



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 11
Implement a program on Applet or AWT Controls
Date of Performance:
Date of Submission:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Aim: Implement a program on Applet or AWT Controls

Objective:

To develop application like Calculator, Games, Animation using AWT Controls.

Theory:

Java AWT (Abstract Window Toolkit) is an API to develop Graphical User Interface (GUI) or windows-based applications in Java.

Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavy weight i.e. its components are using the resources of underlying operating system (OS). The java.awt package provides classes for AWT API such as TextField, Label, TextArea,

RadioButton, CheckBox, Choice, List etc.

1. A general interface between Java and the native system, used for windowing, events and layout managers. This API is at the core of Java GUI programming and is also used by Swing and Java 2D. It contains the interface between the native windowing system and the Java application1.
2. A basic set of GUI widgets such as buttons, text boxes, and menus1. AWT also provides Graphics and imaging tools, such as shape, color, and font classes2. AWT also avails layout managers which helps in increasing the flexibility of the window layouts2

Java AWT calls the native platform calls the native platform (operating systems) subroutine for creating API components like TextField, ChechBox, button, etc.

For example, an AWT GUI with components like TextField, label and button will have different look and feel for the different platforms like Windows, MAC OS, and Unix. The reason for this is the platforms have different view for their native components and AWT directly calls the native subroutine that creates those components.

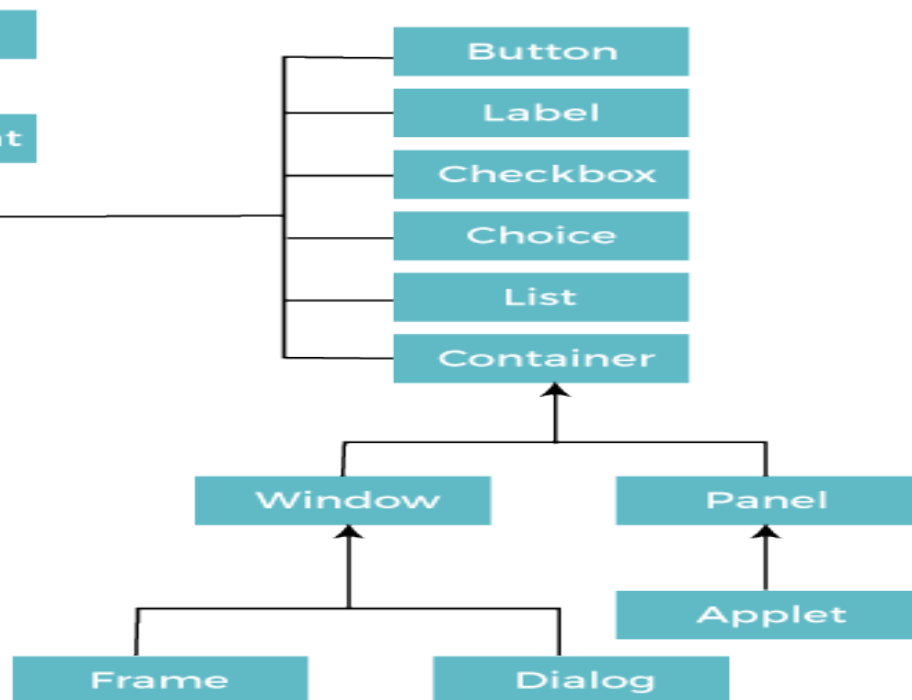


Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

In simple words, an AWT application will look like a windows application in Windows OS whereas it will look like a Mac application in the MAC OS.

Java AWT Hierarchy



Code:

```
import java.awt.*;
import java.awt.event.*;

public class AWTCalculator extends Frame implements ActionListener {
    // Define AWT components
    TextField display;
    Panel panel;
    String[] buttonLabels = { "7", "8", "9", "/",
                              "4", "5", "6", "*",
                              "1", "2", "3", "-",
                              "0", "C", "=", "+" };
    Button[] buttons = new Button[16];

    String currentText = "";
    double num1, num2, result;
    char operator;
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

```
// Constructor
public AWTCalculator() {
    // Set up the frame
    setTitle("AWT Calculator");
    setSize(300, 400);
    setLayout(new BorderLayout());

    // Display field
    display = new TextField();
    display.setEditable(false);
    add(display, BorderLayout.NORTH);

    // Panel for buttons with GridLayout (4x4)
    panel = new Panel();
    panel.setLayout(new GridLayout(4, 4, 10, 10));

    // Initialize buttons and add them to the panel
    for (int i = 0; i < 16; i++) {
        buttons[i] = new Button(buttonLabels[i]);
        buttons[i].addActionListener(this);
        panel.add(buttons[i]);
    }

    // Add panel to the frame
    add(panel, BorderLayout.CENTER);

    // Handle window close event
    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent we) {
            System.exit(0);
        }
    });

    setVisible(true);
}
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

```
// Handle button click events
public void actionPerformed(ActionEvent e) {
    String command = e.getActionCommand();

    // If it's a number or operator, add to current input
    if (command.charAt(0) >= '0' && command.charAt(0) <= '9') {
        currentText += command;
        display.setText(currentText);
    } else if (command.equals("+") || command.equals("-") || command.equals("*") || command.equals("/")) {
        num1 = Double.parseDouble(currentText);
        operator = command.charAt(0);
        currentText = "";
    } else if (command.equals("=")) {
        num2 = Double.parseDouble(currentText);
        result = performOperation(num1, num2, operator);
        display.setText(String.valueOf(result));
        currentText = "";
    } else if (command.equals("C")) {
        display.setText("");
        currentText = "";
    }
}

// Perform arithmetic operations
public double performOperation(double n1, double n2, char operator) {
    switch (operator) {
        case '+': return n1 + n2;
        case '-': return n1 - n2;
        case '*': return n1 * n2;
        case '/': return n1 / n2;
        default: return 0;
    }
}

// Main method to run the calculator
public static void main(String[] args) {
    new AWTCalculator();
}
```

Conclusion:

While AWT provides a foundational understanding of GUI programming in Java, developers today are more likely to use Swing or JavaFX for new applications due to their enhanced capabilities and flexibility. However, AWT remains a useful tool for specific scenarios, particularly in learning contexts or for lightweight applications.



Vidyavardhini's College of Engineering and Technology
Department of Artificial Intelligence & Data Science