

**FINAL PROJECT**

**OBJECT ORINETED PROGRAMMING**

**DATE OF SUBMISSION**

6th June 2023

**PREPARED FOR**

Mam Sajida Kalsoom

**PREPARED BY**

SADIA IMRAN

SP22-BCS-087

BCS-3B

# **Automatic Car Wash System**

import javax.swing.\*;  
import java.io.\*;  
import java.io.File;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Scanner;  
import java.util.Date;  
import java.nio.file.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
  
class CustomerWindow extends CustomerPanel implements ActionListener {  
 private Font font = new Font("Consolas", Font.*BOLD*, 40);  
 private Font buttonFont = new Font("Consolas", Font.*BOLD*, 14);  
 private static final Color *SKY\_BLUE\_COLOR* = new Color(135, 206, 235);  
 private static final Color *WHITE\_COLOR* = Color.*WHITE*;  
 JFrame jFrame ;  
 JPanel customerPanel , functionPanel , carWashPanel ,waitingPanel,slotsPanel,billPanel;  
 JButton slotsButton,waitingButton,carWashButton,billButton ,submitButton , goBackButton , addButton;  
 JTextField packageTextField , memberTextField;  
 JTextArea datArea ;  
  
 CustomerWindow(){  
 super();  
 jFrame = new JFrame("Customer");  
 jFrame.setSize(800,600);  
 jFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
  
 customerPanel = new JPanel(null); // Set layout manager to null  
 customerPanel.setBounds(0, 0, jFrame.getWidth() / 3, jFrame.getHeight());  
 customerPanel.setBackground(*SKY\_BLUE\_COLOR*.darker());  
  
 functionPanel = new JPanel(null);  
 functionPanel.setBackground(*WHITE\_COLOR*);  
 functionPanel.setBounds(0,0,jFrame.getWidth()-customerPanel.getWidth(),jFrame.getHeight());  
  
 datArea = new JTextArea("Your text goes here");  
 datArea.setFont(buttonFont);  
 datArea.setBounds(customerPanel.getWidth()+30,50,jFrame.getWidth()-customerPanel.getWidth(),jFrame.getHeight());  
 functionPanel.add(datArea);  
 // Add admin icon and label  
 ImageIcon *adminIcon* = new ImageIcon("client.png"); // Replace with the path to your admin icon image  
  
 // Image scaledIcon = adminIcon.getImage().getScaledInstance(50, 50, Image.SCALE\_SMOOTH);  
  
  
 JLabel *adminLabel* = new JLabel(*adminIcon*, JLabel.*CENTER*);  
 *adminLabel*.setIcon(*adminIcon*);  
 *adminLabel*.setFont(new Font("Arial", Font.*BOLD*, 24));  
 *adminLabel*.setForeground(Color.*WHITE*);  
 *adminLabel*.setBounds(10, 10, customerPanel.getWidth() - 40, 100);  
 customerPanel.add(*adminLabel*);  
  
  
 carWashButton = new JButton("Car Wash");  
 carWashButton.setBackground(Color.*WHITE*.brighter());  
 carWashButton.setForeground(Color.*BLACK*);  
 carWashButton.addActionListener(this);  
 carWashButton.setBounds(50, 150, customerPanel.getWidth() - 120, 30);  
 customerPanel.add(carWashButton);  
  
 // Add search button  
 waitingButton = new JButton("Waiting Customer");  
 waitingButton.setBackground(Color.*WHITE*.brighter());  
 waitingButton.setForeground(Color.*BLACK*);  
 waitingButton.addActionListener(this);  
 waitingButton.setBounds(50, 200, customerPanel.getWidth() - 120, 30);  
 customerPanel.add(waitingButton);  
  
 // Add delete button  
 billButton = new JButton("Pay Bill");  
 billButton.setBackground(Color.*WHITE*.brighter());  
 billButton.setForeground(Color.*BLACK*);  
 billButton.addActionListener(this);  
 billButton.setBounds(50, 250, customerPanel.getWidth() - 120, 30);  
 customerPanel.add(billButton);  
  
 slotsButton = new JButton("Available Slots");  
 slotsButton.setBackground(Color.*WHITE*.brighter());  
 slotsButton.setForeground(Color.*BLACK*);  
 slotsButton.addActionListener(this);  
 slotsButton.setBounds(50, 300, customerPanel.getWidth() - 120, 30);  
 customerPanel.add(slotsButton);  
  
 goBackButton = new JButton("Back");  
 goBackButton.setBackground(Color.*WHITE*.brighter());  
 goBackButton.setForeground(Color.*BLACK*);  
 goBackButton.addActionListener(this);  
 goBackButton.setBounds(50, 350, customerPanel.getWidth() - 120, 30);  
 customerPanel.add(goBackButton);  
  
 jFrame.add(customerPanel);  
 jFrame.add(functionPanel);  
 jFrame.setVisible(true);  
 }  
 @Override  
 public void actionPerformed(ActionEvent *e*) {  
 String *actionCommand* = *e*.getActionCommand();  
 if(*e*.getSource()==carWashButton){  
 carWashPanel = new JPanel();  
 carWashPanel.setBounds(customerPanel.getWidth(),0,jFrame.getWidth()-customerPanel.getWidth(),jFrame.getHeight());  
 carWashPanel.setLayout(null);  
 String *data* = "";  
 for(int *i* = 0 , *p*= 1 ; *i*<Globals.*readPackages*.size();*i*++,*p*++){  
  
 *data* += "Package "+*p*+" Details : \n"+"Name : "+Globals.*readPackages*.get(*i*).getName()+" \nDescription : "  
 +Globals.*readPackages*.get(*i*).getDescription()+" \nPrice : "+Globals.*readPackages*.get(*i*).getPrice()+" \n" +  
 "------------------------------------------------\n";  
 }  
  
 JLabel *packageLabel* = new JLabel("Enter Package Number: ");  
 *packageLabel*.setForeground(Color.*BLACK*);  
 *packageLabel*.setBounds(functionPanel.getX()+30, 30, 200, 30); // Set coordinates relative to carWashPanel  
 carWashPanel.add(*packageLabel*);  
 packageTextField = new JTextField();  
 packageTextField.setBounds(functionPanel.getX() + 30, carWashPanel.getY() + 50 + 30, 200, 30);  
 packageTextField.addActionListener(this);  
 carWashPanel.add(packageTextField);  
  
 submitButton = new JButton("Submit");  
  
 submitButton.setBounds(packageTextField.getWidth()+70, packageTextField.getY(), 100, 30);  
 submitButton.addActionListener(this);  
 submitButton.setBackground(*SKY\_BLUE\_COLOR*);  
 submitButton.setForeground(Color.*BLACK*);  
 carWashPanel.add(submitButton);  
  
 datArea.setBounds(carWashPanel.getX()+30,carWashPanel.getY()+50+30+50,jFrame.getWidth()-customerPanel.getWidth(),jFrame.getHeight());  
 functionPanel.add(carWashPanel);  
 datArea.setText(*data*);  
 datArea.setEditable(false);  
 jFrame.revalidate();  
 jFrame.repaint();  
 carWashPanel.setBackground(*WHITE\_COLOR*);  
 }  
 else if(*actionCommand*.equals("Submit")){  
 int *number* = Integer.*parseInt*(packageTextField.getText());  
 if(*number*>Globals.*readPackages*.size()-1){  
 System.*out*.println("no");  
 JOptionPane.*showMessageDialog*(null,"No such package exists please re-enter");  
 }  
 else{  
 for(int *i* = 0 ; *i* <Globals.*readCustomers*.size();*i*++){  
 if(Globals.*readCustomers*.get(*i*).getName().equals(Globals.*user*)){  
 Globals.*readCustomers*.get(*i*).setPrice((int) Globals.*readPackages*.get(*number*).getPrice());  
 Globals.*writeObject*(Globals.*readCustomers*);  
 try {  
 Globals.*readCustomers* = Globals.*readObject*();  
 } catch (IOException *ex*) {  
 throw new RuntimeException(*ex*);  
 }  
 JOptionPane.*showMessageDialog*(null,"Successfully Added price");  
 try {  
 String *answer* = Globals.*startWashingProcess*(Globals.*readCustomers*.get(*i*));  
 JOptionPane.*showMessageDialog*(null,*answer*);  
 } catch (IOException *ex*) {  
 throw new RuntimeException(*ex*);  
 }  
 }  
 }  
  
 }  
 } else if (*e*.getSource()==waitingButton) {  
 clearComponents();  
 waitingPanel = new JPanel();  
 waitingPanel.setBackground(*WHITE\_COLOR*);  
 waitingPanel.setBounds(customerPanel.getWidth(),0,jFrame.getWidth()-customerPanel.getWidth(),jFrame.getHeight());  
 waitingPanel.setLayout(null);  
 // waitingPanel.add(new JScrollPane(dataTextArea), BorderLayout.CENTER);  
 String *data* = "";  
 for (int *i* = 0; *i* < Globals.*readWaitingCustomers*.size(); *i*++) {  
 *data* +=Globals.*displayWaiting*(*i*) + "\n" + "\n";  
 }  
 datArea.setFont(buttonFont);  
 datArea.setBounds(customerPanel.getWidth()+50,customerPanel.getY()+50,jFrame.getWidth()-customerPanel.getWidth(),jFrame.getHeight());  
 datArea.setText(*data*);  
 // waitingPanel.add(datArea);  
 functionPanel.add(waitingPanel);  
 jFrame.add(waitingPanel);  
 jFrame.repaint();  
 jFrame.revalidate();  
 waitingPanel.setBackground(*WHITE\_COLOR*);  
  
 } else if (*e*.getSource()==goBackButton) {  
 jFrame.dispose();  
 new CustomerPanel();  
 } else if (*e*.getSource() == slotsButton) {  
 clearComponents();  
 slotsPanel = new JPanel();  
 slotsPanel.setBackground(*WHITE\_COLOR*);  
 slotsPanel.setBounds(customerPanel.getWidth(),0,jFrame.getWidth()-customerPanel.getWidth(),jFrame.getHeight());  
  
 functionPanel.add(slotsPanel);  
 String *data* ="";  
 for(int *i* = 0 ; *i*<Globals.*slotsFree*.length;*i*++){  
 *data* += Globals.*availableSlots*(*i*)+" \n--------------------------------------\n";  
 }  
 datArea.setText(*data*);  
  
 } else if (*e*.getSource()==billButton) {  
 JFrame *billFrame* = new JFrame("Bill");  
 *billFrame*.setSize(500, 400); // Set the size of the frame  
 *billFrame*.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 String *ans* ="";  
  
 JPanel *billPanel* = new JPanel();  
 *billPanel*.setLayout(null);  
  
 String *data* = "";  
 for(int *i* = 0 ; *i* <Globals.*readCustomers*.size();*i*++){  
  
 System.*out*.println(Globals.*user*);  
 if(Globals.*readCustomers*.get(*i*).getName().equals(Globals.*user*)){  
  
 if(Globals.*readCustomers*.get(*i*).isAMember()){  
 JOptionPane.*showMessageDialog*(null,"Congrats you are a member of loyalty program.");  
 }else{  
 JLabel *memberLabel* = new JLabel("Do you want to become a member ? yes/no ");  
 *memberLabel*.setFont(buttonFont);  
 *memberLabel*.setBounds(functionPanel.getX()+30, 30, 200, 30);  
 *memberLabel*.setForeground(Color.*BLACK*);  
 *billPanel*.add(*memberLabel*);  
  
 memberTextField = new JTextField();  
 memberTextField.setBounds(functionPanel.getX() + 30, *billPanel*.getY() + 50 + 30, 200, 30);  
 memberTextField.addActionListener(this);  
 *billPanel*.add(memberTextField);  
  
 addButton = new JButton("Add");  
  
 addButton.setBounds(memberTextField.getWidth()+70, memberTextField.getY(), 100, 30);  
 addButton.addActionListener(this);  
 addButton.setBackground(*SKY\_BLUE\_COLOR*);  
 addButton.setForeground(Color.*BLACK*);  
 *billPanel*.add(addButton);  
  
 }  
  
 for (int *j* = 0; *j* < Globals.*readCustomers*.size(); *j*++) {  
 if (Globals.*readCustomers*.get(*j*).getName().equals(Globals.*user*)) {  
 Globals.*readCustomers*.get(*j*).setCurrentState("Bill paid");  
 *ans* += "Bill for : "+Globals.*readCustomers*.get(*j*).getLicensePlate()+" is Rs:"+Globals.*readCustomers*.get(*j*).getPrice();  
 int *slotNumber* = Globals.*readCustomers*.get(*j*).getSlotNumber();  
 if (*slotNumber* >= 0 && *slotNumber* < Globals.*slotsFree*.length) {  
 Globals.*slotsFree*[*slotNumber*] = true;  
 }  
 Globals.*readCustomers*.get(*j*).setSlotNumber(-1);  
 Globals.*writeObject*(Globals.*readCustomers*);  
 try {  
 Globals.*readCustomers* = Globals.*readObject*();  
 } catch (IOException *ex*) {  
 throw new RuntimeException(*ex*);  
 }  
 break;  
 }  
 }  
 if(Globals.*readWaitingCustomers*.isEmpty()){  
 JOptionPane.*showMessageDialog*(null,"No customer in waiting list");  
 }  
 else{  
  
 try {  
 String *s* = Globals.*startWashingProcess*(Globals.*readWaitingCustomers*.get(0));  
 JOptionPane.*showMessageDialog*(null,*s*);  
 } catch (IOException *ex*) {  
 throw new RuntimeException(*ex*);  
 }  
 Globals.*readWaitingCustomers*.remove(0);  
 Globals.*writeWaiting*(Globals.*readWaitingCustomers*);  
 try {  
 Globals.*readWaitingCustomers* = Globals.*readWait*();  
 } catch (IOException *ex*) {  
 throw new RuntimeException(*ex*);  
 }  
 }  
 }  
 }  
 JLabel *billLabel* = new JLabel(*ans*);  
 *billLabel*.setBounds(50,170,300,70);  
 *billLabel*.setFont(buttonFont);  
 *billPanel*.add(*billLabel*);  
  
  
 *billFrame*.add(*billPanel*); // Add the billPanel to the billFrame  
  
 *billFrame*.setVisible(true);  
 } else if (*e*.getSource()==addButton) {  
 if(*e*.getActionCommand().equals("yes")){  
 for(int *i* = 0 ; *i* <Globals.*readCustomers*.size() ; *i*++){  
 if(Globals.*readCustomers*.get(*i*).getName().equals(Globals.*user*)){  
 Globals.*readCustomers*.get(*i*).setAMember(true);  
 try {  
 Globals.*writeObject*(Globals.*readObject*());  
 } catch (IOException *ex*) {  
 throw new RuntimeException(*ex*);  
 }  
 try {  
 Globals.*readCustomers* = Globals.*readObject*();  
 JOptionPane.*showMessageDialog*(null,"You will get a discount next time");  
 } catch (IOException *ex*) {  
 throw new RuntimeException(*ex*);  
 }  
 }  
 }  
 }  
 else JOptionPane.*showMessageDialog*(null,"Thank you for your opinion");  
  
 }  
 }  
  
 public void clearComponents(){  
 if(carWashPanel!=null){  
 jFrame.remove(carWashPanel);  
 carWashPanel = null ;  
 }  
 if(waitingPanel!=null){  
 jFrame.remove(waitingPanel);  
 waitingPanel = null ;  
 }  
 if(slotsPanel!=null){  
 jFrame.remove(slotsPanel);  
 slotsPanel = null;  
 }  
 if(billPanel!=null){  
 jFrame.remove(billPanel);  
 billPanel=null;  
 }  
 }  
}  
  
class CustomerPanel implements ActionListener{  
 boolean name = false;  
 String userName = "";  
 String userPassword = null;  
 JFrame jFrame;  
 JPanel sloganPanel, whitePanel, loginPanel, blankPanel;  
 JLabel logoLabel, sloganLabel, loginLabel, userLabel, passwordLabel;  
 JButton logInButton, backButton;  
 JTextField adminName, adminPassword;  
 private Font font = new Font("Consolas", Font.*BOLD*, 50);  
 private Font buttonFont = new Font("Consolas", Font.*BOLD*, 18);  
 private static final Color *SKY\_BLUE\_COLOR* = new Color(140, 210, 240).darker();  
 private static final Color *WHITE\_COLOR* = Color.*WHITE*;  
 CustomerPanel(String *name*){  
  
 }  
 CustomerPanel() {  
  
 jFrame = new JFrame("Client");  
 jFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
 jFrame.setSize(800, 600);  
 jFrame.setLayout(new BorderLayout());  
  
 sloganPanel = new JPanel();  
 sloganPanel.setLayout(new GridLayout(4, 1));  
 sloganPanel.setBackground(*SKY\_BLUE\_COLOR*);  
 sloganPanel.setPreferredSize(new Dimension(jFrame.getWidth() / 3, jFrame.getHeight()));  
  
 blankPanel = new JPanel();  
 blankPanel.setBackground(*SKY\_BLUE\_COLOR*);  
 sloganPanel.add(blankPanel);  
  
 ImageIcon *sloganIcon* = new ImageIcon("logo.png");  
 logoLabel = new JLabel(*sloganIcon*);  
 sloganPanel.add(logoLabel);  
  
 sloganLabel = new JLabel("CleanMyCar");  
 sloganLabel.setFont(buttonFont);  
 sloganLabel.setForeground(Color.*WHITE*);  
 sloganLabel.setHorizontalAlignment(SwingConstants.*CENTER*);  
 sloganPanel.add(sloganLabel);  
  
 whitePanel = new JPanel();  
 whitePanel.setBackground(*WHITE\_COLOR*);  
 whitePanel.setLayout(null);  
  
 loginLabel = new JLabel("LogIn");  
 loginLabel.setFont(buttonFont);  
 loginLabel.setForeground(Color.*BLACK*);  
 loginLabel.setBounds(200, 140, 100, 30);  
 whitePanel.add(loginLabel);  
  
 userLabel = new JLabel("Username:");  
 userLabel.setFont(buttonFont);  
 userLabel.setBounds(30, 200, 120, 30);  
 whitePanel.add(userLabel);  
  
 adminName = new JTextField();  
 adminName.setBounds(150, 200, 200, 30);  
 adminName.addActionListener(this);  
 whitePanel.add(adminName);  
  
 passwordLabel = new JLabel("Password:");  
 passwordLabel.setFont(buttonFont);  
 passwordLabel.setBounds(30, 260, 120, 30);  
 whitePanel.add(passwordLabel);  
  
 adminPassword = new JTextField();  
 adminPassword.setBounds(150, 260, 200, 30);  
 adminPassword.addActionListener(this);  
 whitePanel.add(adminPassword);  
  
 logInButton = new JButton("LogIn");  
 logInButton.setBackground(*SKY\_BLUE\_COLOR*);  
 logInButton.setForeground(Color.*WHITE*);  
 logInButton.setFont(buttonFont);  
 logInButton.setBounds(120, 320, 100, 40);  
 logInButton.addActionListener(this);  
 whitePanel.add(logInButton);  
  
 backButton = new JButton("SignUp");  
 backButton.setBackground(*SKY\_BLUE\_COLOR*);  
 backButton.setForeground(Color.*WHITE*);  
 backButton.setFont(buttonFont);  
 backButton.setBounds(250, 320, 100, 40);  
 backButton.addActionListener(this);  
  
 whitePanel.add(backButton);  
  
 jFrame.add(sloganPanel, BorderLayout.*WEST*);  
 jFrame.add(whitePanel, BorderLayout.*CENTER*);  
  
 jFrame.setVisible(true);  
 }  
  
  
 @Override  
 public void actionPerformed(ActionEvent *e*) {  
 String *actionCommand* = *e*.getActionCommand();  
 if(*e*.getSource()==adminName){  
 userName = adminName.getText();  
 Globals.*user* = userName ;  
 name = true ;  
 }else if(*e*.getSource()==adminPassword){  
 userPassword = adminPassword.getText();  
 } else if (*actionCommand*.equals("LogIn")) {  
 if(Globals.*logIn*(userName, userPassword) == null) {  
 JOptionPane.*showMessageDialog*(null,"You are not registered kindly signup");  
 }  
 else{  
 JOptionPane.*showMessageDialog*(null,"Will now given two options ");  
 new CustomerWindow();  
 }  
 } else if (*actionCommand*.equals("SignUp")) {  
 jFrame.dispose();  
 new SignUp();  
 }  
 }  
}  
class SignUp implements ActionListener{  
 String userName , userPassword , carName , licensePlate ;  
 JFrame jFrame ;  
 JLabel singUpLabel , usernameLabel , passwordLabel , carNameLabel , licensePlateLabel , logoLabel , sloganLabel;  
 JPanel signUpPanel , buttonPanel , iconPanel , blankPanel;  
 JTextField usernameField , passwordField , carNameField , licensePlateField;  
  
 JButton signUpButton ;  
 private Font font = new Font("Consolas", Font.*BOLD*, 28);  
 private Font buttonFont = new Font("Consolas", Font.*BOLD*, 18);  
 private static final Color *SKY\_BLUE\_COLOR* = new Color(135, 206, 235);  
 private static final Color *WHITE\_COLOR* = Color.*WHITE*;  
 SignUp(String *n* ){  
  
 }  
 SignUp() {  
 jFrame = new JFrame("Signup Panel");  
 jFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
 jFrame.setSize(800, 600);  
 jFrame.setLayout(new BorderLayout());  
  
 // Create icon panel (25% width)  
 iconPanel = new JPanel();  
 iconPanel.setLayout(new GridLayout(4, 1));  
 iconPanel.setBackground(*SKY\_BLUE\_COLOR*);  
 iconPanel.setPreferredSize(new Dimension(jFrame.getWidth() / 3, jFrame.getHeight()));  
  
 blankPanel = new JPanel();  
 blankPanel.setBackground(*SKY\_BLUE\_COLOR*);  
 iconPanel.add(blankPanel);  
  
 ImageIcon *sloganIcon* = new ImageIcon("logo.png");  
 logoLabel = new JLabel(*sloganIcon*);  
 iconPanel.add(logoLabel);  
  
 sloganLabel = new JLabel("CleanMyCar");  
 sloganLabel.setFont(buttonFont);  
 sloganLabel.setForeground(Color.*WHITE*);  
 sloganLabel.setHorizontalAlignment(SwingConstants.*CENTER*);  
 iconPanel.add(sloganLabel);  
  
 // Create signup panel (75% width)  
 signUpPanel = new JPanel();  
 signUpPanel.setBackground(*WHITE\_COLOR*);  
 signUpPanel.setLayout(null);  
  
 JLabel *signUp* = new JLabel("SignUp");  
 *signUp*.setForeground(Color.*BLACK*);  
 *signUp*.setBackground(*SKY\_BLUE\_COLOR*);  
 *signUp*.setFont(font);  
 *signUp*.setBounds(170,120,150,30);  
 signUpPanel.add(*signUp*);  
  
 usernameLabel = new JLabel("Username:");  
 usernameLabel.setFont(buttonFont);  
 usernameLabel.setBounds(30, 150+30, 150, 30 );  
 signUpPanel.add(usernameLabel);  
  
 usernameField = new JTextField();  
 usernameField.setBounds(150, 150+30, 150, 30);  
 usernameField.addActionListener(this);  
 signUpPanel.add(usernameField);  
  
  
  
 passwordLabel = new JLabel("Password:");  
 passwordLabel.setBounds(30, 200+30, 150, 30);  
 passwordLabel.setFont(buttonFont);  
 signUpPanel.add(passwordLabel);  
  
 passwordField = new JPasswordField();  
 passwordField.setBounds(150, 200+30, 150, 30);  
 passwordField.addActionListener(this);  
 signUpPanel.add(passwordField);  
  
  
  
 carNameLabel = new JLabel("Car Name:");  
 carNameLabel.setBounds(30, 250+30, 150, 30);  
 carNameLabel.setFont(buttonFont);  
 signUpPanel.add(carNameLabel);  
  
 carNameField = new JTextField();  
 carNameField.setBounds(150, 250+30, 150, 30);  
 carNameField.addActionListener(this);  
 signUpPanel.add(carNameField);  
  
  
 licensePlateLabel = new JLabel("License No:");  
 licensePlateLabel.setBounds(30, 300+30, 150, 30);  
 licensePlateLabel.setFont(buttonFont);  
 signUpPanel.add(licensePlateLabel);  
  
 licensePlateField = new JTextField();  
 licensePlateField.setPreferredSize(new Dimension(200, 25));  
 licensePlateField.setBounds(150, 300+30, 150, 30);  
 licensePlateField.addActionListener(this);  
 signUpPanel.add(licensePlateField);  
  
  
 signUpButton = new JButton("Submit");  
 signUpButton.setBounds(160, 380, 130, 40);  
 signUpButton.setBackground(*SKY\_BLUE\_COLOR*);  
 signUpButton.setForeground(Color.*WHITE*);  
 signUpButton.setFont(buttonFont);  
 // signUpButton.setPreferredSize(new Dimension(100, 30));  
 signUpButton.addActionListener(this);  
  
  
 signUpPanel.add(signUpButton);  
  
 jFrame.add(iconPanel, BorderLayout.*WEST*);  
 jFrame.add(signUpPanel, BorderLayout.*CENTER*);  
 jFrame.setVisible(true);  
 }  
  
  
 @Override  
 public void actionPerformed(ActionEvent *e*) {  
 String *actionCommand* = *e*.getActionCommand();  
 if(*e*.getSource()==usernameField){  
 userName = usernameField.getText();  
  
 } else if (*e*.getSource()==passwordField) {  
 userPassword = passwordField.getText();  
 } else if (*e*.getSource()==carNameField) {  
 carName = carNameField.getText();  
 } else if (*e*.getSource()==licensePlateField) {  
 licensePlate = licensePlateField.getText();  
 } else{  
 try {  
 System.*out*.println("Name is : "+userName);  
 Globals.*singUp*(userName,userPassword,carName,licensePlate,2);  
 Globals.*readCustomers* = Globals.*readObject*();  
 JOptionPane.*showMessageDialog*(null,"signedup successfully");  
 jFrame.dispose();  
 new CustomerPanel();  
 } catch (IOException *ex*) {  
 throw new RuntimeException(*ex*);  
 }  
 }  
 }  
}  
  
  
 class AdminWindow implements ActionListener {  
 private JFrame jFrame;  
 private JButton deleteButton, searchButton, displayButton, displayPackageButton, addPackageButton, deletePackageButton, changePricePackageButton, displayWaitingCustomers;  
 private JTextField textField , userNameField , userPasswordField , packageNameField , packageDescriptionField , packagePriceField , priceField , packageTimeField;  
 private JTextArea dataTextArea;  
 private JPanel adminPanel, displayCustomersPanel, searchCustomerPanel, deleteCustomerPanel, displayWaitingCustomersPanel, displayPackagesPanel, addPackagesPanel, deletePackagesPanel, changePricePackagesPanel;  
 private Font font = new Font("Consolas", Font.*BOLD*, 28);  
 private Font buttonFont = new Font("Consolas", Font.*BOLD*, 14);  
 private static final Color *SKY\_BLUE\_COLOR* = new Color(140, 210, 240);  
 private static final Color *WHITE\_COLOR* = Color.*WHITE*;  
  
 public AdminWindow() {  
 jFrame = new JFrame("Admin Window");  
 jFrame.setSize(800, 600);  
 jFrame.setLayout(null);  
  
 adminPanel = new JPanel(null); // Set layout manager to null  
 adminPanel.setBounds(0, 0, jFrame.getWidth() / 3, jFrame.getHeight());  
 adminPanel.setBackground(*SKY\_BLUE\_COLOR*.darker());  
  
 // Add admin icon and label  
 ImageIcon *adminIcon* = new ImageIcon("User.png"); // Replace with the path to your admin icon image  
 // Image scaledIcon = adminIcon.getImage().getScaledInstance(50, 50, Image.SCALE\_SMOOTH);  
  
  
 JLabel *adminLabel* = new JLabel("Admin", JLabel.*CENTER*);  
 *adminLabel*.setIcon(*adminIcon*);  
 *adminLabel*.setFont(new Font("Arial", Font.*BOLD*, 24));  
 *adminLabel*.setForeground(Color.*WHITE*);  
 *adminLabel*.setBounds(10, 10, adminPanel.getWidth() - 40, 70);  
 adminPanel.add(*adminLabel*);  
  
 // Add display button  
 displayButton = new JButton("Display Customers");  
 displayButton.setBackground(Color.*WHITE*.brighter());  
 displayButton.setForeground(Color.*BLACK*);  
 displayButton.addActionListener(this);  
 displayButton.setBounds(50, 100, adminPanel.getWidth() - 120, 30);  
 adminPanel.add(displayButton);  
  
 // Add search button  
 searchButton = new JButton("Search Customer");  
 searchButton.setBackground(Color.*WHITE*.brighter());  
 searchButton.setForeground(Color.*BLACK*);  
 searchButton.addActionListener(this);  
 searchButton.setBounds(50, 150, adminPanel.getWidth() - 120, 30);  
 adminPanel.add(searchButton);  
  
 // Add delete button  
 deleteButton = new JButton("Delete Customer");  
 deleteButton.setBackground(Color.*WHITE*.brighter());  
 deleteButton.setForeground(Color.*BLACK*);  
 deleteButton.addActionListener(this);  
 deleteButton.setBounds(50, 200, adminPanel.getWidth() - 120, 30);  
 adminPanel.add(deleteButton);  
  
 displayWaitingCustomers = new JButton("Waiting Customer");  
 displayWaitingCustomers.setBackground(Color.*WHITE*.brighter());  
 displayWaitingCustomers.setForeground(Color.*BLACK*);  
 displayWaitingCustomers.addActionListener(this);  
 displayWaitingCustomers.setBounds(50, 250, adminPanel.getWidth() - 120, 30);  
 adminPanel.add(displayWaitingCustomers);  
  
 displayPackageButton = new JButton("Display Packages");  
 displayPackageButton.setBackground(Color.*WHITE*.brighter());  
 displayPackageButton.setForeground(Color.*BLACK*);  
 displayPackageButton.addActionListener(this);  
 displayPackageButton.setBounds(50, 300, adminPanel.getWidth() - 120, 30);  
 adminPanel.add(displayPackageButton);  
  
 addPackageButton = new JButton("Add Packages");  
 addPackageButton.setBackground(Color.*WHITE*.brighter());  
 addPackageButton.setForeground(Color.*BLACK*);  
 addPackageButton.addActionListener(this);  
 addPackageButton.setBounds(50, 350, adminPanel.getWidth() - 120, 30);  
 adminPanel.add(addPackageButton);  
  
 deletePackageButton = new JButton("Delete Packages");  
 deletePackageButton.setBackground(Color.*WHITE*.brighter());  
 deletePackageButton.addActionListener(this);  
 deletePackageButton.setForeground(Color.*BLACK*);  
 deletePackageButton.setBounds(50, 400, adminPanel.getWidth() - 120, 30);  
 adminPanel.add(deletePackageButton);  
  
 changePricePackageButton = new JButton("Change Price Packages");  
 changePricePackageButton.setBackground(Color.*WHITE*.brighter());  
 changePricePackageButton.addActionListener(this);  
 changePricePackageButton.setForeground(Color.*BLACK*);  
 changePricePackageButton.setBounds(50, 450, adminPanel.getWidth() - 120, 30);  
 adminPanel.add(changePricePackageButton);  
  
 dataTextArea = new JTextArea();  
 dataTextArea.setBackground(*WHITE\_COLOR*);  
 dataTextArea.setBounds(adminPanel.getWidth(),0,jFrame.getWidth()-adminPanel.getWidth(),jFrame.getHeight());  
 dataTextArea.setEditable(false);  
  
 dataTextArea.setFont(buttonFont);  
 dataTextArea.setText("Your Text goes here");  
 jFrame.add(dataTextArea);  
  
 jFrame.add(adminPanel);  
  
 jFrame.setVisible(true);  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent *e*) {  
  
  
 if (*e*.getSource() == displayButton) {  
 clearComponents();  
 if (displayCustomersPanel == null) {  
 displayCustomersPanel = new JPanel();  
 displayCustomersPanel.setBackground(*WHITE\_COLOR*);  
 displayCustomersPanel.setBounds(adminPanel.getWidth(), 0, jFrame.getWidth() - adminPanel.getWidth() - 20, jFrame.getHeight() - 20);  
 displayCustomersPanel.setLayout(new BorderLayout());  
 displayCustomersPanel.add(new JScrollPane(dataTextArea), BorderLayout.*CENTER*);  
 dataTextArea = new JTextArea();  
 dataTextArea.setEditable(false);  
 dataTextArea.setBackground(*WHITE\_COLOR*);  
  
 dataTextArea.setBounds(displayCustomersPanel.getX()+30 , displayCustomersPanel.getY()+50+30+50, jFrame.getWidth() - adminPanel.getWidth() - 20, jFrame.getHeight() - 20);  
 displayCustomersPanel.add(dataTextArea);  
 String *data* = "";  
 for (int *i* = 0 ,*p* =1; *i* < Globals.*readCustomers*.size(); *i*++ ,*p*++) {  
 *data*+="Customer "+*p*+" Details : "+"\nName : "+Globals.*readCustomers*.get(*i*).getName()+" \nCar Name : "+Globals.*readCustomers*.get(*i*).getCarName()  
 +" \nLicense Plate : "+Globals.*readCustomers*.get(*i*).getLicensePlate()+" \nDate Signed : "+Globals.*readCustomers*.get(*i*).getDate()+" \n-" +  
 "---------------------------------------------------\n";  
 }  
 dataTextArea.setFont(buttonFont);  
 dataTextArea.setBackground(Color.*white*);  
 dataTextArea.setText(*data*);  
  
 }  
 jFrame.add(displayCustomersPanel);  
 jFrame.repaint();  
 jFrame.revalidate();  
 displayCustomersPanel.setBackground(*WHITE\_COLOR*);  
 } else if (*e*.getSource() == searchButton) {  
 clearComponents();  
 if (searchCustomerPanel == null) {  
 searchCustomerPanel = new JPanel();  
  
 }  
 searchCustomerPanel.setBackground(*WHITE\_COLOR*);  
 searchCustomerPanel.setBounds(adminPanel.getWidth() + 30, 50, jFrame.getWidth() - adminPanel.getWidth() - 20, jFrame.getHeight() - 20);  
 searchCustomerPanel.setLayout(null);  
  
 JLabel *searchLabel* = new JLabel("Enter Customer Number:");  
 *searchLabel*.setBounds(10, 10, 200, 30);  
 searchCustomerPanel.add(*searchLabel*);  
  
 textField = new JTextField();  
 textField.setBounds(10, 50, 200, 30);  
 searchCustomerPanel.add(textField);  
  
 JButton *searchSubmitButton* = new JButton("Search");  
 *searchSubmitButton*.setBounds(10, 90, 200, 30);  
 *searchSubmitButton*.addActionListener(this);  
 searchCustomerPanel.add(*searchSubmitButton*);  
  
 // dataTextArea.setBounds(10, 200, searchCustomerPanel.getWidth() - 20, searchCustomerPanel.getHeight() - 140);  
 //searchCustomerPanel.add(dataTextArea);  
  
 jFrame.add(searchCustomerPanel);  
 jFrame.add(searchCustomerPanel);  
 jFrame.repaint();  
 jFrame.revalidate();  
 searchCustomerPanel.setBackground(*WHITE\_COLOR*);  
 } else if (*e*.getActionCommand().equals("Search")) {  
 String *searchName* = textField.getText();  
 String *c* = Globals.*searchCustomer*(*searchName*);  
 if(*c*.equals("Not Found")) dataTextArea.setText("Customer Not Found");  
 else {  
  
 dataTextArea.setText(*c*);  
 }  
  
 } else if (*e*.getSource() == deleteButton) {  
 clearComponents();  
 if (deleteCustomerPanel == null) {  
 deleteCustomerPanel = new JPanel();  
  
 }  
 deleteCustomerPanel = new JPanel();  
 deleteCustomerPanel.setBackground(*WHITE\_COLOR*);  
 deleteCustomerPanel.setBounds(adminPanel.getWidth() + 10, 10, jFrame.getWidth() - adminPanel.getWidth() - 20, jFrame.getHeight() - 20);  
 deleteCustomerPanel.setLayout(null);  
  
  
 JLabel *deleteLabel* = new JLabel("Delete Customer Name ");  
 *deleteLabel*.setBounds(10, 10, 200, 30);  
 deleteCustomerPanel.add(*deleteLabel*);  
  
 userNameField = new JTextField();  
 userNameField.setBounds(10, 50, 200, 30);  
 deleteCustomerPanel.add(userNameField);  
  
 JLabel *deletePasswordLabel* = new JLabel("Delete Customer Password ");  
 *deletePasswordLabel*.setBounds(10, 100, 200, 30);  
 deleteCustomerPanel.add(*deletePasswordLabel*);  
  
 userPasswordField = new JTextField();  
 userPasswordField.setBounds(10, 150, 200, 30);  
 deleteCustomerPanel.add(userPasswordField);  
  
 JButton *deleteSubmitButton* = new JButton("Delete");  
 *deleteSubmitButton*.setBounds(10, 230, 200, 30);  
 *deleteSubmitButton*.addActionListener(this);  
 deleteCustomerPanel.add(*deleteSubmitButton*);  
 dataTextArea.setFont(buttonFont);  
 dataTextArea.setBounds(10, 280, deleteCustomerPanel.getWidth() - 20, deleteCustomerPanel.getHeight() - 140);  
 deleteCustomerPanel.add(dataTextArea);  
  
  
 jFrame.add(deleteCustomerPanel);  
  
 jFrame.add(deleteCustomerPanel);  
 jFrame.add(deleteCustomerPanel);  
 jFrame.repaint();  
 jFrame.revalidate();  
 deleteCustomerPanel.setBackground(*WHITE\_COLOR*);  
 }else if(*e*.getActionCommand().equals("Delete")) {  
  
 try {  
 String *c* = Globals.*deleteCustomer*(userNameField.getText(),userPasswordField.getText());  
 if(*c*.equals("null")) dataTextArea.setText(userNameField.getText()+" Not Found");  
 else dataTextArea.setText(*c*+" Found");  
 } catch (IOException *ex*) {  
 throw new RuntimeException(*ex*);  
 }  
 }  
 else if (*e*.getSource() == displayPackageButton) {  
 clearComponents();  
 if (displayPackagesPanel == null) {  
 displayPackagesPanel = new JPanel();  
 }  
 displayPackagesPanel.setBackground(*WHITE\_COLOR*);  
 displayPackagesPanel.setBounds(adminPanel.getWidth() + 10, 10, jFrame.getWidth() - adminPanel.getWidth() - 20, jFrame.getHeight() - 20);  
 displayPackagesPanel.setLayout(new BorderLayout());  
 displayPackagesPanel.add(new JScrollPane(dataTextArea), BorderLayout.*CENTER*);  
 String *data* = "";  
 for (int *i* = 0 , *p*= 1; *i* < Globals.*readPackages*.size(); *i*++,*p*++) {  
 *data* += "Package "+*p*+" Details : \n"+"Name : "+Globals.*readPackages*.get(*i*).getName()+" \nDescription : "  
 +Globals.*readPackages*.get(*i*).getDescription()+" \nPrice : "+Globals.*readPackages*.get(*i*).getPrice()+" \n" +  
 "------------------------------------------------\n";  
 }  
 dataTextArea.setFont(buttonFont);  
 dataTextArea.setText(*data*);  
 displayPackagesPanel.add(dataTextArea);  
  
  
 jFrame.add(displayPackagesPanel);  
 jFrame.repaint();  
 jFrame.revalidate();  
 displayPackagesPanel.setBackground(*WHITE\_COLOR*);  
 } else if (*e*.getSource() == deletePackageButton) {  
 clearComponents();  
 if (deletePackagesPanel == null) {  
 deletePackagesPanel = new JPanel();  
  
 }  
 deletePackagesPanel = new JPanel();  
 deletePackagesPanel.setBackground(*WHITE\_COLOR*);  
 deletePackagesPanel.setBounds(adminPanel.getWidth() + 10, 10, jFrame.getWidth() - adminPanel.getWidth() - 20, jFrame.getHeight() - 20);  
 deletePackagesPanel.setLayout(null);  
  
  
 JLabel *deleteLabel* = new JLabel("Delete Package Name ");  
 *deleteLabel*.setBounds(10, 10, 200, 30);  
 deletePackagesPanel.add(*deleteLabel*);  
  
 packageNameField = new JTextField();  
 packageNameField.setBounds(10, 50, 200, 30);  
 deletePackagesPanel.add(packageNameField);  
  
  
 JButton *deleteSubmitButton* = new JButton("Delete Package");  
 *deleteSubmitButton*.setBounds(10, 100, 200, 30);  
 *deleteSubmitButton*.addActionListener(this);  
 deletePackagesPanel.add(*deleteSubmitButton*);  
  
 dataTextArea.setBounds(10, 280, deletePackagesPanel.getWidth() - 20, deletePackagesPanel.getHeight() - 140);  
 deletePackagesPanel.add(dataTextArea);  
  
 jFrame.add(deletePackagesPanel);  
  
 jFrame.repaint();  
 jFrame.revalidate();  
 deletePackagesPanel.setBackground(*WHITE\_COLOR*);  
 } else if (*e*.getActionCommand().equals("Delete Package")) {  
 try {  
 String *c* = Globals.*deletePackage*(packageNameField.getText());  
 if(*c*.equals("null")) dataTextArea.setText(packageNameField.getText()+" Not Found");  
 else dataTextArea.setText(*c*+" Found");  
 } catch (IOException *ex*) {  
 throw new RuntimeException(*ex*);  
 }  
  
 } else if (*e*.getSource() == displayWaitingCustomers) {  
 clearComponents();  
 if (displayWaitingCustomersPanel == null) {  
 displayWaitingCustomersPanel = new JPanel();  
 }  
 dataTextArea.setFont(buttonFont);  
 displayWaitingCustomersPanel.setBackground(*WHITE\_COLOR*);  
 displayWaitingCustomersPanel.setBounds(adminPanel.getWidth() + 10, 10, jFrame.getWidth() - adminPanel.getWidth() - 20, jFrame.getHeight() - 20);  
 displayWaitingCustomersPanel.setLayout(new BorderLayout());  
 displayWaitingCustomersPanel.add(new JScrollPane(dataTextArea), BorderLayout.*CENTER*);  
 String *data* = "";  
 for (int *i* = 0; *i* < Globals.*readWaitingCustomers*.size(); *i*++) {  
 *data* += Globals.*displayWaiting*(*i*) + "\n" + "\n";  
 }  
 dataTextArea.setText(*data*);  
 displayWaitingCustomersPanel.add(dataTextArea);  
  
 jFrame.add(displayWaitingCustomersPanel);  
  
 jFrame.repaint();  
 jFrame.revalidate();  
 displayWaitingCustomersPanel.setBackground(*WHITE\_COLOR*);  
 } else if (*e*.getSource() == addPackageButton) {  
 clearComponents();  
 if (addPackagesPanel == null) {  
 addPackagesPanel = new JPanel();  
  
 }  
 addPackagesPanel = new JPanel();  
 addPackagesPanel.setBackground(*WHITE\_COLOR*);  
 addPackagesPanel.setBounds(adminPanel.getWidth() + 10, 10, jFrame.getWidth() - adminPanel.getWidth() - 20, jFrame.getHeight() - 20);  
 addPackagesPanel.setLayout(null);  
  
  
 JLabel *packageNameLabel* = new JLabel("Package Name ");  
 *packageNameLabel*.setBounds(10, 10, 200, 30);  
 addPackagesPanel.add(*packageNameLabel*);  
  
 packageNameField = new JTextField();  
 packageNameField.setBounds(10, 50, 200, 30);  
 addPackagesPanel.add(packageNameField);  
  
 JLabel *packageDescriptionLabel* = new JLabel("Package Description ");  
 *packageDescriptionLabel*.setBounds(10, 100, 200, 30);  
 addPackagesPanel.add(*packageDescriptionLabel*);  
  
 packageDescriptionField = new JTextField();  
 packageDescriptionField.setBounds(10, 150, 200, 30);  
 addPackagesPanel.add(packageDescriptionField);  
  
 JLabel *packagePriceLabel* = new JLabel("Package Price ");  
 *packagePriceLabel*.setBounds(10, 200, 200, 30);  
 addPackagesPanel.add(*packagePriceLabel*);  
  
 packagePriceField = new JTextField();  
 packagePriceField.setBounds(10, 250, 200, 30);  
 addPackagesPanel.add(packagePriceField);  
  
 JLabel *packageTimeLabel* = new JLabel("Package Time ");  
 *packageTimeLabel*.setBounds(10, 300, 200, 30);  
 addPackagesPanel.add(*packageTimeLabel*);  
  
 packageTimeField = new JTextField();  
 packageTimeField.setBounds(10, 350, 200, 30);  
 addPackagesPanel.add(packageTimeField);  
  
 JButton *addPackage* = new JButton("Add Package");  
 *addPackage*.setBounds(10, 400, 200, 30);  
 *addPackage*.addActionListener(this);  
 addPackagesPanel.add(*addPackage*);  
 dataTextArea.setFont(buttonFont);  
 dataTextArea.setBounds(10, 280, addPackagesPanel.getWidth() - 20, addPackagesPanel.getHeight() - 140);  
 addPackagesPanel.add(dataTextArea);  
  
 jFrame.add(addPackagesPanel);  
  
 jFrame.repaint();  
 jFrame.revalidate();  
 displayWaitingCustomersPanel.setBackground(*WHITE\_COLOR*);  
 }else if (*e*.getActionCommand().equals("Add Package")) {  
  
 Globals.*addPackage*(packageDescriptionField.getText(),packageNameField.getText(),Double.*parseDouble*(packagePriceField.getText()),Integer.*parseInt*(packageTimeField.getText()));  
 dataTextArea.setText("Added");  
 JOptionPane.*showMessageDialog*(null,"Successfully added");  
  
  
 }  
 else if (*e*.getSource() == changePricePackageButton) {  
 clearComponents();  
 if (changePricePackagesPanel == null) {  
 changePricePackagesPanel = new JPanel();  
  
 }  
 changePricePackagesPanel = new JPanel();  
 changePricePackagesPanel.setBackground(*WHITE\_COLOR*);  
 changePricePackagesPanel.setBounds(adminPanel.getWidth() + 10, 10, jFrame.getWidth() - adminPanel.getWidth() - 20, jFrame.getHeight() - 20);  
 changePricePackagesPanel.setLayout(null);  
  
  
 JLabel *packageNameLabel* = new JLabel("New Price ");  
 *packageNameLabel*.setBounds(10, 10, 200, 30);  
 changePricePackagesPanel.add(*packageNameLabel*);  
  
 packageNameField = new JTextField();  
 packageNameField.setBounds(10, 50, 200, 30);  
 changePricePackagesPanel.add(packageNameField);  
  
 JLabel *priceLabel* = new JLabel("New Price ");  
 *priceLabel*.setBounds(10, 100, 200, 30);  
 changePricePackagesPanel.add(*priceLabel*);  
  
 priceField = new JTextField();  
 priceField.setBounds(10, 150, 200, 30);  
 changePricePackagesPanel.add(priceField);  
  
  
 JButton *SubmitButton* = new JButton("Change Price");  
 *SubmitButton*.setBounds(10, 200, 200, 30);  
 *SubmitButton*.addActionListener(this);  
 changePricePackagesPanel.add(*SubmitButton*);  
 dataTextArea.setFont(buttonFont);  
 dataTextArea.setBounds(10, 280, changePricePackagesPanel.getWidth() - 20, changePricePackagesPanel.getHeight() - 140);  
 changePricePackagesPanel.add(dataTextArea);  
  
 jFrame.add(changePricePackagesPanel);  
 } else if (*e*.getActionCommand().equals("Change Price")) {  
 try {  
 Globals.*changePrice*(packageNameField.getText() ,Double.*parseDouble*( priceField.getText()));  
 dataTextArea.setFont(buttonFont);  
 dataTextArea.setText("Successfully Changed Price");  
 } catch (IOException *ex*) {  
 throw new RuntimeException(*ex*);  
 }  
 }  
 else if (*e*.getActionCommand().equals("Change Price")) {  
 try {  
 String *c* = Globals.*deletePackage*(packageNameField.getText());  
 if(*c*.equals("null")) dataTextArea.setText(packageNameField.getText()+" Not Found");  
 else dataTextArea.setText(*c*+" Found");  
 JOptionPane.*showMessageDialog*(null,"Successfully added");  
 } catch (IOException *ex*) {  
 throw new RuntimeException(*ex*);  
 }  
  
 }  
  
 jFrame.revalidate();  
 jFrame.repaint();  
 }  
  
  
  
 private void clearComponents() {  
 if (displayCustomersPanel != null) {  
 jFrame.remove(displayCustomersPanel);  
 displayCustomersPanel = null;  
 }  
 if (searchCustomerPanel != null) {  
 jFrame.remove(searchCustomerPanel);  
 searchCustomerPanel = null;  
 }  
 if (deleteCustomerPanel != null) {  
 jFrame.remove(deleteCustomerPanel);  
 deleteCustomerPanel = null;  
 }  
 if(deletePackagesPanel != null){  
 jFrame.remove(deletePackagesPanel);  
 deletePackagesPanel = null;  
 }  
 if(displayPackagesPanel != null){  
 jFrame.remove(displayPackagesPanel);  
 displayPackagesPanel = null;  
 }  
 if(addPackagesPanel != null){  
 jFrame.remove(addPackagesPanel);  
 addPackagesPanel = null;  
 }  
 if(changePricePackagesPanel!=null){  
 jFrame.remove(changePricePackagesPanel);  
 changePricePackagesPanel = null ;  
 }  
 // Clear other panels here  
 }  
}  
class AdminPanel implements ActionListener{  
 boolean name = false;  
 JFrame jFrame;  
 JPanel sloganPanel, whitePanel, loginPanel, blankPanel;  
 JLabel logoLabel, sloganLabel, loginLabel, userLabel, passwordLabel;  
 JButton logInButton, backButton;  
 JTextField adminName, adminPassword;  
 private Font font = new Font("Cooper Black", Font.*BOLD*, 50);  
 private Font buttonFont = new Font("Consolas", Font.*BOLD*, 18);  
 private static final Color *SKY\_BLUE\_COLOR* = new Color(140, 210, 240).darker();  
 private static final Color *WHITE\_COLOR* = Color.*WHITE*;  
  
 AdminPanel() {  
 jFrame = new JFrame("Admin");  
 jFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
 jFrame.setSize(800, 600);  
 jFrame.setLayout(new BorderLayout());  
  
 sloganPanel = new JPanel();  
 sloganPanel.setLayout(new GridLayout(4, 1));  
 sloganPanel.setBackground(*SKY\_BLUE\_COLOR*);  
 sloganPanel.setPreferredSize(new Dimension(jFrame.getWidth() / 3, jFrame.getHeight()));  
  
 blankPanel = new JPanel();  
 blankPanel.setBackground(*SKY\_BLUE\_COLOR*);  
 sloganPanel.add(blankPanel);  
  
 ImageIcon *sloganIcon* = new ImageIcon("logo.png");  
 logoLabel = new JLabel(*sloganIcon*);  
 sloganPanel.add(logoLabel);  
  
 sloganLabel = new JLabel("CleanMyCar");  
 sloganLabel.setFont(buttonFont);  
 sloganLabel.setForeground(Color.*WHITE*);  
 sloganLabel.setHorizontalAlignment(SwingConstants.*CENTER*);  
 sloganPanel.add(sloganLabel);  
  
 whitePanel = new JPanel();  
 whitePanel.setBackground(*WHITE\_COLOR*);  
 whitePanel.setLayout(null);  
  
 loginLabel = new JLabel("LogIn");  
 loginLabel.setFont(buttonFont);  
 loginLabel.setForeground(Color.*BLACK*);  
 loginLabel.setBounds(200, 140, 100, 30);  
 whitePanel.add(loginLabel);  
  
 userLabel = new JLabel("Username:");  
 userLabel.setFont(buttonFont);  
 userLabel.setBounds(30, 200, 120, 30);  
 whitePanel.add(userLabel);  
  
 adminName = new JTextField();  
 adminName.setBounds(150, 200, 200, 30);  
 adminName.addActionListener(this);  
 whitePanel.add(adminName);  
  
 passwordLabel = new JLabel("Password:");  
 passwordLabel.setFont(buttonFont);  
 passwordLabel.setBounds(30, 260, 120, 30);  
 whitePanel.add(passwordLabel);  
  
 adminPassword = new JTextField();  
 adminPassword.setBounds(150, 260, 200, 30);  
 adminPassword.addActionListener(this);  
 whitePanel.add(adminPassword);  
  
 logInButton = new JButton("LogIn");  
 logInButton.setBackground(*SKY\_BLUE\_COLOR*);  
 logInButton.setForeground(Color.*WHITE*);  
 logInButton.setFont(buttonFont);  
 logInButton.setBounds(120, 320, 100, 40);  
 logInButton.addActionListener(this);  
 whitePanel.add(logInButton);  
  
 backButton = new JButton("Back");  
 backButton.setBackground(*SKY\_BLUE\_COLOR*);  
 backButton.setForeground(Color.*WHITE*);  
 backButton.setFont(buttonFont);  
 backButton.setBounds(250, 320, 100, 40);  
 backButton.addActionListener(this);  
  
 whitePanel.add(backButton);  
  
 jFrame.add(sloganPanel, BorderLayout.*WEST*);  
 jFrame.add(whitePanel, BorderLayout.*CENTER*);  
  
 jFrame.setVisible(true);  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent *e*) {  
 String *actionCommand* = *e*.getActionCommand();  
 if(*e*.getSource()==adminName){  
 if(*actionCommand*.equals("a")){  
 name = true ;  
 }  
 else{  
 name = false ;  
 }  
 }else if(*e*.getSource()==adminPassword){  
 if(adminPassword.getText().equals("a")&& name){  
 name = true ;  
 }  
 else{  
 name = false ;  
 }  
 } else if (*actionCommand*.equals("LogIn")) {  
 if(name) {  
 JOptionPane.*showMessageDialog*(null,"Will be directed to new window");  
 new AdminWindow();  
 }  
 else{  
 JOptionPane.*showMessageDialog*(null,"Wrong Entry");  
 }  
 } else if (*actionCommand*.equals("Back")) {  
 jFrame.dispose();  
 new CarWashGUI();  
 }  
 }  
}  
class CarWashGUI implements ActionListener {  
 private JFrame jFrame;  
 private JLayeredPane jLayeredPane;  
 private JPanel jPanel, buttonsPanel;  
 private JLabel carWashLabel;  
 private JButton adminButton, customerButton;  
 private Font font = new Font("Cooper Black", Font.*BOLD*, 60);  
 private Font buttonFont = new Font("Consolas", Font.*BOLD*, 30);  
  
  
 CarWashGUI() {  
 jFrame = new JFrame("Car Wash");  
 jFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
 jFrame.setSize(800, 600);  
 jFrame.setLayout(null);  
  
 jLayeredPane = new JLayeredPane();  
 jLayeredPane.setBounds(0, 0, jFrame.getWidth(), jFrame.getHeight());  
  
 ImageIcon *backgroundIcon* = new ImageIcon("4a4a8f3537407909776d89f6240c1c0a.gif");  
 JLabel *gifLabel* = new JLabel(*backgroundIcon*);  
 *gifLabel*.setBounds(0, 0, jFrame.getWidth(), jFrame.getHeight());  
  
 jLayeredPane.add(*gifLabel*, JLayeredPane.*DEFAULT\_LAYER*);  
  
 buttonsPanel = new JPanel();  
 buttonsPanel.setLayout(new BoxLayout(buttonsPanel, BoxLayout.*X\_AXIS*));  
 buttonsPanel.setBounds(250, 470, 320, 50);  
 buttonsPanel.setOpaque(false);  
  
 // Creating a label to display "Car Wash" text  
 carWashLabel = new JLabel("CleanMyCar");  
 carWashLabel.setFont(font);  
 carWashLabel.setForeground(Color.*WHITE*);  
 carWashLabel.setBounds(220, 50, 600, 100);  
 // carWashLabel.setHorizontalAlignment(SwingConstants.CENTER);  
 jLayeredPane.add(carWashLabel, JLayeredPane.*PALETTE\_LAYER*);  
  
 adminButton = new JButton("Admin");  
 customerButton = new JButton("Client");  
 adminButton.addActionListener(this);  
 customerButton.addActionListener(this);  
 adminButton.setFont(buttonFont);  
 customerButton.setFont(buttonFont);  
 adminButton.setBackground(Color.*white*);  
 customerButton.setBackground(Color.*WHITE*);  
 adminButton.setForeground(Color.*BLACK*);  
 customerButton.setForeground(Color.*BLACK*);  
  
 buttonsPanel.add(Box.*createHorizontalGlue*());  
 buttonsPanel.add(adminButton);  
 buttonsPanel.add(Box.*createHorizontalStrut*(35));  
 /\*EmptyBorder buttonPadding = new EmptyBorder(0, 0, 0, 20);  
 customerButton.setBorder(buttonPadding);\*/  
 buttonsPanel.add(customerButton);  
  
 buttonsPanel.add(Box.*createHorizontalGlue*());  
  
 // Adjusting the size of buttonsPanel to fit the buttons  
 // buttonsPanel.setPreferredSize(new Dimension(buttonsPanel.getPreferredSize().width, customerButton.getPreferredSize().height));  
 // buttonsPanel.setPreferredSize(new Dimension(buttonsPanel.getPreferredSize().width, adminButton.getPreferredSize().height));  
  
 jLayeredPane.add(buttonsPanel, JLayeredPane.*POPUP\_LAYER*);  
  
 jFrame.add(jLayeredPane);  
 jFrame.setVisible(true);  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent *e*) {  
 if (*e*.getSource() == adminButton) {  
 // Perform actions for admin button  
 jFrame.dispose();  
 new AdminPanel();  
 } else if (*e*.getSource() == customerButton) {  
 // Perform actions for customer button  
 new CustomerPanel();  
 }  
 }  
  
}  
  
class Globals {  
 public static Scanner *scanner* = new Scanner(System.*in*);  
 public static String *user* = "";  
 public static final int *slots* = 5;  
 public static boolean[] *slotsFree* = new boolean[*slots*];  
  
 public static ArrayList<Customer> *readWaitingCustomers*;  
  
 static {  
 try {  
 *readWaitingCustomers* = *readWait*();  
 } catch (IOException *e*) {  
 throw new RuntimeException(*e*);  
 }  
 }  
  
 Globals() throws IOException {  
 }  
  
 public static void setSlotsFree(boolean[] *slotsFree*) {  
 Globals.*slotsFree* = *slotsFree*;  
 }  
  
  
 public static void writeWaiting(ArrayList<Customer> *object*) {  
 if (*object*.size() == 0) {  
  
 File *myObj* = new File("waitingCustomers.txt");  
 *myObj*.delete();  
 } else {  
  
  
 try (ObjectOutputStream *objectOutputStream* = new ObjectOutputStream(new FileOutputStream("waitingCustomers.txt"))) {  
 *objectOutputStream*.writeObject(*object*);  
 *objectOutputStream*.close();  
 System.*out*.println("Successfully written.");  
 } catch (IOException *e*) {  
 *e*.printStackTrace();  
 }  
 }  
 }  
  
 public static ArrayList<Customer> readWait() throws IOException {  
 ObjectInputStream *objectInputStream* = null;  
 Path *p* = Paths.*get*("waitingCustomers.txt");  
 if (!Files.*exists*(*p*)) return new ArrayList<Customer>();  
 ArrayList<Customer> *customers* = new ArrayList<>();  
 try {  
 *objectInputStream* = new ObjectInputStream(new  
 FileInputStream("waitingCustomers.txt"));  
 while (true) {  
 *customers* = (ArrayList<Customer>) *objectInputStream*.readObject();  
 if (*customers* == null) return new ArrayList<Customer>();  
 }  
  
 } catch (EOFException *ex*) { //This exception will be caught when EOF is  
 System.*out*.println("End of file reached.");  
 } catch (ClassNotFoundException *ex*) {  
 *ex*.printStackTrace();  
 } catch (FileNotFoundException *ex*) {  
 *ex*.printStackTrace();  
 } catch (IOException *ex*) {  
 *ex*.printStackTrace();  
 }  
 *objectInputStream*.close();  
 return *customers*;  
 }  
  
 //writing in file method  
 public static void writeObject(ArrayList<Customer> *object*) {  
 if (*object*.size() == 0) {  
  
 File *myObj* = new File("Customers.txt");  
 *myObj*.delete();  
 } else {  
  
  
 try (ObjectOutputStream *objectOutputStream* = new ObjectOutputStream(new FileOutputStream("Customers.txt"))) {  
 *objectOutputStream*.writeObject(*object*);  
 *objectOutputStream*.close();  
 System.*out*.println("Successfully written.");  
 } catch (IOException *e*) {  
 *e*.printStackTrace();  
 }  
 }  
 }  
  
 public static void writePackage(ArrayList<Packages> *object*) {  
 if (*object*.size() == 0) {  
  
 File *myObj* = new File("Packages.txt");  
 *myObj*.delete();  
 } else {  
  
  
 try (ObjectOutputStream *objectOutputStream* = new ObjectOutputStream(new FileOutputStream("Packages.txt"))) {  
 *objectOutputStream*.writeObject(*object*);  
 *objectOutputStream*.close();  
 System.*out*.println("Successfully written.");  
 } catch (IOException *e*) {  
 *e*.printStackTrace();  
 }  
 }  
 }  
  
 public static ArrayList<Customer> readObject() throws IOException {  
 ObjectInputStream *objectInputStream* = null;  
 Path *p* = Paths.*get*("Customers.txt");  
 if (!Files.*exists*(*p*)) return new ArrayList<Customer>();  
 ArrayList<Customer> *customers* = new ArrayList<>();  
 try {  
 *objectInputStream* = new ObjectInputStream(new  
 FileInputStream("Customers.txt"));  
 while (true) {  
 *customers* = (ArrayList<Customer>) *objectInputStream*.readObject();  
 if (*customers* == null) return new ArrayList<Customer>();  
 }  
  
 } catch (EOFException *ex*) { //This exception will be caught when EOF is  
 System.*out*.println("End of file reached.");  
 } catch (ClassNotFoundException *ex*) {  
 *ex*.printStackTrace();  
 } catch (FileNotFoundException *ex*) {  
 *ex*.printStackTrace();  
 } catch (IOException *ex*) {  
 *ex*.printStackTrace();  
 }  
 *objectInputStream*.close();  
 return *customers*;  
 }  
  
  
 public static ArrayList<Packages> readPackages() throws FileNotFoundException, IOException {  
 ObjectInputStream *objectInputStream* = null;  
 Path *p* = Paths.*get*("Packages.txt");  
 if (!Files.*exists*(*p*)) return new ArrayList<Packages>();  
 ArrayList<Packages> *packages* = null;  
 try {  
 *objectInputStream* = new ObjectInputStream(new  
 FileInputStream("Packages.txt"));  
 while (true) {  
 *packages* = (ArrayList<Packages>) *objectInputStream*.readObject();  
 }  
  
 } catch (EOFException *ex*) { //This exception will be caught when EOF is  
 System.*out*.println("End of file reached.");  
 } catch (ClassNotFoundException *ex*) {  
 *ex*.printStackTrace();  
 } catch (FileNotFoundException *ex*) {  
 *ex*.printStackTrace();  
 } catch (IOException *ex*) {  
 *ex*.printStackTrace();  
 }  
 *objectInputStream*.close();  
 return *packages*;  
 }  
  
 public static ArrayList<Customer> *readCustomers*;  
  
 static {  
 try {  
 *readCustomers* = *readObject*();  
 } catch (IOException *e*) {  
 *e*.printStackTrace();  
 }  
 }  
  
 public static ArrayList<Packages> *readPackages*;  
  
 static {  
 try {  
 *readPackages* = Globals.*readPackages*();  
 } catch (IOException *e*) {  
 *e*.printStackTrace();  
 }  
 }  
  
  
  
 public static void displayWaiting() {  
 for (Customer *c* : *readWaitingCustomers*) {  
 System.*out*.println(*c*.getName() + " " + *c*.getSlotNumber());  
 }  
 }  
  
 public static void singUp(String *userName*, String *password*, String *carName*, String *licensePlate*, int *packageNUmber*) throws IOException {  
  
 Customer *customer* = new Customer(*userName*, *packageNUmber*, *password*, *carName*, *licensePlate*, *readPackages*());  
 Globals.*readCustomers*.add(*customer*);  
 *customer*.setPrice((int) *readPackages*.get(*customer*.getPackageNumber()).getPrice());  
 Globals.*writeObject*(Globals.*readCustomers*);  
 Globals.*readCustomers* = Globals.*readObject*();  
 }  
  
 public static Customer logIn(String *userName*, String *password*) {  
 Customer *customer* = null;  
  
 for (int *i* = 0; *i* < Globals.*readCustomers*.size(); *i*++) {  
 if (Globals.*readCustomers*.get(*i*) == null) {  
 return *customer*;  
 } else {  
 Customer *currentCustomer* = Globals.*readCustomers*.get(*i*);  
 String *customerName* = *currentCustomer*.getName();  
 String *customerPassword* = *currentCustomer*.getPassword();  
  
 if (*customerName* != null && *customerName*.equals(*userName*) && *customerPassword* != null && *customerPassword*.equals(*password*)) {  
 *customer* = *currentCustomer*;  
 break;  
 }  
 }  
 }  
  
 return *customer*;  
  
 }  
  
 public static String displayCustomers(int *i*) {  
 return String.*format*("Customer Name : %s , Car Name %s , License Number : %s , Entry Date : %s", Globals.*readCustomers*.get(*i*).getName(), Globals.*readCustomers*.get(*i*).getCarName()  
 , Globals.*readCustomers*.get(*i*).getLicensePlate(), Globals.*readCustomers*.get(*i*).getDate());  
 }  
  
 public static String searchCustomer(String *userName*) {  
 String *c* = "Not Found";  
  
 boolean *ans* = false;  
 for (int *i* = 0; *i* < Globals.*readCustomers*.size(); *i*++) {  
 if (Globals.*readCustomers*.get(*i*).getName().equals(*userName*)) {  
 /\*System.out.println("Customer Name : "+customers.get(i).getName()+" Car Name : "+customers.get(i).getCarName()  
 +" \nLicense Number : "+customers.get(i).getLicensePlate()+" Date Singed in : "+customers.get(i).getDate()  
 +" \nPackage Number : "+customers.get(i).getPackageNumber());\*/  
 *c* = *readCustomers*.get(*i*).getName();  
 break;  
  
 }  
 }  
 return *c*;  
 }  
  
 public static String deleteCustomer(String *userName*, String *password*) throws IOException {  
 String *ans* = " null";  
 for (int *i* = 0; *i* < Globals.*readCustomers*.size(); *i*++) {  
 if (Globals.*readCustomers*.get(*i*).getName().equals(*userName*) && Globals.*readCustomers*.get(*i*).getPassword().equals(*password*)) {  
 *ans* = *readCustomers*.get(*i*).getName();  
 Globals.*readCustomers*.remove(*i*);  
 Globals.*writeObject*(Globals.*readCustomers*);  
 Globals.*readCustomers* = Globals.*readObject*();  
 System.*out*.println("Successfully Deleted Selected Customer Data . ");  
 break;  
 }  
 }  
 return *ans*;  
 }  
  
 public static String deletePackage(String *userName*) throws IOException {  
 String *ans* = " null";  
 for (int *i* = 0; *i* < Globals.*readPackages*.size(); *i*++) {  
 if (Globals.*readPackages*.get(*i*).getName().equals(*userName*)) {  
 *ans* = *readPackages*.get(*i*).getName();  
 Globals.*readPackages*.remove(*i*);  
 Globals.*writePackage*(Globals.*readPackages*);  
 Globals.*readPackages* = Globals.*readPackages*();  
 System.*out*.println("Successfully Deleted Selected Customer Data . ");  
 break;  
 }  
 }  
 return *ans*;  
 }  
  
 public static String displayPackages(int *i*) {  
 return String.*format*("Package Name : %s , Package Description %s , Price : %s ", Globals.*readPackages*.get(*i*).getName(), Globals.*readPackages*.get(*i*).getDescription()  
 , Globals.*readPackages*.get(*i*).getPrice());  
 }  
  
 public static void addPackage(String *Description*, String *name*, double *price*, int *i*) {  
 Packages *packages* = new Packages(*Description*, *name*, *price*, *i*);  
  
 Globals.*readPackages*.add(*packages*);  
 Globals.*writePackage*(Globals.*readPackages*);  
 }  
  
 public static String changePrice(String *packageName*, double *price*) throws IOException {  
  
 for (int *i* = 0; *i* < Globals.*readPackages*.size(); *i*++) {  
 if (Globals.*readPackages*.get(*i*).getName().equals(*packageName*)) {  
 Globals.*readPackages*.get(*i*).setPrice(*price*);  
 Globals.*writePackage*(Globals.*readPackages*);  
 Globals.*readPackages* = Globals.*readPackages*();  
 return "changed";  
  
 }  
 }  
 return "Not found";  
 }  
  
 public static String startWashingProcess(Customer *customer*) throws IOException {  
 boolean *ans* = false;  
 for (int *i* = 0; *i* < Globals.*readCustomers*.size(); *i*++) {  
 int *slotNumber* = Globals.*readCustomers*.get(*i*).getSlotNumber();  
 if (*slotNumber* >= 0 && *slotNumber* < Globals.*slotsFree*.length) {  
 Globals.*slotsFree*[*slotNumber*] = false;  
 }  
 }  
 for (int *i* = 0; *i* < Globals.*slotsFree*.length; *i*++) {  
 if (Globals.*slotsFree*[*i*]) {  
 Globals.*slotsFree*[*i*] = false;  
 // Start washing process for customer  
  
 // Update customer's slot number and current state  
 for (Customer *c* : Globals.*readCustomers*) {  
 if (*c*.getLicensePlate().equals(*customer*.getLicensePlate())) {  
 *c*.setSlotNumber(*i*);  
 *c*.setCurrentState("in washing");  
 Globals.*writeObject*(Globals.*readCustomers*);  
 Globals.*readCustomers* = Globals.*readObject*();  
 break;  
 }  
 }  
 return "Washing process started for :" + *customer*.getName() + " 's " + *customer*.getCarName();  
 }  
 }  
  
 Globals.*readWaitingCustomers*.add(*customer*);  
 Globals.*writeWaiting*(Globals.*readWaitingCustomers*);  
 Globals.*readWaitingCustomers* = Globals.*readWait*();  
 return "No Slot free you are added to waiting customers.";  
  
  
  
 }  
  
 public static String availableSlots(int *i* ){  
 for(int *j* = 0 ; *j* <Globals.*readCustomers*.size() ; *j*++){  
 if(Globals.*readCustomers*.get(*j*).getSlotNumber()==*i*){  
 return "Slot "+*i*+" is occupied Car's State is "+Globals.*readCustomers*.get(*j*).getCurrentState();  
 }  
 }  
 return "Slot "+*i*+" is Free ";  
 }  
 public static String displayWaiting(int *i* ){  
 return "Customer "+(*i*+1)+" Details : \n"+"Name : "+Globals.*readWaitingCustomers*.get(*i*).getName()+" \nCar Name : "  
 +Globals.*readWaitingCustomers*.get(*i*).getCarName()+" \nLicense Plate : "+Globals.*readWaitingCustomers*.get(*i*).getLicensePlate()  
 +"\nWaiting From : "+Globals.*readWaitingCustomers*.get(*i*).getDate();  
 }  
  
 public static void payBill(String *user*) throws IOException {  
 for (Customer *customer* : Globals.*readCustomers*) {  
 if(*customer*.getName().equals(*user*)){  
 if (*customer*.isAMember()) {  
 System.*out*.println("Congrats you are a member of our loyalty program we have a discount for you of 20% . :)");  
 *customer*.setPrice((int) (*customer*.getPrice()\* 20 % 100));  
 Globals.*writeObject*(Globals.*readCustomers*);  
 Globals.*readCustomers* = Globals.*readObject*();  
 break;  
 }  
 else {  
 System.*out*.println("Do you want to be a member of our loyalty program ? 1/0");  
 int *ans* = Globals.*scanner*.nextInt();  
 if (*ans* == 1) {  
 for(Customer *customer2* : Globals.*readCustomers*){  
 if(*customer2*.getLicensePlate().equals(*customer*.getLicensePlate())){  
 *customer2*.setAMember(true);  
 Globals.*writeObject*(Globals.*readObject*());  
 Globals.*readCustomers* = Globals.*readObject*();  
 System.*out*.println("You will get a discount next time :)");  
 break;  
 }  
 }  
 }  
 else break;  
 }  
 break;  
 }  
 }  
 for (int *i* = 0; *i* < Globals.*readCustomers*.size(); *i*++) {  
 if (Globals.*readCustomers*.get(*i*).getName().equals(*user*)) {  
 Globals.*readCustomers*.get(*i*).setCurrentState("Bill paid");  
 int *slotNumber* = Globals.*readCustomers*.get(*i*).getSlotNumber();  
 if (*slotNumber* >= 0 && *slotNumber* < Globals.*slotsFree*.length) {  
 Globals.*slotsFree*[*slotNumber*] = true;  
 }  
 Globals.*readCustomers*.get(*i*).setSlotNumber(-1);  
 Globals.*writeObject*(Globals.*readCustomers*);  
 Globals.*readCustomers* = Globals.*readObject*();  
 break;  
 }  
 }  
 if(Globals.*readWaitingCustomers*.isEmpty()){  
 System.*out*.println("No customer in waiting list .");  
 }  
 else{  
  
 *startWashingProcess*(Globals.*readWaitingCustomers*.get(0));  
 Globals.*readWaitingCustomers*.remove(0);  
 Globals.*writeWaiting*(Globals.*readWaitingCustomers*);  
 Globals.*readWaitingCustomers* = Globals.*readWait*();  
 }  
 }  
 }  
  
  
public class AutoMaticCarWashSystem {  
 public static void main(String[] *args*) throws IOException {  
 //new AdminWindow();  
 // new CustomerWindow();  
 new CarWashGUI();  
 //CustomerWindow customerWindow = new CustomerWindow();  
 boolean[] *slots* = new boolean[Globals.*slots*];  
 Arrays.*fill*(*slots*,true);  
 Globals.*setSlotsFree*(*slots*);  
  
 Globals.*readCustomers* = Globals.*readObject*();  
 for(Customer *c* : Globals.*readCustomers*){  
 System.*out*.println(*c*.getName()+" "+*c*.getSlotNumber());  
 }  
 final String *adminId* = "a";  
 final String *adminPassword* = "a";  
 //Packages  
  
 Packages *package1* = new Packages("Basic exterior wash and drying.","Quick Wash",9.99,3);  
 Packages *package2* = new Packages("Exterior wash, interior vacuuming, and window cleaning.","Deluxe Wash",  
 19.99,5);  
 Packages *package3* = new Packages("Full exterior and interior cleaning, waxing, and tire dressing.",  
 "Premium Detailing",29.99,7);  
 ArrayList<Packages> *list* = new ArrayList<>();  
 *list*.add(*package1*);  
 *list*.add(*package2*);  
 *list*.add(*package3*);  
 // Globals.writePackage(list);  
 //Creating an arraylist of packages and then setting it in class og package list  
 Globals.*readPackages* = Globals.*readPackages*();  
 /\* readPackages.add(package1);  
 readPackages.add(package2);  
 readPackages.add(package3);  
 Globals.writePackage(readPackages);\*/  
  
  
 PackageList *packageList* = new PackageList(Globals.*readPackages*) ;  
  
 Globals.*writePackage*(Globals.*readPackages*);  
  
  
 while(true) {  
 String *userName*;  
 String *password*;  
 String *carName*;  
 String *licensePlate*;  
 int *packageNUmber*;  
 int *choice*;  
 String *idAdmin* ;  
 String *passwordAdmin*;  
 Customer *customer* = null;  
 System.*out*.println("1. Admin \n2. Customer \n0. Exit");  
 *choice* = Globals.*scanner*.nextInt();  
 Globals.*scanner*.nextLine();  
 if(*choice*==0) {  
 System.*out*.println("You Exited");  
 break;  
 }  
 //Admin Panel  
 if (*choice* == 1) {  
 System.*out*.println("Enter Admin id : ");  
 *idAdmin* = Globals.*scanner*.nextLine();  
 System.*out*.println("Enter Admin password : ");  
 *passwordAdmin* = Globals.*scanner*.nextLine();  
 if (*idAdmin*.equals(*adminId*) && *passwordAdmin*.equals(*adminPassword*)) {  
 while(true){  
  
 System.*out*.println("1. Display All Customers .\n" +  
 "2. Delete Customer. \n" +  
 "3. Search Customer.\n" +  
 "4. Display All packages.\n" +  
 "5. Add Package.\n" +  
 "6. Delete Package. \n" +  
 "7. Change Price of Package. \n" +  
 "8. Display Waiting. \n"+  
 "0. Close Admin Panel. ");  
 *choice* = Globals.*scanner*.nextInt();  
 if(*choice*==0) break;  
 Globals.*scanner*.nextLine();  
 if (*choice* == 1) {  
 for (int *i* = 0; *i* < Globals.*readCustomers*.size(); *i*++) {  
 System.*out*.println("Customer Name : " + Globals.*readCustomers*.get(*i*).getName() +  
 ", Car Name : " + Globals.*readCustomers*.get(*i*).getCarName() + ", " +  
 "License Number : " + Globals.*readCustomers*.get(*i*).getLicensePlate());  
 }  
 } else if (*choice* == 2) {  
 System.*out*.println("Enter Customer name : ");  
 *userName* = Globals.*scanner*.nextLine();  
 System.*out*.println("Enter License Plate : ");  
 *licensePlate* = Globals.*scanner*.nextLine();  
 boolean *ans* = false;  
 for (int *i* = 0; *i* < Globals.*readCustomers*.size(); *i*++) {  
 if (Globals.*readCustomers*.get(*i*).getName().equals(*userName*) && Globals.*readCustomers*.get(*i*).getLicensePlate().equals(*licensePlate*)) {  
 Globals.*readCustomers*.remove(*i*);  
 Globals.*writeObject*(Globals.*readCustomers*);  
 Globals.*readCustomers* = Globals.*readObject*();  
 System.*out*.println("Successfully Deleted Selected Customer Data . ");  
 *ans* = true;  
 break;  
 }  
 }  
 if (!*ans*) System.*out*.println("Customer Not found . ");  
  
 } else if (*choice* == 3) {  
 System.*out*.println("Enter Customer name : ");  
 *userName* = Globals.*scanner*.nextLine();  
  
 boolean *ans* = false;  
 for (int *i* = 0; *i* < Globals.*readCustomers*.size(); *i*++) {  
 if (Globals.*readCustomers*.get(*i*).getName().equals(*userName*)) {  
 /\*System.out.println("Customer Name : "+customers.get(i).getName()+" Car Name : "+customers.get(i).getCarName()  
 +" \nLicense Number : "+customers.get(i).getLicensePlate()+" Date Singed in : "+customers.get(i).getDate()  
 +" \nPackage Number : "+customers.get(i).getPackageNumber());\*/  
 Globals.*readCustomers*.get(*i*).display();  
 *ans* = true;  
 }  
 }  
 if (!*ans*) System.*out*.println("Customer Not found . ");  
 } else if (*choice* == 4) {  
 for (int *i* = 0, *p* = 1; *i* < Globals.*readPackages*.size(); *i*++, *p*++) {  
 System.*out*.println("Package " + *p* + " \nDescription : " + Globals.*readPackages*.get(*i*).getDescription() +  
 " \nName : " + Globals.*readPackages*.get(*i*).getName()  
 + " \nPrice : " + Globals.*readPackages*.get(*i*).getPrice());  
 }  
 } else if (*choice* == 5) {  
 Packages *packages* = new Packages();  
 System.*out*.println("Enter Description : ");  
 String *Description* = Globals.*scanner*.nextLine();  
 *packages*.setDescription(*Description*);  
 System.*out*.println("Enter Name : ");  
 String *name* = Globals.*scanner*.nextLine();  
 *packages*.setName(*name*);  
 System.*out*.println("Enter Price : ");  
 double *price* = Globals.*scanner*.nextDouble();  
 *packages*.setPrice(*price*);  
 Globals.*readPackages*.add(*packages*);  
 Globals.*writePackage*(Globals.*readPackages*);  
  
 } else if (*choice* == 6) {  
 System.*out*.println("Enter package Name to delete : ");  
 String *packageName* = Globals.*scanner*.nextLine();  
 boolean *ans* = false;  
 for (Packages *p* : Globals.*readPackages*) {  
 if (*p*.getName().equals(*packageName*)) {  
 Globals.*readPackages*.remove(*p*);  
 Globals.*writePackage*(Globals.*readPackages*);  
 Globals.*readPackages* = Globals.*readPackages*();  
 *ans* = true;  
 break;  
 }  
 }  
 if (!*ans*) System.*out*.println("Package not found.");  
 } else if (*choice* == 7 ) {  
 System.*out*.println("Enter package Name to change price : ");  
 String *packageName* = Globals.*scanner*.nextLine();  
 boolean *ans* = false;  
 for (int *i* = 0; *i* < Globals.*readPackages*.size(); *i*++) {  
 if (Globals.*readPackages*.get(*i*).getName().equals(*packageName*)) {  
 System.*out*.println("Enter your new price : ");  
 Globals.*readPackages*.get(*i*).setPrice(Globals.*scanner*.nextDouble());  
 Globals.*writePackage*(Globals.*readPackages*);  
 Globals.*readPackages* = Globals.*readPackages*();  
 *ans* = true;  
 break;  
 }  
 }  
 if (!*ans*) System.*out*.println("Package not found.");  
 }  
 else{  
 Globals.*displayWaiting*();  
 }  
 }  
 } else {  
 System.*out*.println("Mismatched data. ");  
 }  
  
 }  
  
 //Customer panel  
 else {  
 while (true){  
 System.*out*.println("1. Login \n2. SignUp \n0. Exit");  
 *choice* = Globals.*scanner*.nextInt();  
 Globals.*scanner*.nextLine();  
 if (*choice* == 0) {  
 break;  
 } else if(*choice*==1) {  
  
 System.*out*.println("Enter Username : ");  
 *userName* = Globals.*scanner*.nextLine();  
 System.*out*.println("Enter password : ");  
 *password* = Globals.*scanner*.nextLine();  
 boolean *ans* = false ;  
 for(int *i* = 0 ; *i* < Globals.*readCustomers* .size() ; *i*++){  
 if(Globals.*readCustomers* .get(*i*).getName().equals(*userName*)&&Globals.*readCustomers* .get(*i*).getPassword().equals(*password*)){  
 *ans* = true ;  
 *customer* = Globals.*readCustomers* .get(*i*);  
 break;  
 }  
 }  
 if (!*ans*) {  
 System.*out*.println("Enter again");  
 }  
 else {  
 System.*out*.println("Logged In Successfully");  
 break;  
 }  
 } else if (*choice*==2) {  
  
 System.*out*.println("Enter Username : ");  
 *userName* = Globals.*scanner*.nextLine();  
 System.*out*.println("Enter password : ");  
 *password* = Globals.*scanner*.nextLine();  
 System.*out*.println("Enter your car name : ");  
 *carName* = Globals.*scanner*.nextLine();  
 System.*out*.println("Enter License Plate number of : " + *carName*);  
 *licensePlate* = Globals.*scanner*.nextLine();  
 System.*out*.println("We have following packages available . Please Select ");  
 while (true){  
 Globals.*readPackages* = Globals.*readPackages*();  
 for (Packages *p* : Globals.*readPackages*) {  
 System.*out*.println(*p*.getName() + " , " + *p*.getDescription() + " , " + *p*.getPrice());  
 }  
  
 *packageNUmber* = Globals.*scanner*.nextInt();  
 if(*packageNUmber*>Globals.*readPackages*.size()-1){  
 System.*out*.println("No such package.");  
 }  
 else{  
 break;  
 }  
 }  
 *customer* = new Customer(*userName*, *packageNUmber*, *password*, *carName*, *licensePlate*, *packageList*.getPackages());  
 Globals.*readCustomers* .add(*customer*);  
 *customer*.setPrice((int) *packageList*.getPackages().get(*customer*.getPackageNumber()).getPrice());  
 Globals.*writeObject*(Globals.*readCustomers* );  
 Globals.*readCustomers* = Globals.*readObject*();  
  
 for(int *i* = 0 ; *i* < Globals.*readCustomers* .size() ; *i*++){  
 System.*out*.println(Globals.*readCustomers* .get(*i*).getCarName());  
 }  
 System.*out*.println("You Signed up successfully.");  
 break;  
 }  
 }  
 System.*out*.println("1. Car Wash \n2. Pay Bill");  
 *choice* = Globals.*scanner*.nextInt();  
 //Car Wash Procedure  
 if(*choice* ==1){  
 CarWash *carWash* = new CarWash(*packageList*.getPackages(),*customer*);  
 *carWash*.startWashingProcess(*customer*);  
  
 }  
 //Bill Payment  
 else{  
 Pay *pay* = new Pay(Globals.*readPackages*,*customer*);  
 *pay*.payBill(*customer*);  
 }  
  
 }  
 }  
 }  
}  
  
//Create Packages  
class Packages implements Serializable{  
 private String name ;  
 private double price ;  
 private int timeInSeconds ;  
 private String description ;  
 Packages(){  
  
 }  
 Packages(String *description* , String *name* , double *price* , int *timeInSeconds*){  
 this.description = *description* ;  
 this.price = *price* ;  
 this.name = *name* ;  
 this.timeInSeconds = *timeInSeconds* ;  
 }  
  
 public void setName(String *name*) {  
 this.name = *name*;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public void setPrice(double *price*) {  
 this.price = *price*;  
 }  
  
 public void setDescription(String *description*) {  
 this.description = *description*;  
 }  
  
 public String getDescription() {  
 return description;  
 }  
  
 public int getTimeInSeconds() {  
 return timeInSeconds;  
 }  
  
 public void setTimeInSeconds(int *timeInSeconds*) {  
 this.timeInSeconds = *timeInSeconds*;  
 }  
}  
class PackageList implements Serializable{  
  
 ArrayList<Packages> packages;  
  
 PackageList(ArrayList<Packages> *packages* ){  
 this.packages = new ArrayList<>();  
 this.packages = *packages* ;  
 }  
  
 public PackageList() {  
  
 }  
  
 public ArrayList<Packages> getPackages() {  
 return packages;  
 }  
  
 public void setPackages(ArrayList<Packages> *packages*) {  
 this.packages = *packages*;  
 }  
  
 public void startWashingProcess(Customer *customer*) throws IOException {  
 System.*out*.println("Washing process of Car : "+*customer*.getLicensePlate());  
  
  
 }  
  
 public void addPackage(Packages *p*){  
 packages.add(*p*);  
 System.*out*.println("Package "+*p*.getName()+" added successfully");  
 System.*out*.println("Do you want to view all packages ? ");  
 boolean *ans* = Globals.*scanner*.hasNext();  
 if(*ans*){  
 displayPackages();  
 }  
 }  
 public void deletePackage(String *name*){  
 Packages *toBeDeleted* = null;  
 boolean *ans* = false ;  
 for (Packages *p* : packages){  
 if(*p*.getName().equals(*name*)){  
 *ans* = true ;  
 *toBeDeleted* = *p* ;  
 packages.remove(*p*);  
 break;  
 }  
 }  
 if(*ans*){  
 System.*out*.println("Package "+*toBeDeleted*.getName()+" deleted successfully\n");  
 }  
 else{  
 System.*out*.println("No package found \n");  
 }  
 }  
 public void displayPackages(){  
 int *count* = 1 ;  
 System.*out*.println("------------------------------------------------------------");  
 for(int *i* = 0 ; *i*< packages.size() ; *i*++){  
 System.*out*.println("Package " +*count*+" \nName : "+packages.get(*i*).getName()+" \nPrice : "+packages.get(*i*).getPrice()  
 +" \nDescription : "+packages.get(*i*).getDescription());  
 System.*out*.println("------------------------------------------------------------");  
 *count*++;  
 }  
 }  
}  
// class Car implements Serializable{  
  
  
  
  
// Car(){}  
// // public abstract void display();  
// }  
class Customer implements Serializable {  
 public String carName ;  
 public String licensePlate ;  
 private String name ;  
 private int packageNumber ;  
 private String password ;  
 private double price = 0 ;  
 private int slotNumber = -1 ;  
 private boolean isAMember = false ;  
 private Date date = new Date();  
 private String currentState = "waiting";  
  
 Customer(){  
  
 }  
 Customer(String *name* , int *packageNumber* , String *password* , String *carName* , String *licensePlate*,ArrayList<Packages> *p* ){  
 this.name = *name* ;  
 this.packageNumber = *packageNumber* ;  
 this.carName = *carName* ;  
 this.licensePlate = *licensePlate* ;  
 this.password = *password* ;  
  
 }  
  
 public boolean isAMember() {  
 return isAMember;  
 }  
  
 public void setAMember(boolean *AMember*) {  
 isAMember = *AMember*;  
 }  
  
 public void setPrice(int *price*) {  
 this.price = *price*;  
 }  
  
 public int getSlotNumber() {  
 return slotNumber;  
 }  
  
 public void setSlotNumber(int *slotNumber*) {  
 this.slotNumber = *slotNumber*;  
 }  
  
 public String getCarName() {  
 return carName;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setCurrentState(String *currentState*) {  
 this.currentState = *currentState*;  
 }  
  
 public String getCurrentState() {  
 return String.*format*("License Plate : %s is in %s ",getLicensePlate(),currentState);  
 }  
 public void setName(String *name*) {  
 this.name = *name*;  
 }  
  
 public Date getDate() {  
 return date;  
 }  
  
 public int getPackageNumber() {  
 return packageNumber;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
 public String getLicensePlate() {  
 return licensePlate;  
 }  
  
 public void setDate(Date *date*) {  
 this.date = *date*;  
 }  
  
 public void setPassword(String *password*) {  
 this.password = *password*;  
 }  
  
  
 public void display(){  
 System.*out*.println("------------------");  
 System.*out*.println("Car Details : "+"\nCar Name : "+getCarName()+"\nLicense Plate : "+getLicensePlate()+  
 "\nOwner Details : "+"\nOwner Name : "+name +"\nPackage Number : "+packageNumber);  
 }  
}  
class CarWash extends PackageList{  
 private Customer customer ;  
 CarWash(ArrayList<Packages> *p* , Customer *customer*){  
 super(*p*);  
 this.customer = *customer*;  
 }  
  
 @Override  
 public void startWashingProcess(Customer *customer*) throws IOException {  
 boolean *ans* = false ;  
 for (int *i* = 0; *i* < Globals.*readCustomers*.size(); *i*++) {  
 int *slotNumber* = Globals.*readCustomers*.get(*i*).getSlotNumber();  
 if (*slotNumber* >= 0 && *slotNumber* < Globals.*slotsFree*.length) {  
 Globals.*slotsFree*[*slotNumber*] = false;  
 }  
 }  
 for (int *i* = 0; *i* < Globals.*slotsFree*.length; *i*++) {  
 if (Globals.*slotsFree*[*i*]) {  
 Globals.*slotsFree*[*i*] = false;  
 // Start washing process for customer  
 super.startWashingProcess(*customer*);  
 // Update customer's slot number and current state  
 for (Customer *c* : Globals.*readCustomers*) {  
 if (*c*.getLicensePlate().equals(*customer*.getLicensePlate())) {  
 *c*.setSlotNumber(*i*);  
 *c*.setCurrentState("in washing");  
 Globals.*writeObject*(Globals.*readCustomers*);  
 Globals.*readCustomers* = Globals.*readObject*();  
 break;  
 }  
 }  
 *ans* = true;  
 break;  
 }  
 }  
  
 if(!*ans*) {  
 System.*out*.println("No slot free you are added to waiting area .");  
 Globals.*readWaitingCustomers*.add(*customer*);  
 Globals.*writeWaiting*(Globals.*readWaitingCustomers*);  
 Globals.*readWaitingCustomers* = Globals.*readWait*();  
 }  
 else{  
  
 }  
 }  
  
 @Override  
 public void displayPackages() {  
 super.displayPackages();  
 }  
  
 public Customer getCustomer() {  
 return customer;  
 }  
  
 public void setCustomer(Customer *customer*) {  
 this.customer = *customer*;  
 }  
  
 @Override  
 public void addPackage(Packages *p*) {  
 super.addPackage(*p*);  
 }  
  
 @Override  
 public void deletePackage(String *name*) {  
 super.deletePackage(*name*);  
 }  
  
}  
  
  
class Pay extends CarWash{  
  
 Pay(ArrayList<Packages> *p* , Customer *c* ){  
 super(*p*,*c*);  
 }  
 public void payBill(Customer *customer*) throws IOException {  
 //customer.setCurrentState("On Cash Counter");  
 for (Customer *customer1* : Globals.*readCustomers*) {  
 if(*customer1*.getName().equals(*customer*.getName())){  
 if (*customer*.isAMember()) {  
 System.*out*.println("Congrats you are a member of our loyalty program we have a discount for you of 20% . :)");  
 *customer*.setPrice((int) super.packages.get(*customer*.getPackageNumber()).getPrice() \* 20 % 100);  
 break;  
 }  
 else {  
 System.*out*.println("Do you want to be a member of our loyalty program ? 1/0");  
 int *ans* = Globals.*scanner*.nextInt();  
 if (*ans* == 1) {  
 for(Customer *customer2* : Globals.*readCustomers*){  
 if(*customer2*.getLicensePlate().equals(*customer*.getLicensePlate())){  
 *customer2*.setAMember(true);  
 Globals.*writeObject*(Globals.*readObject*());  
 Globals.*readCustomers* = Globals.*readObject*();  
 System.*out*.println("You will get a discount next time :)");  
 break;  
 }  
 }  
 }  
 else break;  
 }  
 break;  
 }  
 }  
 for (int *i* = 0; *i* < Globals.*readCustomers*.size(); *i*++) {  
 if (Globals.*readCustomers*.get(*i*).getName().equals(*customer*.getName())) {  
 Globals.*readCustomers*.get(*i*).setCurrentState("Bill paid");  
 int *slotNumber* = Globals.*readCustomers*.get(*i*).getSlotNumber();  
 if (*slotNumber* >= 0 && *slotNumber* < Globals.*slotsFree*.length) {  
 Globals.*slotsFree*[*slotNumber*] = true;  
 }  
 Globals.*readCustomers*.get(*i*).setSlotNumber(-1);  
 Globals.*writeObject*(Globals.*readCustomers*);  
 Globals.*readCustomers* = Globals.*readObject*();  
 break;  
 }  
 }  
 if(Globals.*readWaitingCustomers*.isEmpty()){  
 System.*out*.println("No customer in waiting list .");  
 }  
 else{  
  
 startWashingProcess(Globals.*readWaitingCustomers*.get(0));  
 Globals.*readWaitingCustomers*.remove(0);  
 Globals.*writeWaiting*(Globals.*readWaitingCustomers*);  
 Globals.*readWaitingCustomers* = Globals.*readWait*();  
 }  
 }  
  
}