mostraJanela() {

janela.pack(); // serve para organizar os componentes do frame para que eles ocupem o menor espaço possível.

janela.setSize(350, 300); //dimensões do Frame

janela.setVisible(true);// recebe um boolean indicando se queremos que

//a janela esteja visível ou não.

}

// prepara o Painel que receberá os botões e adiciona o Painel no Frame

private void preparaPainelPrincipal() {

painelPrincipal = new JPanel();

painelPrincipal.setLayout(new BoxLayout(painelPrincipal, BoxLayout.PAGE\_AXIS));

// layout em forma de tabela com 3 linhas e 2 colunas;

painelPrincipal.setLayout(new GridLayout(5, 4));

janela.add(painelPrincipal);

}

// insere texto no label1 e adiciona o label no painel;

private void preparaLabel() {

label1 = new Label("");

painelPrincipal.add(label1);

}

;

// insere texto no label , espaços aos TextFields e adiciona os componentes no painel;

private void preparaText() {

display = new TextField("0");

invisivel01 = new Label("");

invisivel02 = new Label("");

painelPrincipal.add(invisivel01);

painelPrincipal.add(invisivel02);

painelPrincipal.add(display);

}

;

//criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotao00() {

JButton Botao00 = new JButton("0");

Botao00.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (display.getText().equals("0")) {

display.setText(String.valueOf("0"));

} else if (apagarParaOProximoNumero == true) {

display.setText(String.valueOf("0"));

apagarParaOProximoNumero = false;

} else {

display.setText(display.getText() + String.valueOf("0"));

}

}

});

painelPrincipal.setLocation(50, 150);//posição do painel no frame

painelPrincipal.add(Botao00);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

//criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotao01() {

JButton Botao01 = new JButton("1");

Botao01.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (display.getText().equals("0")) {

display.setText(String.valueOf("1"));

} else if (apagarParaOProximoNumero == true) {

display.setText(String.valueOf("1"));

apagarParaOProximoNumero = false;

} else {

display.setText(display.getText() + String.valueOf("1"));

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(Botao01);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

//criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotao02() {

JButton Botao02 = new JButton("2");

Botao02.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (display.getText().equals("0")) {

display.setText(String.valueOf("2"));

} else if (apagarParaOProximoNumero == true) {

display.setText(String.valueOf("2"));

apagarParaOProximoNumero = false;

} else {

display.setText(display.getText() + String.valueOf("2"));

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(Botao02);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

//criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotao03() {

JButton Botao03 = new JButton("3");

Botao03.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (display.getText().equals("0")) {

display.setText(String.valueOf("3"));

} else if (apagarParaOProximoNumero == true) {

display.setText(String.valueOf("3"));

apagarParaOProximoNumero = false;

} else {

display.setText(display.getText() + String.valueOf("3"));

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(Botao03);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

//criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotao04() {

JButton Botao04 = new JButton("4");

Botao04.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (display.getText().equals("0")) {

display.setText(String.valueOf("4"));

} else if (apagarParaOProximoNumero == true) {

display.setText(String.valueOf("4"));

apagarParaOProximoNumero = false;

} else {

display.setText(display.getText() + String.valueOf("4"));

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(Botao04);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotao05() {

JButton Botao05 = new JButton("5");

Botao05.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (display.getText().equals("0")) {

display.setText(String.valueOf("5"));

} else if (apagarParaOProximoNumero == true) {

display.setText(String.valueOf("5"));

apagarParaOProximoNumero = false;

} else {

display.setText(display.getText() + String.valueOf("5"));

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(Botao05);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotao06() {

JButton Botao06 = new JButton("6");

Botao06.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (display.getText().equals("0")) {

display.setText(String.valueOf("6"));

} else if (apagarParaOProximoNumero == true) {

display.setText(String.valueOf("6"));

apagarParaOProximoNumero = false;

} else {

display.setText(display.getText() + String.valueOf("6"));

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(Botao06);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotao07() {

JButton Botao07 = new JButton("7");

Botao07.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (display.getText().equals("0")) {

display.setText(String.valueOf("7"));

} else if (apagarParaOProximoNumero == true) {

display.setText(String.valueOf("7"));

apagarParaOProximoNumero = false;

} else {

display.setText(display.getText() + String.valueOf("7"));

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(Botao07);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotao08() {

JButton Botao08 = new JButton("8");

Botao08.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (display.getText().equals("0")) {

display.setText(String.valueOf("8"));

} else if (apagarParaOProximoNumero == true) {

display.setText(String.valueOf("8"));

apagarParaOProximoNumero = false;

} else {

display.setText(display.getText() + String.valueOf("8"));

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(Botao08);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotao09() {

JButton Botao09 = new JButton("9");

Botao09.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (display.getText().equals("0")) {

display.setText(String.valueOf("9"));

} else if (apagarParaOProximoNumero == true) {

display.setText(String.valueOf("9"));

apagarParaOProximoNumero = false;

} else {

display.setText(display.getText() + String.valueOf("9"));

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(Botao09);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotaoPonto() {

JButton BotaoPonto = new JButton(".");

BotaoPonto.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (ponto == false) {

if (display.getText().equals("0")) {

display.setText(String.valueOf("0."));

ponto = true;

} else if (apagarParaOProximoNumero == true) {

display.setText(String.valueOf("0."));

apagarParaOProximoNumero = false;

} else {

display.setText(display.getText() + String.valueOf("."));

ponto = true;

}

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(BotaoPonto);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotaoIgual() {

JButton BotaoIgual = new JButton("=");

BotaoIgual.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

switch (operacao) {

case "/":

if (memoria != 0) {

display.setText(String.valueOf(memoria / Double.parseDouble(display.getText())));

apagarParaOProximoNumero = true;

fazendoOperacao = true;

operacao = "/";

ponto = false;

}

;

break;

case "\*":

if (memoria != 0) {

display.setText(String.valueOf(memoria \* Double.parseDouble(display.getText())));

apagarParaOProximoNumero = true;

fazendoOperacao = true;

operacao = "\*";

ponto = false;

}

;

break;

case "-":

if (memoria != 0) {

display.setText(String.valueOf(memoria - Double.parseDouble(display.getText())));

apagarParaOProximoNumero = true;

fazendoOperacao = true;

operacao = "-";

ponto = false;

}

;

break;

case "+":

if (memoria != 0) {

display.setText(String.valueOf(memoria + Double.parseDouble(display.getText())));

apagarParaOProximoNumero = true;

fazendoOperacao = true;

operacao = "+";

ponto = false;

}

;

break;

case "":

break;

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(BotaoIgual);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotaoDividir() {

JButton BotaoDividir = new JButton("/");

BotaoDividir.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (fazendoOperacao == false) {

memoria = Double.parseDouble(display.getText());

apagarParaOProximoNumero = true;

fazendoOperacao = true;

operacao = "/";

ponto = false;

} else {

display.setText(String.valueOf(memoria / Double.parseDouble(display.getText())));

apagarParaOProximoNumero = true;

fazendoOperacao = true;

operacao = "/";

ponto = false;

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(BotaoDividir);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotaoMultiplicar() {

JButton BotaoMultiplicar = new JButton("\*");

BotaoMultiplicar.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (fazendoOperacao == false) {

memoria = Double.parseDouble(display.getText());

apagarParaOProximoNumero = true;

fazendoOperacao = true;

operacao = "\*";

ponto = false;

} else {

display.setText(String.valueOf(memoria \* Double.parseDouble(display.getText())));

apagarParaOProximoNumero = true;

fazendoOperacao = true;

operacao = "\*";

ponto = false;

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(BotaoMultiplicar);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotaoSubtrair() {

JButton BotaoSubtrair = new JButton("-");

BotaoSubtrair.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (fazendoOperacao == false) {

memoria = Double.parseDouble(display.getText());

apagarParaOProximoNumero = true;

fazendoOperacao = true;

operacao = "-";

ponto = false;

} else {

display.setText(String.valueOf(memoria - Double.parseDouble(display.getText())));

apagarParaOProximoNumero = true;

fazendoOperacao = true;

operacao = "-";

ponto = false;

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(BotaoSubtrair);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

private void preparaBotaoSomar() {

JButton BotaoSomar = new JButton("+");

BotaoSomar.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (fazendoOperacao == false) {

memoria = Double.parseDouble(display.getText());

apagarParaOProximoNumero = true;

fazendoOperacao = true;

operacao = "+";

ponto = false;

} else {

display.setText(String.valueOf(memoria + Double.parseDouble(display.getText())));

apagarParaOProximoNumero = true;

fazendoOperacao = true;

operacao = "+";

ponto = false;

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(BotaoSomar);//adiciona botão ao painel

} //criar botão, usar a interface ActionListener para disparar o clique e acionar o método actionPerformed

/\*private void preparaBotaoSair() {

JButton botaoSair = new JButton("Sair");

botaoSair.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

painelPrincipal.add(botaoSair);//adiciona botão Sair ao painel

}\*/

}

**2) Escreva um programa capaz de converter o valor em Reais fornecido numa caixa de entrada (TextField) em outro valor monetário conforme a seleção de uma moeda (no mínimo 3 moedas diferentes) em componentes de seleção (Radio Button) indicando o resultado em um rótulo (Label). Se necessário adicione botões.**

CLASSE MAIN

package conversordereal;

//importando todas as funções das bibliotecas swing e awt

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.text.DecimalFormat;

public class ConversorDeReal {

private JFrame janela; //cria componente JFrame

private JPanel painelPrincipal; //cria componente JPainel

private Label label1, lblResultado; // cria componentes label

private TextField display; // cria componentes TextField

DecimalFormat df = new DecimalFormat("0.00");

ButtonGroup grupoDeBotoes = new ButtonGroup();

public static void main(String[] args) {

new ConversorDeReal().montaTela(); //define método montatela em Classe fatec, responsável pela interface gráfica

}

private void montaTela() {//chama os métodos que formarão a tela

preparaJanela();

preparaPainelPrincipal();

preparaLabel();

preparaText();

preparaBotaoRadioDolar();

preparaBotaoRadioEuro();

preparaBotaoRadioIene();

preparaBotaoSair();

mostraJanela();

}

private void preparaJanela() {

janela = new JFrame("Conversor de Moeda para Real");

// como é necessário utilizar o Jframe no método mostraJanela é necessário que

// janela seja um atributo ao invés de uma variável local

janela.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); //indica ao nosso frame que a aplicação deve ser terminada quando o usuário fechar a janela

}

private void mostraJanela() {

janela.pack(); // serve para organizar os componentes do frame para que eles ocupem o menor espaço possível.

janela.setSize(350, 300); //dimensões do Frame

janela.setVisible(true);// recebe um boolean indicando se queremos que

//a janela esteja visível ou não.

}

// prepara o Painel que receberá os botões e adiciona o Painel no Frame

private void preparaPainelPrincipal() {

painelPrincipal = new JPanel();

painelPrincipal.setLayout(new BoxLayout(painelPrincipal, BoxLayout.PAGE\_AXIS));

// layout em forma de tabela com 3 linhas e 2 colunas;

//painelPrincipal.setLayout(new GridLayout(5, 4));

janela.add(painelPrincipal);

}

// insere texto no label1 e adiciona o label no painel;

private void preparaLabel() {

label1 = new Label("Valor em Reais");

painelPrincipal.add(label1);

}

;

// insere texto no label , espaços aos TextFields e adiciona os componentes no painel;

private void preparaText() {

display = new TextField("");

lblResultado = new Label("");

painelPrincipal.add(display);

painelPrincipal.add(lblResultado);

}

;

private void preparaBotaoRadioDolar() {

final JRadioButton jRadioButtonDolar = new JRadioButton("Dolar");

jRadioButtonDolar.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

lblResultado.setText(String.valueOf("O valor em Dólar é US$ " + df.format(Double.parseDouble(display.getText()) / 5.44)));

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

grupoDeBotoes.add(jRadioButtonDolar);//adiciona botão ao painel

painelPrincipal.add(jRadioButtonDolar);//adiciona botão ao painel

}

private void preparaBotaoRadioEuro() {

final JRadioButton jRadioButtonEuro = new JRadioButton("Euro");

jRadioButtonEuro.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

lblResultado.setText(String.valueOf("O valor em Euro é € " + df.format(Double.parseDouble(display.getText()) / 6.55)));

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

grupoDeBotoes.add(jRadioButtonEuro);//adiciona botão ao painel

painelPrincipal.add(jRadioButtonEuro);//adiciona botão ao painel

}

private void preparaBotaoRadioIene() {

final JRadioButton jRadioButtonIene = new JRadioButton("Iene");

jRadioButtonIene.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

lblResultado.setText(String.valueOf("O valor em Iene é ¥ " + df.format(Double.parseDouble(display.getText()) / 0.05)));

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

grupoDeBotoes.add(jRadioButtonIene);//adiciona botão ao painel

painelPrincipal.add(jRadioButtonIene);//adiciona botão ao painel

}

private void preparaBotaoSair() {

JButton botaoSair = new JButton("Sair");

botaoSair.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(botaoSair);//adiciona botão Sair ao painel

}

}

**3) Escreva um programa que peça como entrada o peso (em kilogramas) e a altura (em metros) da pessoa e calcule e exiba o seu IMC – Índice de Massa Corpórea. Adicionalmente, o programa deve emitir as mensagens correspondentes conforme a tabela a seguir:**

**Cálculo IMC Situação**

**Abaixo de 18.5 Você está abaixo do peso ideal**

**Entre 18.5 e 24.9 Parabéns — você está em seu peso normal!**

**Entre 25.0 e 29.9 Você está acima de seu peso (sobrepeso)**

**Entre 30.0 e 34.9 Obesidade grau I**

**Entre 35.0 e 39.9 Obesidade grau II**

**40.0 e acima Obesidade grau III**

**O IMC é calculado da seguinte forma: divide-se o peso (quilogramas) pela altura ao quadrado (em metros).**

**Exemplo1: peso = 65, altura = 1.67 => IMC = 65 / (1.67 \* 1.67) = 23.30**

**Exemplo2: peso = 85, altura = 1.60 => IMC = 85 / (1.60 \* 1.60) = 33.2**

CLASSE MAIN

package calculadoraimc;

//importando todas as funções das bibliotecas swing e awt

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.text.DecimalFormat;

public class CalculadoraIMC {

private JFrame janela; //cria componente JFrame

private JPanel painelPrincipal; //cria componente JPainel

private Label label1, lblPesoKg, lblAlturaMt, lblIMC, lblSituacao; // cria componentes label

private TextField txtPesoKg, txtAlturaMt; // cria componentes TextField

DecimalFormat df = new DecimalFormat("0.00");

double situacao;

public static void main(String[] args) {

new CalculadoraIMC().montaTela(); //define método montatela em Classe fatec, responsável pela interface gráfica

}

private void montaTela() {//chama os métodos que formarão a tela

preparaJanela();

preparaPainelPrincipal();

preparaLabel();

preparaText();

preparaBotaoCalcular();

preparaBotaoSair();

mostraJanela();

}

private void preparaJanela() {

janela = new JFrame("Conversor de Moeda para Real");

// como é necessário utilizar o Jframe no método mostraJanela é necessário que

// janela seja um atributo ao invés de uma variável local

janela.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); //indica ao nosso frame que a aplicação deve ser terminada quando o usuário fechar a janela

}

private void mostraJanela() {

janela.pack(); // serve para organizar os componentes do frame para que eles ocupem o menor espaço possível.

janela.setSize(350, 300); //dimensões do Frame

janela.setVisible(true);// recebe um boolean indicando se queremos que

//a janela esteja visível ou não.

}

// prepara o Painel que receberá os botões e adiciona o Painel no Frame

private void preparaPainelPrincipal() {

painelPrincipal = new JPanel();

painelPrincipal.setLayout(new BoxLayout(painelPrincipal, BoxLayout.PAGE\_AXIS));

// layout em forma de tabela com 3 linhas e 2 colunas;

//painelPrincipal.setLayout(new GridLayout(5, 4));

janela.add(painelPrincipal);

}

// insere texto no label1 e adiciona o label no painel;

private void preparaLabel() {

label1 = new Label("Calculadora de Índice de Massa Corpórea");

painelPrincipal.add(label1);

}

;

// insere texto no label , espaços aos TextFields e adiciona os componentes no painel;

private void preparaText() {

lblPesoKg = new Label("Peso em KG");

txtPesoKg = new TextField("");

lblAlturaMt = new Label("Altura em Metros");

txtAlturaMt = new TextField("");

lblIMC = new Label("");

lblSituacao = new Label("");

painelPrincipal.add(lblPesoKg);

painelPrincipal.add(txtPesoKg);

painelPrincipal.add(lblAlturaMt);

painelPrincipal.add(txtAlturaMt);

painelPrincipal.add(lblIMC);

painelPrincipal.add(lblSituacao);

}

;

private void preparaBotaoCalcular() {

JButton botaoCalcular = new JButton("Calcular");

botaoCalcular.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

situacao = ( (Double.parseDouble(txtPesoKg.getText())) / ( (Double.parseDouble(txtAlturaMt.getText())) \* (Double.parseDouble(txtAlturaMt.getText())) ) );

lblIMC.setText(String.valueOf(df.format(situacao)));

if(situacao < 18.5){

lblSituacao.setText("Você está abaixo do peso ideal");

} else if(situacao >= 18.5 && situacao <= 24.9){

lblSituacao.setText("Parabéns - você está em seu peso normal!");

} else if(situacao >= 25.0 && situacao <= 29.9){

lblSituacao.setText("Você está acima de seu peso (sobrepeso)");

} else if(situacao >= 30.0 && situacao <= 34.9){

lblSituacao.setText("Obesidade grau I");

} else if(situacao >= 35.0 && situacao <= 39.9){

lblSituacao.setText("Obesidade grau II");

} else if(situacao >= 40.0){

lblSituacao.setText("Obesidade grau III");

}

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(botaoCalcular);//adiciona botão Sair ao painel

}

private void preparaBotaoSair() {

JButton botaoSair = new JButton("Sair");

botaoSair.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

painelPrincipal.setLocation(50, 100);//posição do painel no frame

painelPrincipal.add(botaoSair);//adiciona botão Sair ao painel

}

}