****

**OOP Project**

Group Name: Hajra Batool, Shiza Fatima

Department: SE-2nd

To: Mr.Usman

Roll no.221, 209

Project name: Workshop for Vehicles

Coding:-

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.util.HashMap;

import java.util.Map;

public class CustomerForm implements ActionListener {

JFrame frame = new JFrame("Vehicle Workshop");

ImageIcon icon1 = new ImageIcon("C:\\Users\\Hajra SE\\Pictures\\vehicle.PNG");

JButton doneButton = new JButton("Done");

JButton submitButton = new JButton("Submit");

JPanel cardsPanel = new JPanel(new CardLayout());

JPanel customerDetailsPanel = new JPanel(new SpringLayout());

JPanel optionsPanel = new JPanel(new SpringLayout());

JLabel titleLabel = new JLabel("Customer Detail");

JTextField firstNameField = new JTextField();

JTextField emailField = new JTextField();

JTextField contactNumberField = new JTextField();

JLabel firstNameLabel = new JLabel("First Name:");

JLabel emailLabel = new JLabel("Email:");

JLabel contactNumberLabel = new JLabel("Contact Number:");

String[] receiveOptions = {"D or R", "Home Delivery", "Self Receive"};

String[] serviceOptions = {"Choose Services", "Oil Change", "Car Wash", "Tire Air Check", "Wheel Balancing", "Engine Repair"};

JComboBox<String> receiveComboBox = new JComboBox<>(receiveOptions);

JComboBox<String> serviceComboBox = new JComboBox<>(serviceOptions);

JRadioButton cashRadioButton = new JRadioButton("Cash");

JRadioButton onlineRechargeRadioButton = new JRadioButton("Online Recharge");

ButtonGroup paymentMethodGroup = new ButtonGroup();

JLabel iconLabel;

JLabel lp = new JLabel("Select Options");

ImageIcon icon2 = new ImageIcon("C:\\Users\\Hajra SE\\Pictures\\c.PNG");

ImageIcon largeIcon;

JLabel largeIconLabel;

Map<String, Integer> serviceRecharge = new HashMap<>();

public CustomerForm() {

frame.setIconImage(icon1.getImage());

frame.setSize(600, 700);

frame.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

serviceRecharge.put("Oil Change", 50);

serviceRecharge.put("Car Wash", 20);

serviceRecharge.put("Tire Air Check", 10);

serviceRecharge.put("Wheel Balancing", 40);

serviceRecharge.put("Engine Repair", 150);

Image scaledImage = icon2.getImage().getScaledInstance(50, 50, Image.SCALE\_SMOOTH);

iconLabel = new JLabel(new ImageIcon(scaledImage));

File largeIconFile = new File("C:\\Users\\Hajra SE\\Pictures\\vehicle.PNG");

if (largeIconFile.exists()) {

largeIcon = new ImageIcon(largeIconFile.getPath());

Image scaledLargeImage = largeIcon.getImage().getScaledInstance(150, 150, Image.SCALE\_SMOOTH);

largeIconLabel = new JLabel(new ImageIcon(scaledLargeImage));

} else {

System.err.println("Error: Large icon image not found at " + largeIconFile.getPath());

largeIconLabel = new JLabel("Image not found");

}

SpringLayout customerLayout = new SpringLayout();

customerDetailsPanel.setLayout(customerLayout);

customerDetailsPanel.setBackground(Color.gray);

customerDetailsPanel.add(firstNameLabel);

customerDetailsPanel.add(firstNameField);

customerDetailsPanel.add(emailLabel);

customerDetailsPanel.add(emailField);

customerDetailsPanel.add(contactNumberLabel);

customerDetailsPanel.add(contactNumberField);

customerDetailsPanel.add(doneButton);

customerLayout.putConstraint(SpringLayout.WEST, firstNameLabel, 10, SpringLayout.WEST, customerDetailsPanel);

customerLayout.putConstraint(SpringLayout.NORTH, firstNameLabel, 20, SpringLayout.NORTH, customerDetailsPanel);

customerLayout.putConstraint(SpringLayout.WEST, firstNameField, 150, SpringLayout.WEST, customerDetailsPanel);

customerLayout.putConstraint(SpringLayout.NORTH, firstNameField, 20, SpringLayout.NORTH, customerDetailsPanel);

firstNameField.setPreferredSize(new Dimension(300, 30));

customerLayout.putConstraint(SpringLayout.WEST, emailLabel, 10, SpringLayout.WEST, customerDetailsPanel);

customerLayout.putConstraint(SpringLayout.NORTH, emailLabel, 70, SpringLayout.NORTH, customerDetailsPanel);

customerLayout.putConstraint(SpringLayout.WEST, emailField, 150, SpringLayout.WEST, customerDetailsPanel);

customerLayout.putConstraint(SpringLayout.NORTH, emailField, 70, SpringLayout.NORTH, customerDetailsPanel);

emailField.setPreferredSize(new Dimension(300, 30));

customerLayout.putConstraint(SpringLayout.WEST, contactNumberLabel, 10, SpringLayout.WEST, customerDetailsPanel);

customerLayout.putConstraint(SpringLayout.NORTH, contactNumberLabel, 120, SpringLayout.NORTH, customerDetailsPanel);

customerLayout.putConstraint(SpringLayout.WEST, contactNumberField, 150, SpringLayout.WEST, customerDetailsPanel);

customerLayout.putConstraint(SpringLayout.NORTH, contactNumberField, 120, SpringLayout.NORTH, customerDetailsPanel);

contactNumberField.setPreferredSize(new Dimension(300, 30));

customerLayout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, doneButton, 0, SpringLayout.HORIZONTAL\_CENTER, customerDetailsPanel);

customerLayout.putConstraint(SpringLayout.NORTH, doneButton, 180, SpringLayout.NORTH, customerDetailsPanel);

SpringLayout optionsLayout = new SpringLayout();

optionsPanel.setLayout(optionsLayout);

optionsPanel.setBackground(Color.gray);

optionsPanel.add(lp);

optionsPanel.add(serviceComboBox);

optionsPanel.add(receiveComboBox);

optionsPanel.add(cashRadioButton);

optionsPanel.add(onlineRechargeRadioButton);

optionsPanel.add(submitButton);

optionsPanel.add(largeIconLabel);

lp.setFont(new Font("Arial", Font.BOLD, 35));

optionsLayout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, lp, 0, SpringLayout.HORIZONTAL\_CENTER, optionsPanel);

optionsLayout.putConstraint(SpringLayout.NORTH, lp, 20, SpringLayout.NORTH, optionsPanel);

optionsLayout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, serviceComboBox, 0, SpringLayout.HORIZONTAL\_CENTER, optionsPanel);

optionsLayout.putConstraint(SpringLayout.NORTH, serviceComboBox, 10, SpringLayout.SOUTH, lp);

optionsLayout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, receiveComboBox, 0, SpringLayout.HORIZONTAL\_CENTER, optionsPanel);

optionsLayout.putConstraint(SpringLayout.NORTH, receiveComboBox, 10, SpringLayout.SOUTH, serviceComboBox);

optionsLayout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, cashRadioButton, -50, SpringLayout.HORIZONTAL\_CENTER, optionsPanel);

optionsLayout.putConstraint(SpringLayout.NORTH, cashRadioButton, 10, SpringLayout.SOUTH, receiveComboBox);

optionsLayout.putConstraint(SpringLayout.WEST, onlineRechargeRadioButton, 10, SpringLayout.EAST, cashRadioButton);

optionsLayout.putConstraint(SpringLayout.NORTH, onlineRechargeRadioButton, 0, SpringLayout.NORTH, cashRadioButton);

optionsLayout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, submitButton, 0, SpringLayout.HORIZONTAL\_CENTER, optionsPanel);

optionsLayout.putConstraint(SpringLayout.NORTH, submitButton, 20, SpringLayout.SOUTH, cashRadioButton);

optionsLayout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, largeIconLabel, 0, SpringLayout.HORIZONTAL\_CENTER, optionsPanel);

optionsLayout.putConstraint(SpringLayout.NORTH, largeIconLabel, 20, SpringLayout.SOUTH, submitButton);

paymentMethodGroup.add(cashRadioButton);

paymentMethodGroup.add(onlineRechargeRadioButton);

JPanel titlePanel = new JPanel();

titlePanel.setLayout(new FlowLayout(FlowLayout.CENTER));

titlePanel.add(titleLabel);

titlePanel.add(iconLabel);

titleLabel.setFont(new Font("Arial", Font.BOLD, 25));

cardsPanel.add(customerDetailsPanel, "customerDetails");

cardsPanel.add(optionsPanel, "options");

frame.getContentPane().add(cardsPanel, BorderLayout.CENTER);

frame.add(titlePanel, BorderLayout.NORTH);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

doneButton.addActionListener(this);

submitButton.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == doneButton) {

CardLayout cardLayout = (CardLayout) cardsPanel.getLayout();

cardLayout.show(cardsPanel, "options");

} else if (e.getSource() == submitButton) {

String firstName = firstNameField.getText();

String email = emailField.getText();

String contact = contactNumberField.getText();

if (firstNameField.getText().isEmpty() || emailField.getText().isEmpty() || contactNumberField.getText().isEmpty()) {

JOptionPane.showMessageDialog(frame, "Every text field must be filled.");

return;

}

String receiveOption = (String) receiveComboBox.getSelectedItem();

String service = (String) serviceComboBox.getSelectedItem();

String paymentMethod = cashRadioButton.isSelected() ? "Cash" : onlineRechargeRadioButton.isSelected() ? "Online Recharge" : "Not specified";

int serviceFee = serviceRecharge.getOrDefault(service, 0);

int deliveryFee = "Home Delivery".equals(receiveOption) ? 50 : 0;

int totalRecharges = serviceFee + deliveryFee;

String message = "Your information is saved successfully.\n" +

"First Name: " + firstName + "\n" +

"Email: " + email + "\n" +

"Contact: " + contact + "\n" +

"Option for vehicle receive: " + receiveOption + "\n" +

"Service: " + service + "\n" +

"Service Fee: $" + serviceFee + "\n" +

"Delivery Fee: $" + deliveryFee + "\n" +

"Total Recharges: $" + totalRecharges + "\n" +

"Payment Method: " + paymentMethod;

JOptionPane.showMessageDialog(frame, message, "Customer Information", JOptionPane.INFORMATION\_MESSAGE);

System.out.println("Your information has been saved successfully. Which is given below:");

System.out.println("First Name: " + firstName);

System.out.println("Email: " + email);

System.out.println("Contact: " + contact);

System.out.println("Option for vehicle receive: " + receiveOption);

System.out.println("Service: " + service);

System.out.println("Service Fee: $" + serviceFee);

System.out.println("Delivery Fee: $" + deliveryFee);

System.out.println("Total Recharges: $" + totalRecharges);

System.out.println("Payment Method: " + paymentMethod);

System.out.println("Thank you for choosing this service center.");

}

}

}

***Another class and link it with other :-***

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.HashMap;

import java.util.Map;

public class parkingsetupApp extends JFrame {

private boolean spaceAvailable = true;

private Map<String, Double> rates;

private JLabel label1;

private JLabel vehicleLabel;

private JLabel timeLabel;

private JTextField timeField;

private JTextField vehicleField;

private JButton checkButton;

private JButton parkButton;

private JButton calculateBillButton;

private JButton customerFormButton;

private JLabel resultLabel;

private JLabel billLabel;

private JPanel frame;

public parkingsetupApp() {

setTitle("Parking System App");

setSize(500, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

label1 = new JLabel("Parking System App");

vehicleLabel = new JLabel("Vehicle Type:");

timeLabel = new JLabel("Parking Time (hrs):");

timeField = new JTextField();

vehicleField = new JTextField();

checkButton = new JButton("Check Space");

parkButton = new JButton("Park Vehicle");

calculateBillButton = new JButton("Calculate Bill");

customerFormButton = new JButton("Open Customer Form"); // Initialize new button

resultLabel = new JLabel("");

billLabel = new JLabel("");

label1.setFont(new Font("Serif", Font.BOLD | Font.ITALIC, 25));

label1.setHorizontalAlignment(JLabel.CENTER);

label1.setBorder(BorderFactory.createEmptyBorder(20, 0, 20, 0));

frame = new JPanel();

frame.setLayout(new SpringLayout());

frame.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

frame.setBackground(Color.pink);

SpringLayout layout = (SpringLayout) frame.getLayout();

frame.add(label1);

frame.add(vehicleLabel);

frame.add(vehicleField);

frame.add(timeLabel);

frame.add(timeField);

frame.add(checkButton);

frame.add(parkButton);

frame.add(calculateBillButton);

frame.add(customerFormButton);

frame.add(resultLabel);

frame.add(billLabel);

layout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, label1, 0, SpringLayout.HORIZONTAL\_CENTER, frame);

layout.putConstraint(SpringLayout.NORTH, label1, 10, SpringLayout.NORTH, frame);

layout.putConstraint(SpringLayout.WEST, vehicleLabel, 10, SpringLayout.WEST, frame);

layout.putConstraint(SpringLayout.NORTH, vehicleLabel, 10, SpringLayout.SOUTH, label1);

layout.putConstraint(SpringLayout.WEST, vehicleField, 10, SpringLayout.EAST, vehicleLabel);

layout.putConstraint(SpringLayout.NORTH, vehicleField, 10, SpringLayout.SOUTH, label1);

layout.putConstraint(SpringLayout.EAST, vehicleField, -10, SpringLayout.EAST, frame);

layout.putConstraint(SpringLayout.WEST, timeLabel, 10, SpringLayout.WEST, frame);

layout.putConstraint(SpringLayout.NORTH, timeLabel, 10, SpringLayout.SOUTH, vehicleLabel);

layout.putConstraint(SpringLayout.WEST, timeField, 10, SpringLayout.EAST, timeLabel);

layout.putConstraint(SpringLayout.NORTH, timeField, 10, SpringLayout.SOUTH, vehicleField);

layout.putConstraint(SpringLayout.EAST, timeField, -10, SpringLayout.EAST, frame);

layout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, checkButton, 0, SpringLayout.HORIZONTAL\_CENTER, frame);

layout.putConstraint(SpringLayout.NORTH, checkButton, 20, SpringLayout.SOUTH, timeField);

layout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, parkButton, 0, SpringLayout.HORIZONTAL\_CENTER, frame);

layout.putConstraint(SpringLayout.NORTH, parkButton, 10, SpringLayout.SOUTH, checkButton);

layout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, calculateBillButton, 0, SpringLayout.HORIZONTAL\_CENTER, frame);

layout.putConstraint(SpringLayout.NORTH, calculateBillButton, 10, SpringLayout.SOUTH, parkButton);

layout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, customerFormButton, 0, SpringLayout.HORIZONTAL\_CENTER, frame);

layout.putConstraint(SpringLayout.NORTH, customerFormButton, 10, SpringLayout.SOUTH, calculateBillButton);

layout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, resultLabel, 0, SpringLayout.HORIZONTAL\_CENTER, frame);

layout.putConstraint(SpringLayout.NORTH, resultLabel, 10, SpringLayout.SOUTH, customerFormButton);

layout.putConstraint(SpringLayout.HORIZONTAL\_CENTER, billLabel, 0, SpringLayout.HORIZONTAL\_CENTER, frame);

layout.putConstraint(SpringLayout.NORTH, billLabel, 10, SpringLayout.SOUTH, resultLabel);

add(frame);

rates = new HashMap<>();

rates.put("Car", 10.0);

rates.put("Bike", 5.0);

rates.put("Truck", 20.0);

checkButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

checkSpace();

}

});

parkButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

parkVehicle();

}

});

calculateBillButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

calculateBill();

}

});

customerFormButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

openCustomerForm();

}

});

setVisible(true);

}private void checkSpace() {

if (spaceAvailable) {

resultLabel.setText("Space is available for parking.");

} else {

resultLabel.setText("No space available for parking.");

}

}private void parkVehicle() {

if (spaceAvailable) {

spaceAvailable = false;

resultLabel.setText("Vehicle parked successfully.");

} else {

resultLabel.setText("No space available to park the vehicle.");

}

}private void calculateBill() {

String vehicleType = vehicleField.getText();

String timeText = timeField.getText();

double rate = rates.getOrDefault(vehicleType, 0.0);

if (rate == 0.0) {

billLabel.setText("Invalid vehicle type.");

return;

}

try {

double time = Double.parseDouble(timeText);

double bill = rate \* time;

billLabel.setText("Total bill: $" + bill);

} catch (NumberFormatException e) {

billLabel.setText("Invalid time input.");

}

}private void openCustomerForm() {

this.setVisible(false);

new CustomerForm();

}public static void main(String[] args) {

new parkingsetupApp();

}

}