

KK14203

OBJECT-ORIENTED PROGRAMMING

NAME : NUR FARAHIN BINTI JAMIRIN

MATRIC NO. : BI1911010

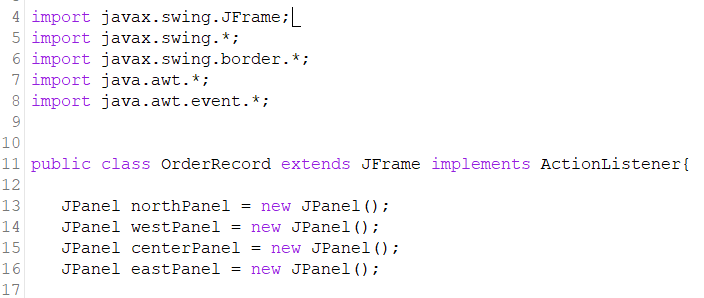
LECTURER’S NAME : DR. SAMRY@MOHD SHAMRIE SAININ

Java Code

1 /\*NUR FARAHIN BINTI JAMIRIN  
 2 BI19110107\*/  
 3   
 4 import javax.swing.JFrame;  
 5 import javax.swing.\*;  
 6 import javax.swing.border.\*;  
 7 import java.awt.\*;  
 8 import java.awt.event.\*;  
 9   
 10   
 11 public class OrderRecord extends JFrame implements ActionListener{  
 12   
 13 private JButton calc, submit;  
 14   
 15 JPanel northPanel = new JPanel();  
 16 JPanel westPanel = new JPanel();  
 17 JPanel centerPanel = new JPanel();  
 18 JPanel eastPanel = new JPanel();  
 19   
 20 JLabel Total = new JLabel("Price");  
 21 JLabel Rating = new JLabel("Rating");  
 22 JLabel end = new JLabel("Please give a rating.");  
 23   
 24 JTextField TotalResult = new JTextField(8);  
 25   
 26 JRadioButton burger = new JRadioButton("Burger");  
 27 JRadioButton sp = new JRadioButton("Spagetti");  
 28 JRadioButton fc = new JRadioButton("Fried Chicken");  
 29 JRadioButton rate1 = new JRadioButton("1");  
 30 JRadioButton rate2 = new JRadioButton("2");  
 31 JRadioButton rate3 = new JRadioButton("3");  
 32   
 33 JCheckBox coffee = new JCheckBox ("Coffee");  
 34 JCheckBox tea = new JCheckBox ("Tea");  
 35 JCheckBox mt\_dew = new JCheckBox ("Mountain Dew");  
 36 JCheckBox pepsi= new JCheckBox ("Pepsi");  
 37   
 38 //variable for calculate  
 39 double foodCost = 0;  
 40 double DrinkCost = 0;  
 41 double TotalCost = 0;  
 42   
 43 public OrderRecord(){  
 44   
 45 northPanel.setLayout(new GridLayout());  
 46 northPanel.setBorder(new TitledBorder("Food:"));  
 47   
 48 calc = new JButton("Calculate");  
 49 submit = new JButton("Submit");  
 50   
 51 ButtonGroup food = new ButtonGroup();  
 52 food.add(burger);  
 53 food.add(sp);  
 54 food.add(fc);  
 55 //register listeners  
 56 burger.addActionListener(this);  
 57 sp.addActionListener(this);  
 58 fc.addActionListener(this);  
 59 calc.addActionListener(this);  
 60 submit.addActionListener(this);  
 61   
 62 northPanel.add(burger);  
 63 northPanel.add(sp);  
 64 northPanel.add(fc);  
 65   
 66 westPanel.setLayout(new GridLayout(4, 3, 5, 5));  
 67 westPanel.setBorder(new TitledBorder("Drinks:"));  
 68 westPanel.add(coffee);  
 69 westPanel.add(tea);  
 70 westPanel.add(mt\_dew);  
 71 westPanel.add(pepsi);  
 72 westPanel.add(calc);  
 73   
 74 centerPanel.setLayout(new GridLayout(4,2));  
 75 centerPanel.add(Total);  
 76 centerPanel.add(TotalResult);  
 77 centerPanel.add(end);  
 78   
 79 eastPanel.setLayout(new GridLayout(6,2));  
 80 eastPanel.add(Rating);  
 81 eastPanel.add(rate1);  
 82 eastPanel.add(rate2);  
 83 eastPanel.add(rate3);  
 84 eastPanel.add(submit);  
 85   
 86 setLayout(new BorderLayout());  
 87 add(northPanel, BorderLayout.NORTH);  
 88 add(westPanel, BorderLayout.WEST);  
 89 add(centerPanel, BorderLayout.CENTER);  
 90 add(eastPanel, BorderLayout.EAST);  
 91   
 92 }  
 93   
 94 public double Burger()   
 95 {  
 96 return foodCost = 2.50;  
 97 }  
 98 public double Spaghetti()   
 99 {  
100 return foodCost = 5.00;  
101 }  
102 public double FriedChicken()   
103 {  
104 return foodCost = 3.00;  
105 }  
106 public double setDrinkCost()  
107 {   
108 if ( coffee.isSelected())  
109 DrinkCost = 1.80;  
110   
111 else if ( tea.isSelected())  
112 DrinkCost = 1.80;  
113   
114 else if (mt\_dew.isSelected())  
115 DrinkCost = 2.00;  
116   
117 else if (pepsi.isSelected())  
118 DrinkCost = 2.00;  
119   
120 return DrinkCost;  
121 }  
122   
123 public void calculate(double foodCost, double DrinkCost)   
124 {  
125   
126 TotalCost = foodCost + DrinkCost;  
127 TotalResult.setText(String.valueOf(TotalCost));  
128   
129 }  
130   
131 public void actionPerformed(ActionEvent e)   
132 {  
133 if (e.getSource() == burger)  
134 Burger();  
135 else if (e.getSource() == sp)  
136 Spaghetti();  
137 else if (e.getSource() == fc)  
138 FriedChicken();  
139 setDrinkCost();  
140 if (e.getSource() == calc)  
141 calculate(foodCost, DrinkCost);  
142 if(e.getSource() == submit)  
143 end.setText("Thank you for coming!");  
144 }  
145   
146   
147 public static void main (String[] args)  
148 {  
149 OrderRecord order = new OrderRecord();  
150 // set window title  
151 order.setTitle("Order Record");  
152 // set window location  
153 order.setLocationRelativeTo(null);  
154 // specify window location  
155 order.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
156 // set window size  
157 order.setSize(500, 300);  
158 // display window  
159 order.setVisible(true);  
160 }  
161   
162   
163 }

Object-Oriented Implementation

Inheritance

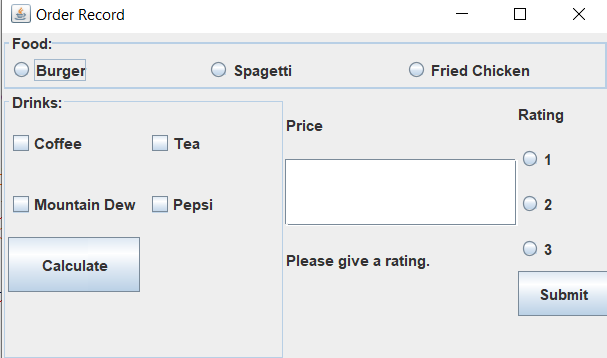


OrderRecord inherits from the javax.swing.JFrame class.

Encapsulation

Encapsulation describes the idea of bundling data and methods that work on that data within one unit.

User Manual



The user first need to choose food and drinks that they wanted. After they are done choosing, they can press the calculate button to show the total price and then give a rating after they are done.