**Practical No. 3**

**Aim**: Design Pizza Order System using swing components

**Theory:**

The provided Java code is a simple Pizza Order Application implemented using the Swing graphical user interface (GUI) framework.

**Swing Framework:**

Swing is a set of GUI components for Java applications, providing a toolkit for building graphical user interfaces. In this program, Swing components such as JFrame, JPanel, JLabel, JTextField, JComboBox, JCheckBox, and JButton are used to create a user-friendly pizza ordering interface.

**Class Structure:**

The code defines a class named PizzaOrderApplication that extends the JFrame class. This class serves as the main window for the pizza ordering application, encapsulating the entire functionality.

**GUI Layout:**

The graphical user interface is structured using layout managers (GridLayout and BorderLayout). The main input form is organized into a JPanel with a grid layout, containing labels and corresponding input components for customer details, pizza type, toppings, quantity, and payment options.

**User Input:**

Various Swing components, such as text fields, combo boxes, and check boxes, are employed to collect user input. Users can enter their name, address, mobile number, select pizza type, choose toppings, specify quantity, and pick a payment option.

**Event Handling:**

Event handling is implemented using the ActionListener interface. The program responds to user actions, such as clicking the "Place Order" button or selecting a pizza type or toppings. The corresponding event listeners trigger actions, such as placing an order or updating the total price dynamically.

**Dynamic Updates:**

The application dynamically updates the total price based on the selected pizza type, toppings, and quantity. The updatePrice method calculates the total price and displays it in the designated text field.

**Order Placement:**

When the user clicks the "Place Order" button, the placeOrder method retrieves the entered information, constructs an order details string, and displays it in a confirmation dialog using JOptionPane. The input fields are then cleared for the next order.

**Main Method:**

The main method creates an instance of the PizzaOrderApplication class, initiating the application and displaying the main frame.

**Source Code:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

class PizzaOrderApplication extends JFrame {

JTextField customerNameField, addressField, mobileNumberField, quantityField, priceField;

JComboBox pizzaTypeComboBox, paymentOptionComboBox;

JCheckBox pepperoniCheckBox, mushroomsCheckBox, extraCheeseCheckBox;

JButton orderButton;

ArrayList<Double> pizzaPrices;

PizzaOrderApplication() {

setTitle("Pizza Order Application");

setSize(400, 500);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

initUI();

setVisible(true);

}

private void initUI() {

setLayout(new BorderLayout());

JPanel formPanel = new JPanel(new GridLayout(10, 2, 10, 10));

formPanel.add(new JLabel("Customer Name:"));

customerNameField = new JTextField();

formPanel.add(customerNameField);

formPanel.add(new JLabel("Address:"));

addressField = new JTextField();

formPanel.add(addressField);

formPanel.add(new JLabel("Mobile Number:"));

mobileNumberField = new JTextField();

formPanel.add(mobileNumberField);

formPanel.add(new JLabel("Pizza Type:"));

pizzaTypeComboBox = new JComboBox(new String[]{"Margherita", "Pepperoni", "Vegetarian"});

formPanel.add(pizzaTypeComboBox);

formPanel.add(new JLabel("Toppings:"));

pepperoniCheckBox = new JCheckBox("Pepperoni");

mushroomsCheckBox = new JCheckBox("Mushrooms");

extraCheeseCheckBox = new JCheckBox("Extra Cheese");

JPanel toppingsPanel = new JPanel(new FlowLayout());

toppingsPanel.add(pepperoniCheckBox);

toppingsPanel.add(mushroomsCheckBox);

toppingsPanel.add(extraCheeseCheckBox);

formPanel.add(toppingsPanel);

formPanel.add(new JLabel("Quantity:"));

quantityField = new JTextField();

formPanel.add(quantityField);

formPanel.add(new JLabel("Price:"));

priceField = new JTextField();

priceField.setEditable(false);

formPanel.add(priceField);

formPanel.add(new JLabel("Payment Option:"));

paymentOptionComboBox = new JComboBox(new String[]{"Credit Card", "Cash", "Online Payment"});

formPanel.add(paymentOptionComboBox);

formPanel.add(new JLabel()); // Empty label for layout

orderButton = new JButton("Place Order");

formPanel.add(orderButton);

add(formPanel, BorderLayout.CENTER);

orderButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

placeOrder();

}

});

pizzaPrices = new ArrayList<>();

pizzaPrices.add(8.99); // Margherita

pizzaPrices.add(10.99); // Pepperoni

pizzaPrices.add(9.99); // Vegetarian

pizzaTypeComboBox.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

updatePrice();

}

});

pepperoniCheckBox.addActionListener(e -> updatePrice());

mushroomsCheckBox.addActionListener(e -> updatePrice());

extraCheeseCheckBox.addActionListener(e -> updatePrice());

}

void placeOrder() {

String customerName = customerNameField.getText();

String address = addressField.getText();

String mobileNumber = mobileNumberField.getText();

String pizzaType = (String) pizzaTypeComboBox.getSelectedItem();

String toppings = getSelectedToppings();

String quantity = quantityField.getText();

String price = priceField.getText();

String paymentOption = (String) paymentOptionComboBox.getSelectedItem();

String orderDetails = "Order Details:\n" +

"Customer Name: " + customerName + "\n" +

"Address: " + address + "\n" +

"Mobile Number: " + mobileNumber + "\n" +

"Pizza Type: " + pizzaType + "\n" +

"Toppings: " + toppings + "\n" +

"Quantity: " + quantity + "\n" +

"Price: $" + price + "\n" +

"Payment Option: " + paymentOption;

JOptionPane.showMessageDialog(this, orderDetails, "Order Confirmation", JOptionPane.INFORMATION\_MESSAGE);

clearInputFields();

}

void updatePrice() {

int selectedIndex = pizzaTypeComboBox.getSelectedIndex();

double basePrice = pizzaPrices.get(selectedIndex);

int selectedToppings = getSelectedToppingsCount();

double totalPrice = basePrice + (selectedToppings \* 1.50);

try {

int quantity = Integer.parseInt(quantityField.getText());

totalPrice \*= quantity;

} catch (NumberFormatException e) {

// Ignore invalid quantity input

}

priceField.setText(String.format("%.2f", totalPrice));

}

String getSelectedToppings() {

StringBuilder toppings = new StringBuilder();

if (pepperoniCheckBox.isSelected()) {

toppings.append("Pepperoni ");

}

if (mushroomsCheckBox.isSelected()) {

toppings.append("Mushrooms ");

}

if (extraCheeseCheckBox.isSelected()) {

toppings.append("Extra Cheese ");

}

return toppings.toString().trim();

}

int getSelectedToppingsCount() {

int count = 0;

if (pepperoniCheckBox.isSelected()) {

count++;

}

if (mushroomsCheckBox.isSelected()) {

count++;

}

if (extraCheeseCheckBox.isSelected()) {

count++;

}

return count;

}

void clearInputFields() {

customerNameField.setText("");

addressField.setText("");

mobileNumberField.setText("");

pizzaTypeComboBox.setSelectedIndex(0);

pepperoniCheckBox.setSelected(false);

mushroomsCheckBox.setSelected(false);

extraCheeseCheckBox.setSelected(false);

quantityField.setText("");

priceField.setText("");

paymentOptionComboBox.setSelectedIndex(0);

}

public static void main(String[] args) {

new PizzaOrderApplication();

}

}

**Output:**

