**WEB PROG**

Team Leader: Getizo, Gabriel Franco

Members:

Espinase, Kristian Anthony

Geguna, Arvin Joseph

Magnaye, Sheera Mae

Tuquib, Jaela Mae

**JAVASWING**

**Descripton:**  
 Java Swing tutorial is a part of Java Foundation Classes (JFC) that is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

WHAT IS JFC?

The Java Foundation Classes (JFC) are a set of GUI components which simplify the development of desktop applications.

**Hieararchy:**

The hierarchy of java swing API

**Commonly used Methods of Component class:**

The methods of Component class are widely used in java swing that are given

**Sample program**

**Reference:**

**Imports:**

javax.swing.\*: For GUI components like JFrame, JButton, JLabel, etc.

java.awt.\*: For basic windowing and layout management, such as FlowLayout and Font.

java.awt.event.\*: For event handling, like ActionListener.

Class Declaration:

The class SwingComponents extends JFrame (to create a window) and implements ActionListener (to handle button click events).

Variables:

FlowLayout fl: Defines the layout style of the frame, where components are added in a left-to-right flow.

Font bFont: Sets a bold font for the text displayed.

JLabel lbl1: A label prompting the user to enter their name.

JTextField tf: A text field where the user can input their name.

JButton bt: A button labeled “Click me” that the user can click to trigger the event.

JLabel lbl2: A label that will display the greeting message after the button is clicked.

Constructor SwingComponents:

super(“Sample Event”): Sets the title of the JFrame window to “Sample Event.”

The frame size is set using setSize(275, 225).

The layout is set to FlowLayout using setLayout(fl).

Components (lbl1, tf, bt, lbl2) are added to the frame using the add() method.

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE) ensures that the application will close when the window is closed.

The button bt is registered with an action listener (this) so that it can respond to click events.

Action Event Handling (actionPerformed method):

When the button is clicked, the actionPerformed method is called.

It retrieves the text entered in the text field tf using getText().

A greeting message is constructed (Hi, <name>!) and set as the text of lbl2 using setText().

Sample Code:  
package javaawt;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class JavaAWT extends JFrame implements ActionListener {

    JFrame inputFrame = new JFrame();

    JFrame outputFrame = new JFrame();

    JPanel inputPanel = new JPanel();

    JPanel outputPanel = new JPanel();

    JTextField inputField = new JTextField(20);

    JLabel inputLabel = new JLabel("Enter Host Address: ");

    JLabel outputIp = new JLabel();

    JTextArea outputArea = new JTextArea(5, 15);

    JButton btnFindIp = new JButton("Find IP");

    JButton btnClear = new JButton("Clear");

    JButton btnOkay = new JButton("Okay");

    FlowLayout f1 = new FlowLayout();

    Font setFont = new Font("", Font.BOLD, 14);

    public JavaAWT() {

        this.setSize(280, 150);

        this.setLocation(200, 200);

        this.setTitle("INPUT");

        this.setResizable(true);

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

        inputFrame.add(inputPanel);

        inputPanel.add(inputLabel);

        inputPanel.add(inputField);

        inputPanel.add(btnFindIp);

        btnFindIp.addActionListener(new btnFindIp());

        inputPanel.add(btnClear);

        btnClear.addActionListener(new btnClear());

        this.add(inputPanel);

        this.setVisible(true);

    }

    @Override

    public void actionPerformed(ActionEvent e) {

        throw new UnsupportedOperationException();

    }

    class btnFindIp implements ActionListener {

        @Override

        public void actionPerformed(ActionEvent a) {

            try {

                String host = inputField.getText();

                String ip = java.net.InetAddress.getByName(host).getHostAddress();

                outputArea.append("Host Address: \n " + host + "\n\nIP Address: \n " + ip + "\n");

            } catch (Exception ex) {

                System.out.println(ex);

            }

            outputFrame.setSize(230, 185);

            outputFrame.setLocation(430, 240);

            outputFrame.setTitle("OUTPUT");

            outputFrame.setResizable(false);

            outputFrame.setVisible(true);

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

            outputArea.setEditable(false);

            outputArea.setBackground(Color.LIGHT\_GRAY);

            outputArea.setFont(setFont);

            outputFrame.add(outputPanel);

            outputPanel.add(outputArea);

            outputPanel.add(outputIp);

            outputPanel.add(btnOkay);

            btnFindIp.setEnabled(false);

            btnOkay.addActionListener(new btnOkay());

        }

    }

    class btnClear implements ActionListener {

        @Override

        public void actionPerformed(ActionEvent e) {

            inputField.setText("");

            outputArea.setText("");

            btnFindIp.setEnabled(true);

            outputFrame.dispose();

        }

    }

    class btnOkay implements ActionListener {

        @Override

        public void actionPerformed(ActionEvent i) {

            inputField.setText("");

            outputArea.setText("");

            btnFindIp.setEnabled(true);

            outputFrame.dispose();

        }

    }

    public static void main(String[] args) {

        JavaAWT javaAwt = new JavaAWT();

    }

}

Output:

Before clicking the button: The window prompts “Enter your name” and shows a button.

After clicking the button: It displays the greeting, like “Hi, Ranika!” based on the text entered.

Summary:

Class Inheritance: extends JFrame allows creating a window.

Interfaces: implements ActionListener handles user actions (button click).

Swing Components: JLabel, JTextField, JButton are used for the user interface.

Layout Management: FlowLayout arranges components in the window.

Event Handling: actionPerformed() is implemented to define what happens when the button is clicked.