package estacionamentojg;

import java.sql.\*;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import java.sql.Timestamp;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

public class BD {

private Connection pegaConexao() { //metodo para criar uma conexao com o banco

try {

String url = "jdbc:postgresql://localhost:5432/estacionamentojg"; //define o caminho

String usuario = "postgres"; //define o usuario

String senha = "12345"; //define a senha

Connection con = DriverManager.getConnection(url, usuario, senha); //cria conexao com banco atraves de um driver passando a url,usuario,senha

return con;

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

private Statement conecta() { //metodo para conectar no banco

try {

Class.forName("org.postgresql.Driver");

Statement s = pegaConexao().createStatement();

return s;

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

public ArrayList getModelos() { //pega modelos de carros do banco

try {

ResultSet rs = conecta().executeQuery("SELECT \* FROM carros ORDER BY modelo ASC;"); //consulta para pegar modelos em ordem alfabética

ArrayList<String> dados = new ArrayList<>();//cria um objeto dados da classe arraylist

while (rs.next()) { //consulta para navegar no banco enquanto tiver dados para serem adicionados

dados.add(rs.getString("modelo"));//add o modelo

}

pegaConexao().close();//termina conexao

return dados;

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

public Boolean Estacionado(String modelo, String placa, String cor, Timestamp entrada, String dia, Double pN, Double pH) { // insere no banco os carros que entraram no estacionamento

try {

ResultSet rs = conecta().executeQuery("SELECT COUNT(\*) FROM estacionados WHERE placa='" + placa + "'"); // conta vezes que o veiculo foi cadastrado e não saiu do estacionamento

int cont = 0;

while (rs.next()) {

cont = rs.getInt(1); // pega valor de vezes que a placa esta presente na tabela estacionados

}

if (cont == 0) { // se a placa ainda não foi cadastrada

conecta().execute("INSERT INTO estacionados (placa, modelo, cor, datahora\_inicial, dia, primeira\_hora, preco\_hora) "

+ "VALUES ('" + placa + "', '" + modelo + "', '" + cor + "', '" + entrada + "', '" + dia + "', '" + pN + "', '" + pH + "');"); // insere carro estacionado

pegaConexao().close();

JOptionPane.showMessageDialog(null, "Carro cadastrado com sucesso!"); // mensagem ao usuário

return true;

} else {

JOptionPane.showMessageDialog(null, "ERRO! Carro já cadastrado.", "Erro", JOptionPane.ERROR\_MESSAGE); // mensagem ao usuário

return false;

}

} catch (SQLException e) {

e.printStackTrace();

JOptionPane.showMessageDialog(null, "ERRO! Tente novamente.", "Erro", JOptionPane.ERROR\_MESSAGE); // mensagem ao usuário

return false;

}

}

public Boolean Encerrado(String placa, Timestamp saida) { // insere no banco os carros que sairam do estacionamento

try {

ResultSet rs = conecta().executeQuery("SELECT \* FROM estacionados WHERE placa='" + placa + "' "); // busca a placa que ira sair dentro da tabela estacionados

Timestamp entrada = null;

String diaEntrada = null;

Double pP = null;

Double pH = null;

int cont = 0;

while (rs.next()) {

entrada = rs.getTimestamp("datahora\_inicial"); // pega Timestamp de entrada da placa

pP = rs.getDouble("primeira\_hora");//pega valor da primeira hora definido

pH = rs.getDouble("preco\_hora");//pega valor do preço da hora definido

cont++; // pega valor de vezes que a placa esta registrada

}

if (cont != 0) { // se placa está cadastrada

Valores valor = new Valores(pP, pH); //cria um objeto valor da classe valores recebendo os preço da primeira hora e o preco da hora definidos

Double preco = valor.total(entrada, saida);//recebe o preco total

new PrecoTotal(preco); // janela de preço

conecta().execute("INSERT INTO encerrados (placa, datahora\_inicial, datahora\_final, valor) VALUES ('" + placa + "', '" + entrada + "', '" + saida + "', '" + preco + "');"); // insere carro encerrado

conecta().execute("DELETE FROM estacionados WHERE placa = '" + placa + "'"); // deleta carro da tabela estacionado

return true;

} else {

JOptionPane.showMessageDialog(null, "ERRO! Placa não cadastrada.", "Erro", JOptionPane.ERROR\_MESSAGE); // mensagem ao usuário

return false;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "ERRO! Tente novamente.", "Erro", JOptionPane.ERROR\_MESSAGE); // mensagem ao usuário

return false;

}

}

public Boolean buscaPlaca(String placa) { //metodo para buscar a placa dentro da tabela estacionados

try {

ResultSet rs = conecta().executeQuery("SELECT \* FROM estacionados WHERE placa='" + placa + "'");//consulta para buscar a placa desejada dentro da tabela estacionados

String achou = null;

while (rs.next()) { //laço para buscar a placa desejada

achou = rs.getString("placa");

}

pegaConexao().close();

if (achou != null) { //condicao para verificar se achou ou nao a placa

return true;

} else {

return false;

}

} catch (Exception e) {

return false;

}

}

public String relDiario(String dataAtual) { //metodo para gerar o relatorio diario

String total = "";

Double valorTotal = 0.0;

int cont = 0;

Calendar c = Calendar.getInstance(); // cria um objeto da classe calendar

String dataBanco;

try {

ResultSet rs = conecta().executeQuery("SELECT \* FROM encerrados;"); //busca todos valores da tabela encerrados

while (rs.next()) {//laço para passar os valores para o relatorio diario

c.setTime(rs.getTimestamp("datahora\_final"));//para o valor da data e hora da saida

dataBanco = c.get(Calendar.DAY\_OF\_MONTH) + "." + (c.get(Calendar.MONTH) + 1) + "." + c.get(Calendar.YEAR);//pega o dia.mes.ano

if (dataBanco.equals(dataAtual)) {//condicao para verificar se a data requisitada é igual a data do banco

total += rs.getString("placa");//pega placa

total += "\t\t\t\t\t\t\t\t\t\t";

total += rs.getTimestamp("datahora\_inicial").toString();//pega datahora inicial

total += "\t\t\t\t\t\t\t\t\t\t";

total += rs.getTimestamp("datahora\_final").toString();//pega data e hora final

total += "\t\t\t\t\t\t\t\t\t\t";

total += "R$ " + String.format("%.2f", rs.getDouble("valor")); //pega valor gasto pelo carro

total += "\n";

valorTotal += rs.getDouble("valor"); // pega valor do carro e adiciona no total

cont++;

}

}

total += "\n\n";

total += "total de carros: " + cont; //informa o numero de carros

total += " ";

total += "valor total: R$ " + String.format("%.2f", valorTotal);;//informa o valor total recebido no dia

if (cont == 0) {//condicao para data nao encontrada

JOptionPane.showMessageDialog(null, "Data não encontrada", "Erro", JOptionPane.ERROR\_MESSAGE);

return null;

}

return total;

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

public String relMensal(String dataAtual) {//metodo para gerar o relatorio mensal

String total = "";

Double valorTotal = 0.0;

Calendar c = Calendar.getInstance();//cria um objeto da classe calendar

String dataBanco;

int cont = 0;

try {

ResultSet rs = conecta().executeQuery("SELECT \* FROM encerrados;");//consulta para buscar todos valores da tabela encerrados

while (rs.next()) {

c.setTime(rs.getTimestamp("datahora\_final"));//pega data e hora final

dataBanco = (c.get(Calendar.MONTH) + 1) + "." + c.get(Calendar.YEAR);//pega o mes.ano

if (dataBanco.equals(dataAtual)) { //se a data requisitada é igual a data do banco

total += rs.getString("placa");//pega placa

total += "\t\t\t\t\t\t\t\t\t\t";

total += rs.getTimestamp("datahora\_inicial").toString();//pega data e hora que entrou o carro

total += "\t\t\t\t\t\t\t\t\t\t";

total += rs.getTimestamp("datahora\_final").toString();//pega data e hora que saiu o carro

total += "\t\t\t\t\t\t\t\t\t\t";

total += "R$ " + String.format("%.2f", rs.getDouble("valor"));//pega o valor gasto pelo carro

total += "\n";

valorTotal += rs.getDouble("valor"); //acumula o valor gasto no total gasto

cont++;

}

}

total += "\n\n";

total += "total de carros: " + cont;//informa o numero de carros

total += " ";

total += "valor total: R$ " + String.format("%.2f", valorTotal);;//informa o valor total

if (cont == 0) {

JOptionPane.showMessageDialog(null, "Data não encontrada", "Erro", JOptionPane.ERROR\_MESSAGE);

return null;

}

return total;

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

}

package estacionamentojg;

import java.awt.Desktop;

import java.io.File;

import java.io.IOException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

public class Estacionamento extends javax.swing.JFrame {

private Double primeiraHora;

private Double precoHora;

Estacionamento(Double pP, Double pH) { //construtor

initComponents();

setLocationRelativeTo(null); // coloca janela no centro da tela

setVisible(true);

this.primeiraHora = pP;//variavel primeira hora recebe Preco primeira hora da classe

this.precoHora = pH;

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

buttonGroup1 = new javax.swing.ButtonGroup();

buttonGroup2 = new javax.swing.ButtonGroup();

jOptionPane1 = new javax.swing.JOptionPane();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jMenuBar1 = new javax.swing.JMenuBar();

jMenu1 = new javax.swing.JMenu();

jMenuItem2 = new javax.swing.JMenuItem();

jMenuItem3 = new javax.swing.JMenuItem();

jMenuItem4 = new javax.swing.JMenuItem();

jMenuItem5 = new javax.swing.JMenuItem();

jMenu2 = new javax.swing.JMenu();

jMenuItem1 = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Estacionamento JG");

jButton1.setBackground(new java.awt.Color(153, 255, 153));

jButton1.setFont(new java.awt.Font("Calibri", 1, 24)); // NOI18N

jButton1.setText(" Entrada de Veículo");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setBackground(new java.awt.Color(255, 102, 102));

jButton2.setFont(new java.awt.Font("Calibri", 0, 24)); // NOI18N

jButton2.setText("Saída de Veículo");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jMenu1.setText("Administrador");

jMenuItem2.setText("Gerar relatório diário");

jMenuItem2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenuItem2ActionPerformed(evt);

}

});

jMenu1.add(jMenuItem2);

jMenuItem3.setText("Gerar relatório mensal");

jMenuItem3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenuItem3ActionPerformed(evt);

}

});

jMenu1.add(jMenuItem3);

jMenuItem4.setText("Abrir relatórios diários");

jMenuItem4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenuItem4ActionPerformed(evt);

}

});

jMenu1.add(jMenuItem4);

jMenuItem5.setText("Abrir relatórios mensais");

jMenuItem5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenuItem5ActionPerformed(evt);

}

});

jMenu1.add(jMenuItem5);

jMenuBar1.add(jMenu1);

jMenu2.setText("Tabela de Preços");

jMenuItem1.setText("Alterar tabela de preços");

jMenuItem1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenuItem1ActionPerformed(evt);

}

});

jMenu2.add(jMenuItem1);

jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 244, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 244, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(205, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(31, 31, 31)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 140, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 119, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

new RegistraEntrada(this.primeiraHora, this.precoHora);

// ação do botão registrar veiculo instancia novo objeto da classe RegistraEntrada

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

new RegistraSaida();

// ação do botão registrar veiculo instancia novo objeto da classe RegistraSaida

}

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {

new ManipulaValores();

this.dispose(); // fecha janela

}

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {

new RelatorioDia();

}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {

new RelatorioMes();

}

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {

File diretorioDiario = new File("Relatorios/Relatorios Diarios");

if (!diretorioDiario.exists()) // se diretorioDiario ainda não foi criado

{

JOptionPane.showMessageDialog(null, "Erro. Nenhum relatório diário gerado.", "Erro", JOptionPane.ERROR\_MESSAGE);

}

try {

Desktop desktop = Desktop.getDesktop();

desktop.open(new File("Relatorios/Relatorios Diarios")); // abrir diretorio automaticamente

} catch(Exception e) {

System.out.println("Erro ao abrir diretório. Erro:" + e);

}

}

private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {

File diretorioMensal = new File ("Relatorios/Relatorios Mensais");

if (!diretorioMensal.exists()) { // se diretorioMensal ainda não foi criado

JOptionPane.showMessageDialog(null, "Erro. Nenhum relatório mensal gerado.", "Erro", JOptionPane.ERROR\_MESSAGE);

}

try {

Desktop desktop = Desktop.getDesktop();

desktop.open(new File("Relatorios/Relatorios Mensais")); // abrir diretorio automaticamente

this.dispose();

new Estacionamento(this.primeiraHora, this.precoHora);

} catch(Exception e) {

System.out.println("Erro ao abrir diretório. Erro:" + e);

}

}

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new ManipulaValores();

//new Estacionamento().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.ButtonGroup buttonGroup1;

private javax.swing.ButtonGroup buttonGroup2;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JMenu jMenu1;

private javax.swing.JMenu jMenu2;

private javax.swing.JMenuBar jMenuBar1;

private javax.swing.JMenuItem jMenuItem1;

private javax.swing.JMenuItem jMenuItem2;

private javax.swing.JMenuItem jMenuItem3;

private javax.swing.JMenuItem jMenuItem4;

private javax.swing.JMenuItem jMenuItem5;

private javax.swing.JOptionPane jOptionPane1;

// End of variables declaration

}

package estacionamentojg;

import com.itextpdf.text.Document;

import com.itextpdf.text.PageSize;

import com.itextpdf.text.Paragraph;

import com.itextpdf.text.pdf.PdfWriter;

import java.awt.Desktop;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.OutputStream;

import java.sql.Timestamp;

import java.text.SimpleDateFormat;

import java.util.Date;

public class GeraRelatorio {

private String r = "Relatorios";

private String d = "Relatorios/Relatorios Diarios";

private String m = "Relatorios/Relatorios Mensais";

public void diario(String data) throws Exception { //metodo que gera o relatorio diario da data solicitada

data = getData(data);//pega a data solicitada

criaDiretorio(r);// cria diretorio relatorios

criaDiretorio(d);// cria diretorio relatorios/relatorios diarios

BD bd = new BD();//cria objeto bd da classe BD

String dados = bd.relDiario(data); //passa os dados registrados no banco que tem a data desejada

if (dados != null) {

String nomePDF = geraPDF(d, dados);//gera o pdf na pasta rel/rel diarios com os dados da data desejada

Desktop desktop = Desktop.getDesktop();//cria um objeto desktop da classe desktop para ser usado para abrir o pdf aut

abrePDF(nomePDF);//abre pdf automaticamente

}

}

public void mensal(String data) throws Exception { //metodo que gera o relatorio mensal da data solicitada

data = getData(data); //pega a data solicitada

criaDiretorio(r);//cria o diretorio relatorio

criaDiretorio(m);//cria o diretorio relatorio/relatorios mensais

BD bd = new BD();//cria um objeto bd da classe BD

String dados = bd.relMensal(data);//passa os dados registrados no banco que tem a data desejada

if (dados != null) {//se a data for encontrada

String nomePDF = geraPDF(m, dados); // gera o pdf na pasta rel/rel mensais com os dados da data desejada

abrePDF(nomePDF);//abre pdf automaticamente

}

}

private void abrePDF(String nomePDF) throws IOException {//metodo para abrir o pdf automaticamente no desktop

Desktop desktop = Desktop.getDesktop(); // cria objeto desktop da classe Desktop

desktop.open(new File(nomePDF)); // abrir pdf automaticamente

}

private String geraPDF(String pasta, String dados) throws Exception {//metodo para gerar o pdf

String nomePDF = null;

Document doc = null;

OutputStream os = null;

String titulo = null;

String abas = "\n\nPlaca\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\tData/hora de entrada\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t"

+ "Data/hora de saída\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\tValor\n";

try {

doc = new Document(PageSize.A4, 72, 72, 72, 72); //cria o documento tamanho A4, margens de 2,54cm

if (pasta.equals(d)) {//se a pasta for de rel/rel diarios

nomePDF = pasta + "/RD" + data() + ".pdf";//arruma o nome do pdf para ficar dentro da pasta rd com seu nome certo

os = new FileOutputStream(nomePDF); //cria a stream de saída

titulo = "Relatorio Diario - Estacionamento JG";//coloca o titulo do pdf dos relatorios diarios

} else if (pasta.equals(m)) {//se a pasta for de rel/rel mensais

nomePDF = pasta + "/RM" + data() + ".pdf";//arruma o nome do pdf para ficar dentro da pasta rm com seu nome certo

os = new FileOutputStream(nomePDF); //cria a stream de saída

titulo = "Relatorio Mensal - Estacionamento JG";//coloca o titulo do pdf nos relatorios mensais

}

PdfWriter.getInstance(doc, os); //associa a stream de saída ao pdfwriter

doc.open(); //abre o documento

Paragraph p = new Paragraph(titulo);//cria um paragrafo dentro do pdf com o titulo

doc.add(p); //adiciona o titulo no pdf

p = new Paragraph(abas);//cria um paragrafo dentro do pdf com as abas

doc.add(p);//adiciona as abas no pdf

p = new Paragraph(dados);//cria um paragrafo dentro do pdf com os dados

doc.add(p);//adiciona os dados no pdf

return nomePDF;

} finally {

if (doc != null) {

doc.close(); //fechamento do documento

}

if (os != null) {

os.close(); //fechamento da stream de saída

}

}

}

private void criaDiretorio(String nome) {//metodo para criar um diretorio

File diretorio = new File(nome);//objeto diretorio da classe File com o nome passado pelo metodo

if (!diretorio.exists()) // se diretorio ainda não foi criado

{

diretorio.mkdir(); // cria diretorio

}

}

private String data() {// metodo que transforma timestamp em string para titulo do pdf

Date data = new Date(); //cria objeto data da classe date

Timestamp timestamp = new Timestamp(data.getTime());// cria objeto timestamp da clase Timestamp passando da data desejada

String aux = timestamp.toString();//transforma a data para string

String sData = "";

for (int i = 0; i < 20; i++) {//laco que percorre timestamp e para antes dos centésimos

if (i == 13) {

sData += 'h';

} else if (i == 16) {

sData += 'm';

} else if (i == 19) {

sData += 's';

} else {

sData += aux.charAt(i); // concatena todas as posições numa nova string

}

}

return sData; // retorna timestamp editada como string

}

private String getData(String data) { //metodo para pegar a data e alterar sua estrutura

String d = "";

for (int i = 0; i < data.length(); i++) {//laco para alterar a barra por ponto e retirar o 0 do dia e do mes

if (data.charAt(i) == '/') {

d += '.';

} else {

if (i == 0 || i == 3) {

if (data.charAt(i) != '0') {

d += data.charAt(i);

}

} else {

d += data.charAt(i);

}

}

}

return d;

}

}

package estacionamentojg;

import java.sql.Timestamp;

import java.text.ParseException;

import java.util.Calendar;

import java.util.Date;

import javax.swing.JFormattedTextField;

import javax.swing.JOptionPane;

import javax.swing.text.MaskFormatter;

public class ManipulaValores extends javax.swing.JFrame {

MaskFormatter real = new MaskFormatter(); //cria objeto para a classe maskFormatter

public ManipulaValores() {//construtor

initComponents();

setLocationRelativeTo(null); // coloca janela no centro da pagina

setVisible(true);

Date data = new Date(); // cria objeto data da classe Date

Timestamp timestamp = new Timestamp(data.getTime()); //cria objeto timestamp da clsse TimeStamp

Calendar c = Calendar.getInstance();//cria objeto da claasse calendar

c.setTime(timestamp);//seta o objeto timestamp

if (c.get(Calendar.DAY\_OF\_WEEK) == 1) // pega o domingo

{

jButton2.setEnabled(false); //seta como false o botao ok porque nao eh possivel definir valores para o domingo

}

try {

real.setMask("R$ #.##"); // mascara para preço

real.setPlaceholderCharacter('0'); // caracter que fica ocupando o espaço

} catch (ParseException e) {

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel4 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jTextField1 = new JFormattedTextField(real);

jTextField2 = new JFormattedTextField(real);

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jLabel3 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel4.setText("jLabel4");

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Estacionamento JG - Tabela de Valores");

jLabel1.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N

jLabel1.setText("Primeira hora:");

jLabel2.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N

jLabel2.setText("Hora extra:");

jTextField1.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N

jTextField2.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N

jButton1.setFont(new java.awt.Font("Calibri", 1, 14)); // NOI18N

jButton1.setText("Padrão");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

jButton2.setText("Ok");

jButton2.setToolTipText("");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jLabel3.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

jLabel3.setText("Domingo: R$ 3,00 diária");

jLabel5.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

jLabel5.setText("Padrão: Primeira hora: R$2,00 | Hora extra: R$ 1,00");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(90, 90, 90)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 66, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton1)

.addGap(75, 75, 75))

.addGroup(layout.createSequentialGroup()

.addGap(53, 53, 53)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 111, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 103, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(45, 45, 45)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jTextField2, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 107, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField1, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 107, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(84, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 145, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(121, 121, 121))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(52, Short.MAX\_VALUE)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 303, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(45, 45, 45)))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(49, 49, 49)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 38, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 38, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 37, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 41, Short.MAX\_VALUE)

.addComponent(jLabel3)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton1)

.addComponent(jButton2))

.addGap(56, 56, 56))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(159, Short.MAX\_VALUE)

.addComponent(jLabel5)

.addGap(124, 124, 124)))

);

pack();

}// </editor-fold>

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

//acao do botao Ok

Boolean a = true;

//condicao para nao deixa setar o dia como de "graça" e nem deixar os campos nulos

if (jTextField2.getText().equals("") || jTextField2.getText().equals("R$ 0.00") || jTextField1.getText().equals("R$ 0.00")){

a = false;

}

if (a == true) {// trata os dois valores informados pelo usuario

new Estacionamento(trata(this.jTextField1.getText()), trata(this.jTextField2.getText()));

this.dispose();

}

else

JOptionPane.showMessageDialog(null, "Erro, campo vazio/zerado.", "Erro", JOptionPane.ERROR\_MESSAGE);

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

//acao do botao padrao

new Estacionamento(2.0, 1.0);//define valores 2.0 para primeirahora e 1.0 para precohora

this.dispose(); //fechar a tela

}

public Double trata(String texto) { //metodo para tratar string para double

String stringFinal = ""; //inicializa stringfinal

for (int i = 0; i < texto.length(); i++) { //laco para concatenar cada caracter em uma string

if (i > 2) {//pega a partir do terceiro elemento transformando R$ 3,00 em 3,00

stringFinal += Character.toString(texto.charAt(i));

}

}

return Double.parseDouble(stringFinal);//transforma stringfinal para double

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

// End of variables declaration

}

package estacionamentojg;

import javax.swing.JLabel;

public class PrecoTotal extends javax.swing.JFrame {

public PrecoTotal(Double preco) { //construtor

this.setVisible(true);

this.textoJLabel(preco);//inicializa a label preco total

initComponents();

this.setSize(350, 150);

setLocationRelativeTo(null);

}

private JLabel jPreco = new javax.swing.JLabel(); //cria objeto jpreco da classe jlabel

public void textoJLabel(Double preco) {

String precoTotal = "R$ " + preco + "0"; // string recebe preco total

jPreco.setFont(new java.awt.Font("Calibri", 1, 30));//define fonte

jPreco.setText(precoTotal);//seta preco total

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jLabel2 = jPreco;

jButton1 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Estacionamento JG - Preço Total");

jLabel1.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N

jLabel1.setText("Preço Total: ");

jLabel2.setFont(new java.awt.Font("Calibri", 1, 30)); // NOI18N

jButton1.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

jButton1.setText("Ok");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(48, 48, 48)

.addComponent(jLabel1)

.addGap(27, 27, 27)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 164, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(140, 140, 140)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 66, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(68, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 43, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addGap(11, 11, 11)

.addComponent(jLabel1)))

.addGap(18, 18, 18)

.addComponent(jButton1)

.addContainerGap(203, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose();

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

// End of variables declaration

}

package estacionamentojg;

import java.text.ParseException;

import java.util.ArrayList;

import java.util.Date;

import javax.swing.JComboBox;

import javax.swing.text.MaskFormatter;

import javax.swing.JFormattedTextField;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

public class RegistraEntrada extends javax.swing.JFrame {

private Double primeiraHora;

private Double precoHora;

MaskFormatter mPLACA = new MaskFormatter(); //cria objeto mPLACA da classe MaskFormatter

BD c = new BD(); //cria objeto c da classe BD

Veiculo carro = new Veiculo(); //cria objeto carro da classe Veiculo

public RegistraEntrada(Double pP, Double pH) {//cria construtor

setValores(pP, pH);//chama o metodo para setar os valores das variaveis primeiraHora e preçoHora

initComponents();

setLocationRelativeTo(null); // coloca janela no centro da pagina

setVisible(true);

try {

mPLACA.setMask("UUU-####"); // mascara para placa

mPLACA.setPlaceholderCharacter('\_'); // caracter que fica ocupando o espaço

} catch (ParseException e) { //controle de excecao

e.printStackTrace();

}

}

private void setValores(Double pP, Double pH) { //cria metodo para setar as variaveis primeiraHora e preçoHora

this.primeiraHora = pP;

this.precoHora = pH;

}

@SuppressWarnings("unchecked")

private JComboBox addComboBox(ArrayList x) {//cria um metodo para adicionar dados do ArrayList no combobox

JComboBox cb = new JComboBox(); // cria objeto cb da classe JComboBox

String str; //declara str da classe String

ArrayList<String> listaModelos = x; // define o tipo da variavel que armazenara os modelos do banco e recebe a variavel passada por parametro pelo metodo

cb.addItem("Escolha um modelo...");//adiciona a string no combobox

for (String n : listaModelos) { //laço para adicionar os valores da lista de modelos do banco no combobox

str = n; // pega cada item

cb.addItem(n); // e add no JComboBox

}

cb.addItem("Outro");//adiciona uma string para definir um carro inexistente no banco

return cb; // retorna JComboBox

}

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jToggleButton1 = new javax.swing.JToggleButton();

placa = new javax.swing.JLabel();

modelo = new javax.swing.JLabel();

cor = new javax.swing.JLabel();

textoCor = new javax.swing.JTextField();

registrar = new javax.swing.JButton();

tiposModelo = addComboBox(c.getModelos());

textoPlaca = new JFormattedTextField(mPLACA);

jButton1 = new javax.swing.JButton();

jToggleButton1.setText("jToggleButton1");

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Estacionamento JG - Registro de Entrada");

placa.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N

placa.setText("Placa");

modelo.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N

modelo.setText("Modelo");

cor.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N

cor.setText("Cor");

textoCor.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N

registrar.setBackground(new java.awt.Color(0, 102, 255));

registrar.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N

registrar.setForeground(new java.awt.Color(255, 255, 255));

registrar.setText("Registrar");

registrar.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

registrarActionPerformed(evt);

}

});

tiposModelo.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N

textoPlaca.setFont(new java.awt.Font("Calibri", 1, 48)); // NOI18N

jButton1.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

jButton1.setText("Cancelar");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(64, 64, 64)

.addComponent(registrar, javax.swing.GroupLayout.PREFERRED\_SIZE, 111, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(63, 63, 63)

.addComponent(jButton1))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(47, 47, 47)

.addComponent(placa))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(29, 29, 29)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(cor, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(modelo, javax.swing.GroupLayout.Alignment.TRAILING))))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(textoPlaca, javax.swing.GroupLayout.PREFERRED\_SIZE, 216, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(textoCor, javax.swing.GroupLayout.PREFERRED\_SIZE, 180, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(tiposModelo, javax.swing.GroupLayout.PREFERRED\_SIZE, 281, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(20, 20, 20)

.addComponent(textoPlaca, javax.swing.GroupLayout.PREFERRED\_SIZE, 48, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(29, 29, 29)

.addComponent(placa, javax.swing.GroupLayout.PREFERRED\_SIZE, 29, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(26, 26, 26)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(tiposModelo, javax.swing.GroupLayout.PREFERRED\_SIZE, 32, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(modelo))

.addGap(27, 27, 27)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(textoCor, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(cor))

.addGap(37, 37, 37)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(registrar, javax.swing.GroupLayout.PREFERRED\_SIZE, 47, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton1))

.addContainerGap(38, Short.MAX\_VALUE))

);

tiposModelo.getAccessibleContext().setAccessibleParent(tiposModelo);

pack();

}// </editor-fold>

private void registrarActionPerformed(java.awt.event.ActionEvent evt) {

// AÇÃO DO BOTÃO REGISTRAR

carro.setPlaca(textoPlaca.getText());// pega valor da placa

carro.setModelo((String)tiposModelo.getSelectedItem()); // converte em string o item selecionado

carro.setCor(textoCor.getText()); // pega valor da cor

carro.setEntrada(); // inicia data e hora da entrada

Boolean a = true;

for (int i = 0; i < textoPlaca.getText().length(); i++) { //laco com condicao para nao deixar registar se o usuario for registrar a placa com \_

if (textoPlaca.getText().charAt(i) == '\_') {

a = false;

}

}

if (textoCor.getText().equals("")) { //laco com condicao para nao deixar registar se o campo cor estiver vazio

a = false;

}

if (carro.getModelo().equals("Escolha um modelo...")) {//condicao para nao deixar registar se o usuario nao escolher um modelo

a = false;

}

if (a == true) { //condicao para registar o carro

BD registra = new BD(); // cria objeto registra da classe BD

Boolean result = registra.Estacionado(carro.getModelo(), carro.getPlaca(), carro.getCor(), carro.getEntrada(), carro.diaDaSemanaInicial(), this.primeiraHora, this.precoHora); // insere carro estacionado

if (result == true) { //se carro for registrado passar os dados do carro registrado para o objeto ticket e o ticket eh gerado

Ticket ticket = new Ticket(); //cria objeto ticket da classe Ticket

ticket.setModelo(carro.getModelo());//pega modelo do carro

ticket.setPlaca(carro.getPlaca());//pega placa do carro

ticket.setCor(carro.getCor());//pega cor do carro

ticket.setDia(carro.diaDaSemanaInicial());//pega dia da semana

ticket.setDatahora(carro.getEntrada());//pega data e hora da entrada do carro

ticket.emite();//emite o ticket

this.dispose(); // fechar janela

}

}

else //controle de excecao para caso n registrar o carro

JOptionPane.showMessageDialog(null, "Erro, tente novamente.", "Erro", JOptionPane.ERROR\_MESSAGE);

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose();//fechar a janela no botao cancelar

}

// Variables declaration - do not modify

private javax.swing.JLabel cor;

private javax.swing.JButton jButton1;

private javax.swing.JToggleButton jToggleButton1;

private javax.swing.JLabel modelo;

private javax.swing.JLabel placa;

private javax.swing.JButton registrar;

private javax.swing.JTextField textoCor;

private javax.swing.JFormattedTextField textoPlaca;

private javax.swing.JComboBox tiposModelo;

// End of variables declaration

}

package estacionamentojg;

import java.text.ParseException;

import javax.swing.text.MaskFormatter;

import javax.swing.JFormattedTextField;

import javax.swing.JOptionPane;

public class RegistraSaida extends javax.swing.JFrame {

MaskFormatter mPLACA = new MaskFormatter();//cria objeto mPlaca da classe MaskFormatter

public RegistraSaida() {//cria construtor

initComponents();

setLocationRelativeTo(null); // coloca janela no centro da pagina

setVisible(true);

try {

mPLACA.setMask("UUU-####"); // mascara para placa

mPLACA.setPlaceholderCharacter('\_'); // caracter que fica ocupando o espaço

} catch (ParseException e) {

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

placa = new JFormattedTextField(mPLACA);

encerrar = new javax.swing.JButton();

cancelar = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Estacionamento JG - Registro de Saída");

jLabel1.setFont(new java.awt.Font("Calibri", 0, 24)); // NOI18N

jLabel1.setText("Placa");

placa.setFont(new java.awt.Font("Calibri", 1, 48)); // NOI18N

encerrar.setBackground(new java.awt.Color(51, 153, 255));

encerrar.setFont(new java.awt.Font("Calibri", 1, 14)); // NOI18N

encerrar.setForeground(new java.awt.Color(255, 255, 255));

encerrar.setText("Encerrar hospedagem de veículo");

encerrar.setToolTipText("");

encerrar.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

encerrarActionPerformed(evt);

}

});

cancelar.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

cancelar.setText("Cancelar");

cancelar.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

cancelarActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(47, 47, 47)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 63, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(placa, javax.swing.GroupLayout.PREFERRED\_SIZE, 216, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(encerrar)

.addGap(36, 36, 36)

.addComponent(cancelar)

.addGap(53, 53, 53))))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(73, 73, 73)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 31, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(placa, javax.swing.GroupLayout.PREFERRED\_SIZE, 58, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 60, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(encerrar, javax.swing.GroupLayout.PREFERRED\_SIZE, 52, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(cancelar, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(57, 57, 57))

);

pack();

}// </editor-fold>

private void encerrarActionPerformed(java.awt.event.ActionEvent evt) {

Veiculo carro = new Veiculo(); // cria objeto carro da classe veiculo

BD registra = new BD(); // cria objeto registra da classe bd

carro.setPlaca(placa.getText());// pega valor da placa

carro.setSaida();//seta o valor de saida

carro.getSaida(); // inicia data e hora de saida

Boolean a = true;

for (int i = 0; i < placa.getText().length(); i++) {//laco com condicao para nao deixar registar se o usuario for registrar a placa com \_

if (placa.getText().charAt(i) == '\_') {

a = false;

}

}

if (a == true) {//condicao registrar a saida do carro

Boolean result = registra.Encerrado(carro.getPlaca(), carro.getSaida()); // insere carro encerrado na tabela encerrados do bd

if (result == true) {//se foi registrado

this.dispose(); // fechar janela

}

}

else

JOptionPane.showMessageDialog(null, "Erro, placa inválida.", "Erro", JOptionPane.ERROR\_MESSAGE);

}

private void cancelarActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose(); // acao de fechar janela no botao cancelar

}

// Variables declaration - do not modify

private javax.swing.JButton cancelar;

private javax.swing.JButton encerrar;

private javax.swing.JLabel jLabel1;

private javax.swing.JFormattedTextField placa;

// End of variables declaration

}

package estacionamentojg;

import java.text.ParseException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFormattedTextField;

import javax.swing.text.MaskFormatter;

public class RelatorioDia extends javax.swing.JFrame {

MaskFormatter mData = new MaskFormatter();//cria objeto mData da classe MaskFormatter

public RelatorioDia() { //cria construtor

initComponents();

setLocationRelativeTo(null); // coloca janela no centro da pagina

setVisible(true);

try {

mData.setMask("##/##/####"); // mascara para placa

mData.setPlaceholderCharacter('\_'); // caracter que fica ocupando o espaço

} catch (ParseException e) {

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jTextField1 = new JFormattedTextField(mData);

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setFont(new java.awt.Font("Calibri", 0, 20)); // NOI18N

jLabel1.setText("Data");

jTextField1.setFont(new java.awt.Font("Calibri", 0, 20)); // NOI18N

jButton1.setFont(new java.awt.Font("Calibri", 0, 15)); // NOI18N

jButton1.setText("Ok");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setFont(new java.awt.Font("Calibri", 0, 15)); // NOI18N

jButton2.setText("Cancelar");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(83, 83, 83)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 52, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(21, 21, 21))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap()

.addComponent(jButton1)

.addGap(6, 6, 6)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 145, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addGap(45, 45, 45)

.addComponent(jButton2)))

.addContainerGap(99, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(82, 82, 82)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 39, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 39, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(28, 28, 28)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1)

.addComponent(jButton2))

.addContainerGap(124, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose();//acao de fechar para o botao cancelar

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

//acao do botao OK

String data = jTextField1.getText(); //variavel data recebe valor do campo data

GeraRelatorio rDiario = new GeraRelatorio(); //cria objeto rDiario da classe GeraRelatorio

try {

rDiario.diario(data); //objeto rdiario chama o metodo diario passando o valor data

} catch (Exception ex) {

Logger.getLogger(RelatorioDia.class.getName()).log(Level.SEVERE, null, ex);

}

this.dispose();//fechar a janela

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JTextField jTextField1;

// End of variables declaration

}

package estacionamentojg;

import java.text.ParseException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFormattedTextField;

import javax.swing.text.MaskFormatter;

public class RelatorioMes extends javax.swing.JFrame {

MaskFormatter mMesAno = new MaskFormatter();

public RelatorioMes() {

initComponents();

setLocationRelativeTo(null); // coloca janela no centro da pagina

setVisible(true);

try {

mMesAno.setMask("##/####"); // mascara para mesAno

mMesAno.setPlaceholderCharacter('\_'); // caracter que fica ocupando o espaço

} catch (ParseException e) {

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jTextField1 = new JFormattedTextField(mMesAno);

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setFont(new java.awt.Font("Calibri", 0, 20)); // NOI18N

jLabel1.setText("Mes/Ano");

jTextField1.setFont(new java.awt.Font("Calibri", 0, 20)); // NOI18N

jButton1.setFont(new java.awt.Font("Calibri", 0, 15)); // NOI18N

jButton1.setText("Ok");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setFont(new java.awt.Font("Calibri", 0, 15)); // NOI18N

jButton2.setText("Cancelar");

jButton2.setToolTipText("");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 57, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addGap(77, 77, 77)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 93, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(18, 18, 18)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 146, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(60, 60, 60)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 89, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(107, 107, 107))))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(93, 93, 93)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 26, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(39, 39, 39)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton2)

.addComponent(jButton1))

.addContainerGap(109, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

String mesAno = jTextField1.getText(); //string mesAno pega o valor do campo jtextfield1

GeraRelatorio rMensal = new GeraRelatorio();//cria um objeto rMensal da classe GeraRelatorio

try {

rMensal.mensal(mesAno);//objeto rmensal chama o metodo mensal passando o valor digitado no campo

} catch (Exception ex) {

Logger.getLogger(RelatorioMes.class.getName()).log(Level.SEVERE, null, ex);

}

this.dispose();//fecha a janela

}

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new RelatorioMes().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JTextField jTextField1;

// End of variables declaration

}

package estacionamentojg;

import java.awt.Color;

import java.sql.Timestamp;

import javax.swing.JLabel;

public class Ticket extends javax.swing.JFrame {

private String modelo;

private String placa;

private String cor;

private String data;

private String dia;

private Timestamp datahora;

private JLabel jModelo = new javax.swing.JLabel(); //cria objeto jmodelo da classe jlabel

private JLabel jPlaca = new javax.swing.JLabel();//cria objeto jplaca da classe jlabel

private JLabel jCor = new javax.swing.JLabel();//cria objeto jcor da classe jlabel

private JLabel jDatahora = new javax.swing.JLabel();//cria objeto jdatahora da classe jlabel

public void setModelo(String modelo) { //metodo para setar o modelo

this.modelo = modelo;

}

public void setPlaca(String placa){ //metodo para setar a placa

this.placa = placa;

}

public void setCor(String cor) { //metodo para setar a cor

this.cor = cor;

}

public void setDia(String dia) { //metodo para setar dia

this.dia = dia;

}

public void setDatahora(Timestamp datahora) { //metodo para setar data e hora

this.datahora = datahora;

}

public String timestampToString() { //metodo para transformar a data em string

this.data = "" + this.datahora; //concatena a string com a datahora

return this.data;

}

public void textoJLabel() { //metodo para configurar o modelo,placa,cor e data

jModelo.setFont(new java.awt.Font("Calibri", 0, 18));

jModelo.setText(modelo);

jPlaca.setFont(new java.awt.Font("Calibri", 0, 48));

jPlaca.setText(placa);

jCor.setFont(new java.awt.Font("Calibri", 0, 18));

jCor.setText(cor);

jDatahora.setFont(new java.awt.Font("Calibri", 0, 18));

jDatahora.setText(timestampToString() + " " + this.dia);

}

public void emite() { //metodo para emitir ticket

textoJLabel(); //chama metodo para setar os dados

initComponents();

setLocationRelativeTo(null); //nao precisa posicionar no centro

this.setVisible(true);

this.setSize(300, 400);//define tamanho da janela

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLayeredPane1 = new javax.swing.JLayeredPane();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jLabel7 = jDatahora;

jLabel8 = jPlaca;

jLabel9 = jModelo;

jLabel10 = jCor;

jLabel11 = new javax.swing.JLabel();

fechar = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Estacionamento JG - Ticket");

setBackground(new java.awt.Color(255, 255, 255));

jLabel1.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

jLabel1.setText("Ticket comprovante de estacionamento");

jLabel2.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

jLabel2.setText("Placa:");

jLabel3.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

jLabel3.setText("modelo:");

jLabel4.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

jLabel4.setText("cor:");

jLabel5.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

jLabel5.setText("Estacionamento JG LTDA");

jLabel5.setToolTipText("");

jLabel6.setFont(new java.awt.Font("Calibri", 0, 12)); // NOI18N

jLabel6.setText("Data/hora de ent.:");

jLabel7.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

jLabel8.setFont(new java.awt.Font("Calibri", 1, 18)); // NOI18N

jLabel9.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

jLabel10.setFont(new java.awt.Font("Calibri", 0, 14)); // NOI18N

jLabel11.setText("Obrigado pela preferência.");

fechar.setText("Fechar");

fechar.setToolTipText("");

fechar.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

fecharActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED\_SIZE, 209, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(35, 35, 35)

.addComponent(jLabel5))

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 232, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addGap(2, 2, 2)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED\_SIZE, 123, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel2)

.addGap(18, 18, 18)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED\_SIZE, 153, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel4)

.addGap(18, 18, 18)

.addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED\_SIZE, 120, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel11)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(fechar)))))

.addContainerGap(418, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel5)

.addGap(2, 2, 2)

.addComponent(jLabel1)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(28, 28, 28)

.addComponent(jLabel6))

.addGroup(layout.createSequentialGroup()

.addGap(18, 18, 18)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED\_SIZE, 32, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(23, 23, 23)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED\_SIZE, 40, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(33, 33, 33)

.addComponent(jLabel2)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(30, 30, 30)

.addComponent(jLabel3))

.addGroup(layout.createSequentialGroup()

.addGap(18, 18, 18)

.addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(44, 44, 44)

.addComponent(jLabel4))

.addGroup(layout.createSequentialGroup()

.addGap(34, 34, 34)

.addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED\_SIZE, 36, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(26, 26, 26)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 27, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(fechar))

.addContainerGap(256, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void fecharActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose();//acao de fechar para o botao fechar

}

// Variables declaration - do not modify

private javax.swing.JButton fechar;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JLayeredPane jLayeredPane1;

// End of variables declaration

}

package estacionamentojg;

import java.sql.Timestamp;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Locale;

public class Valores {

private Double precoHora;

private Double primeiraHora;

private Double precoDomingo = 3.0;

private Double precoTotal = 0.0;

Valores(Double pP, Double pH) { //construtor para receber valores da primeiraHora e precoHora definidos

this.precoHora = pH;

this.primeiraHora = pP;

}

public Double total(Timestamp ent, Timestamp sai) { //metodo para calcular os valor total

Calendar entrada = Calendar.getInstance(); //cria um objeto da classe calendar

entrada.setTime(ent);

Calendar saida = Calendar.getInstance(); //cria um objeto da classe calendar

saida.setTime(sai);

long diferenca = saida.getTimeInMillis() - entrada.getTimeInMillis(); // Calcula a diferença de dias entre a saida e da data de entrada

int tempoHora = 1000 \* 60 \* 60; // Quantidade de milissegundos em uma hora

Double totalHoras = (double) diferenca / tempoHora; //calculo das horas

totalHoras = Math.ceil(totalHoras); // arredonda horas para cima

if (entrada.get(Calendar.DATE) == saida.get(Calendar.DATE)) { // se for o mesmo dia da entrada e da saida

if (entrada.get(Calendar.DAY\_OF\_WEEK) == 1) { //se for domingo

this.precoTotal = this.precoDomingo; //pega valor que foi definido para o domingo que é unico

} else {

if (entrada.get(Calendar.HOUR) == saida.get(Calendar.HOUR)) // se entrou e saiu na mesma hora cheia

{

this.precoTotal = this.primeiraHora; // pega o valor da primeira hora

} else {

this.precoTotal = this.primeiraHora + ((totalHoras - 1) \* this.precoHora); // soma total de horas com o preco normal

}

}

} else { // se for dias diferentes

Double fechamentoDia = (double) 24 - entrada.get(Calendar.HOUR\_OF\_DAY); // quantia de horas do primeiro dia

totalHoras -= fechamentoDia; // calcula o restante de horas pendentes

ArrayList<Double> dias = new ArrayList();//cria um arraylist de dias

if (entrada.get(Calendar.DAY\_OF\_WEEK) == 1) { // se a entrada for domingo

dias.add(this.precoDomingo); // add precoDomingo no arrayList

} else { // se não for domingo

dias.add(((fechamentoDia - 1) \* this.precoHora) + this.primeiraHora);//calcula o valor do dia - 1 pq a primeira hora n conta

}

while (true) {

entrada.add(Calendar.DATE, +1); // vai para o proximo dia

if (totalHoras > 24) { // se faltar mais de um dia | não for ultimo dia

if (entrada.get(Calendar.DAY\_OF\_WEEK) == 1) { // se for domingo

dias.add(this.precoDomingo);//adiciona no array o preco de domingo

} else {

dias.add((23 \* this.precoHora) + this.primeiraHora); // calcula o preco do dia inteiro

}

totalHoras -= 24; // diminuir 24 horas do total

} else { // se for o ultimo dia

if (entrada.get(Calendar.DAY\_OF\_WEEK) == 1) { // domingo

dias.add(this.precoDomingo);//add o preco de domingo no array

} else if (entrada.get(Calendar.HOUR\_OF\_DAY) == saida.get(Calendar.HOUR\_OF\_DAY)) {//se for na mesma hora a entrada e a saida

dias.add(this.primeiraHora); // add a valor da primeira hora no array

} else if (entrada.get(Calendar.HOUR\_OF\_DAY) != saida.get(Calendar.HOUR\_OF\_DAY)) {//se n for na mesma hora

dias.add(((totalHoras - 1) \* this.precoHora) + this.primeiraHora); // calcula o preco pelo restante de horas

}

break;

}

}

for (Double n : dias) { // add todos os valores do arrrayList

this.precoTotal += n; // soma todos os valores do arraylist

}

}

return precoTotal; // retorna preco total

}

}

package estacionamentojg;

import java.sql.Timestamp;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Locale;

public class Veiculo {

private String placa;

private String modelo;

private String cor;

private Timestamp timestampINICIAL;

private Timestamp timestampFINAL;

private Date dataInicial;

private Date dataFinal;

public String getPlaca() { //metodo para pegar a placa

return placa;

}

public void setPlaca(String placa) { //metodo para setar a placa

this.placa = placa;

}

public String getModelo() { //metodo para pegar o modelo

return modelo;

}

public void setModelo(String modelo) { //metodo para setar o modelo

this.modelo = modelo;

}

public String getCor() { //metodo para pegar a cor

return cor;

}

public void setCor(String cor) { //metodo para setar a cor

this.cor = cor;

}

public void setEntrada() { //metodo para setar a data de entrada

dataInicial = new Date();

timestampINICIAL = new Timestamp(dataInicial.getTime());

}

public void setSaida() { //metodo para setar a data de saida

dataFinal = new Date();

timestampFINAL = new Timestamp(dataFinal.getTime());

}

public Timestamp getEntrada() { // metodo para pegar a data e hora de entrada

return timestampINICIAL;

}

public Timestamp getSaida() { // metodo para pegar a data e hora de saida

return timestampFINAL;

}

public String diaDaSemanaInicial() { //metodo para transformar o dia da semana em portugues

Locale local1 = new Locale("pt", "BR"); // transforma para Português o dia da semana, Brasil

String dia = new SimpleDateFormat("EEE", local1).format(dataInicial); // pega a string do dia da semana

return dia;

}

}