Exp-13

//Khan Arman

//231P072,13

//Write a program to display Human Face using applet.

import javax.swing.\*; import java.awt.\*;

public class HumanFaceFrame extends JPanel {

@Override public void paintComponent(Graphics g) { super.paintComponent(g);

// Set the color for the face g.setColor(Color.YELLOW);

// Draw the face (Oval)

g.fillOval(100, 100, 200, 200); // x, y, width, height

// Set the color for the eyes g.setColor(Color.BLACK);

// Draw the eyes (Ovals)

g.fillOval(150, 150, 30, 30); // Left eye

g.fillOval(220, 150, 30, 30); // Right eye

// Draw the nose (Line)

g.drawLine(200, 180, 200, 220); // Nose

// Set the color for the mouth g.setColor(Color.RED);

// Draw the mouth (Arc)

g.drawArc(150, 230, 100, 50, 0, -180); // x, y, width, height, startAngle, arcAngle

}

public static void main(String[] args) {

// Create a frame to display the face JFrame frame = new JFrame("Human Face"); frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); frame.setSize(400, 400);

// Create an instance of the JPanel subclass (where the face is drawn) HumanFaceFrame facePanel = new HumanFaceFrame();

// Add the panel to the frame

frame.add(facePanel);

// Make the frame visible frame.setVisible(true);

}

}



//Khan Arman

//231P072,13

//WAP:Write an applet to all four types of rectangle (normal rectangle, filled rectangle, round rectangle, round corner rectangle and filled round corner rectangle).

import java.awt.\*; import javax.swing.\*; import java.applet.Applet;

public class RectangleApplet extends Applet {

public void paint(Graphics g) { // Normal Rectangle (Outlined)

g.drawRect(50, 50, 100, 60); // x, y, width, height

g.drawString("Normal Rectangle", 50, 130); // Label

// Filled Rectangle

g.setColor(Color.BLUE); // Set color to blue

g.fillRect(200, 50, 100, 60); // x, y, width, height

g.setColor(Color.BLACK); // Reset color to black

g.drawString("Filled Rectangle", 200, 130); // Label

// Round Rectangle (Outlined)

g.drawRoundRect(50, 150, 100, 60, 30, 30); // x, y, width, height, arcWidth, arcHeight

g.drawString("Round Rectangle", 50, 230); // Label

// Filled Round-Corner Rectangle

g.setColor(Color.RED); // Set color to red

g.fillRoundRect(200, 150, 100, 60, 30, 30); // x, y, width, height, arcWidth, arcHeight

g.setColor(Color.BLACK); // Reset color to black

g.drawString("Filled Round-Corner Rectangle", 200, 230); // Label

}

// Main method to run applet as a Java application

public static void main(String[] args) {

JFrame frame = new JFrame("Rectangle Applet");

RectangleApplet applet = new RectangleApplet();

// Initialize the applet (same as if it were run in a browser) applet.init();

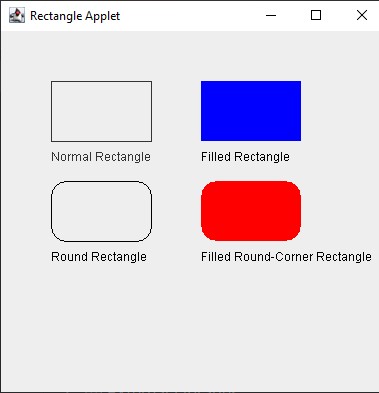
applet.start();

// Add the applet to the frame frame.add(applet); frame.setSize(400, 400);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); frame.setVisible(true);

}

}



//Khan Arman

//231P072,13

//Write a program to display circle and filled circle using Applet import java.awt.\*; import javax.swing.\*;

import java.applet.Applet;

public class CircleApplet extends Applet {

public void paint(Graphics g) {

// Outlined Circle

g.drawOval(50, 50, 100, 100); // x, y, width, height

g.drawString("Outlined Circle", 70, 170); // Label

// Filled Circle

g.setColor(Color.RED); // Set color to red

g.fillOval(150, 50, 100, 100); // x, y, width, height

g.setColor(Color.BLACK); // Reset color to black

g.drawString("Filled Circle", 180, 170); // Label

}

public static void main(String[] args) { JFrame frame = new JFrame("Circle Applet");

CircleApplet applet = new CircleApplet();

// Initialize the applet (same as if it were run in a browser) applet.init();

applet.start();

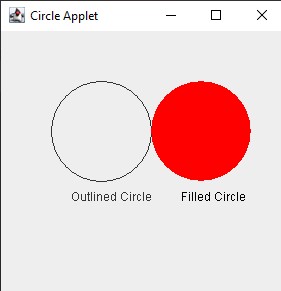
// Add the applet to the frame frame.add(applet); frame.setSize(300, 300);

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

frame.setVisible(true);

}

}



Exp-14

//Syed Tabaraq

//231P054,46

//Write a program to find square of a given number using applet. import java.awt.\*; import java.awt.event.\*; import javax.swing.\*;

public class SquareApplet extends JApplet implements ActionListener {

// Declaring components private Label label; private TextField inputField; private Button squareButton;

private String result = "";

@Override

public void init() {

// Setting layout and adding components setLayout(new FlowLayout());

label = new Label("Enter a number: ");

add(label);

inputField = new TextField(10);

add(inputField);

squareButton = new Button("Find Square");

add(squareButton);

// Adding action listener to the button squareButton.addActionListener(this);

}

// Action performed when the button is clicked

@Override

public void actionPerformed(ActionEvent e) { try {

// Get the input from the text field and calculate the square int number = Integer.parseInt(inputField.getText()); int square = number \* number;

result = "Square of " + number + " is " + square;

} catch (NumberFormatException ex) {

// Handle invalid input

result = "Please enter a valid number.";

}

// Repaint the applet to display the result

repaint();

}

@Override

public void paint(Graphics g) {

// Display the result

g.drawString(result, 50, 150);

}

public static void main(String[] args) { // Create a JFrame to hold the applet

JFrame frame = new JFrame("Square Applet");

SquareApplet applet = new SquareApplet();

// Initialize the applet (same as if it were run in a browser) applet.init(); applet.start();

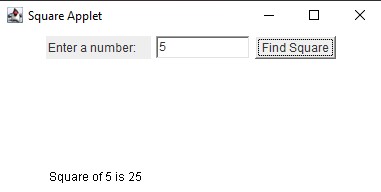
// Add the applet to the frame frame.add(applet); frame.setSize(400, 200);

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

frame.setVisible(true);

}

}



//Khan Arman //231P072,13 import javax.swing.\*; import java.awt.\*;

import java.awt.event.ActionEvent; import java.awt.event.ActionListener;

public class Calculator implements ActionListener {

JFrame frame;

JTextField display;

JButton[] numberButtons;

JButton addButton, subButton, mulButton, divButton, equButton, delButton, clrButton; JPanel panel;

double num1 = 0, num2 = 0, result = 0; char operator;

public Calculator() { // Frame settings

frame = new JFrame("Calculator"); frame.setSize(400, 600);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); frame.setLayout(null);

// Display panel display = new JTextField(); display.setBounds(50, 25, 300, 50);

display.setFont(new Font("Arial", Font.BOLD, 24)); // Increased font size display.setBackground(Color.LIGHT\_GRAY); // Background color display.setEditable(false);

frame.add(display);

// Number buttons

numberButtons = new JButton[10];

for (int i = 0; i < 10; i++) {

numberButtons[i] = new JButton(String.valueOf(i)); numberButtons[i].setFont(new Font("Arial", Font.BOLD, 24)); numberButtons[i].addActionListener(this);

}

// Operation buttons addButton = new JButton("+"); subButton = new JButton("-"); mulButton = new JButton("\*"); divButton = new JButton("/"); equButton = new JButton("="); delButton = new JButton("Delete"); clrButton = new JButton("Clear");

// Set button fonts

addButton.setFont(new Font("Arial", Font.BOLD, 24)); subButton.setFont(new Font("Arial", Font.BOLD, 24)); mulButton.setFont(new Font("Arial", Font.BOLD, 24)); divButton.setFont(new Font("Arial", Font.BOLD, 24)); equButton.setFont(new Font("Arial", Font.BOLD, 24)); delButton.setFont(new Font("Arial", Font.BOLD, 24));

clrButton.setFont(new Font("Arial", Font.BOLD, 24));

// Add action listeners to operation buttons addButton.addActionListener(this); subButton.addActionListener(this); mulButton.addActionListener(this); divButton.addActionListener(this); equButton.addActionListener(this); delButton.addActionListener(this); clrButton.addActionListener(this);

// Panel settings panel = new JPanel(); panel.setBounds(50, 100, 300, 400);

panel.setLayout(new GridLayout(4, 4, 10, 10)); // Adjusted gaps panel.add(numberButtons[1]); panel.add(numberButtons[2]); panel.add(numberButtons[3]); panel.add(addButton); panel.add(numberButtons[4]); panel.add(numberButtons[5]); panel.add(numberButtons[6]);

panel.add(subButton); panel.add(numberButtons[7]); panel.add(numberButtons[8]); panel.add(numberButtons[9]); panel.add(mulButton); panel.add(clrButton); panel.add(numberButtons[0]); panel.add(delButton); panel.add(equButton);

panel.add(divButton);

frame.add(panel); frame.setVisible(true);

}

public void actionPerformed(ActionEvent e) {

for (int i = 0; i < 10; i++) { if (e.getSource() == numberButtons[i]) {

display.setText(display.getText().concat(String.valueOf(i)));

}

}

// Check if display is not empty before parsing if (e.getSource() == addButton) { if (!display.getText().isEmpty()) { num1 = Double.parseDouble(display.getText()); operator = '+';

display.setText("");

}

}

if (e.getSource() == subButton) { if (!display.getText().isEmpty()) { num1 = Double.parseDouble(display.getText()); operator = '-'; display.setText("");

}

}

if (e.getSource() == mulButton) { if (!display.getText().isEmpty()) { num1 = Double.parseDouble(display.getText()); operator = '\*';

display.setText("");

}

}

if (e.getSource() == divButton) { if (!display.getText().isEmpty()) { num1 = Double.parseDouble(display.getText()); operator = '/';

display.setText("");

}

}

if (e.getSource() == equButton) { if (!display.getText().isEmpty()) { num2 = Double.parseDouble(display.getText());

switch (operator) { case '+':

result = num1 + num2; break; case '-':

result = num1 - num2;

break; case '\*':

result = num1 \* num2; break; case '/':

if (num2 != 0) { // Prevent division by zero result = num1 / num2;

} else {

display.setText("Error");

return;

} break;

}

display.setText(String.valueOf(result));

num1 = result;

}

}

if (e.getSource() == delButton) { String str = display.getText();

display.setText(str.length() > 0 ? str.substring(0, str.length() - 1) : "");

}

if (e.getSource() == clrButton) { display.setText(""); num1 = num2 = result = 0;

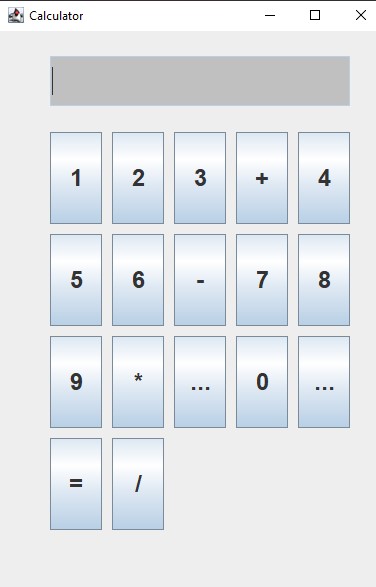
}

}

public static void main(String[] args) { new Calculator();

}

}



//Khan Arman

//231P072,13

//Write an applet using ActionEvent and textfield to find largest between three numbers.

import javax.swing.\*; import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class LargestNumberCalculator extends JFrame implements ActionListener {

private JTextField num1Field, num2Field, num3Field, resultField; private JButton findLargestButton;

public LargestNumberCalculator() {

// Set up the frame

setTitle("Largest Number Calculator");

setSize(300, 200);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new FlowLayout());

// Create text fields num1Field = new JTextField(10); num2Field = new JTextField(10); num3Field = new JTextField(10); resultField = new JTextField(10); resultField.setEditable(false); // Result field is not editable

// Create button

findLargestButton = new JButton("Find Largest"); findLargestButton.addActionListener(this);

// Add components to the frame add(new JLabel("Number 1:")); add(num1Field); add(new JLabel("Number 2:")); add(num2Field); add(new JLabel("Number 3:")); add(num3Field); add(findLargestButton); add(new JLabel("Largest Number:"));

add(resultField);

}

@Override

public void actionPerformed(ActionEvent e) {

// Get numbers from text fields

try {

double num1 = Double.parseDouble(num1Field.getText()); double num2 = Double.parseDouble(num2Field.getText()); double num3 = Double.parseDouble(num3Field.getText());

// Find the largest number

double largest = Math.max(num1, Math.max(num2, num3));

resultField.setText(String.valueOf(largest)); } catch (NumberFormatException ex) { resultField.setText("Invalid input");

}

}

public static void main(String[] args) {

// Create the GUI

SwingUtilities.invokeLater(() -> {

LargestNumberCalculator calculator = new LargestNumberCalculator(); calculator.setVisible(true);

});

}

}

