

Java Programming Language SE - 6

Module 12: Building Java GUIs Using the Swing API

Team Emertxe

ORACLE®

Certified Professional

Java SE 6 Programmer



Objectives

- Describe the JFC Swing technology
- Define Swing
- Identify the Swing packages
- Describe the GUI building blocks: containers, components, and layout managers
- Examine top-level, general-purpose, and special-purpose properties of container
- Examine components
- Examine layout managers
- Describe the Swing single-threaded model
- Build a GUI using Swing components



What Are the Java Foundation Classes (JFC)?

Java Foundation Classes are a set of Graphical User Interface (GUI) support packages, including:

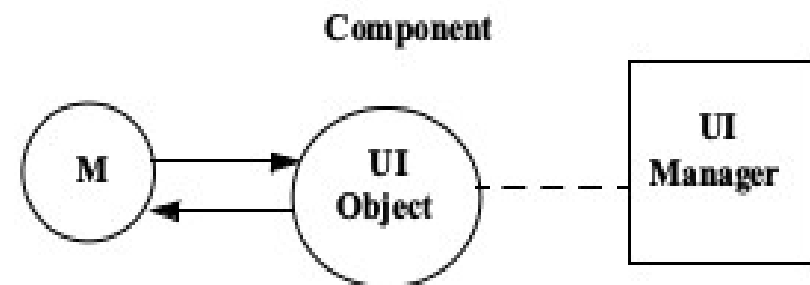
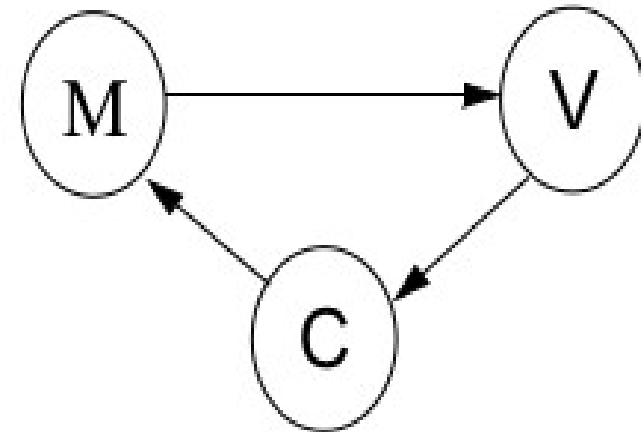
- Abstract Window Toolkit (AWT)
- The Swing component set
- 2D graphics
- Pluggable look-and-feel
- Accessibility
- Drag-and-drop
- Internationalization

What Is Swing?

- An enhanced GUI component set
- Provides replacement components for those in the original AWT
- Has special features, such as a pluggable look-and feel

Swing Architecture

- Has its roots in the Model-View-Controller (MVC) architecture
- The Swing components follow Separable Model Architecture



Swing Packages

- Package Name
 - javax.swing
 - javax.swing.border
 - javax.swing.event
 - javax.swing.undo
 - javax.swing.plaf
 - javax.swing.plaf.basic
 - javax.swing.plaf.metal
 - javax.swing.plaf.multi
 - javax.swing.plaf.synth

Swing Packages

- Package Name
 - javax.swing.colorchooser
 - javax.swing.filechooser
 - javax.swing.table
 - javax.swing.tree
 - javax.swing.text
 - javax.swing.text.html
 - javax.swing.text.html.parser
 - javax.swing.text.rtf
 - javax.swing.undo

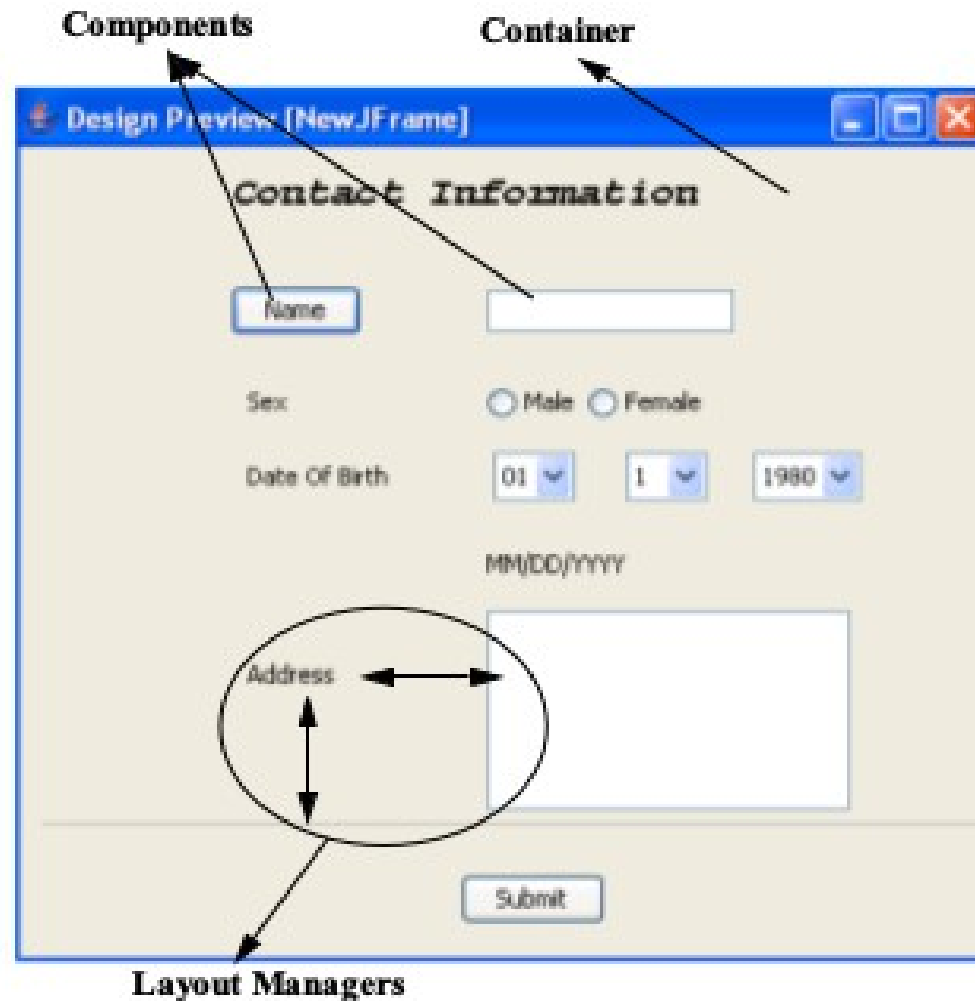
Examining the Composition of a Java Technology GUI



A Swing API-based GUI is composed of the following elements:

- Containers - Are on top of the GUI containment hierarchy.
- Components - Contain all the GUI components that are derived from the JComponent class.
- Layout Managers - Are responsible for laying out components in a container.

Examining the Composition of a Java Technology GUI

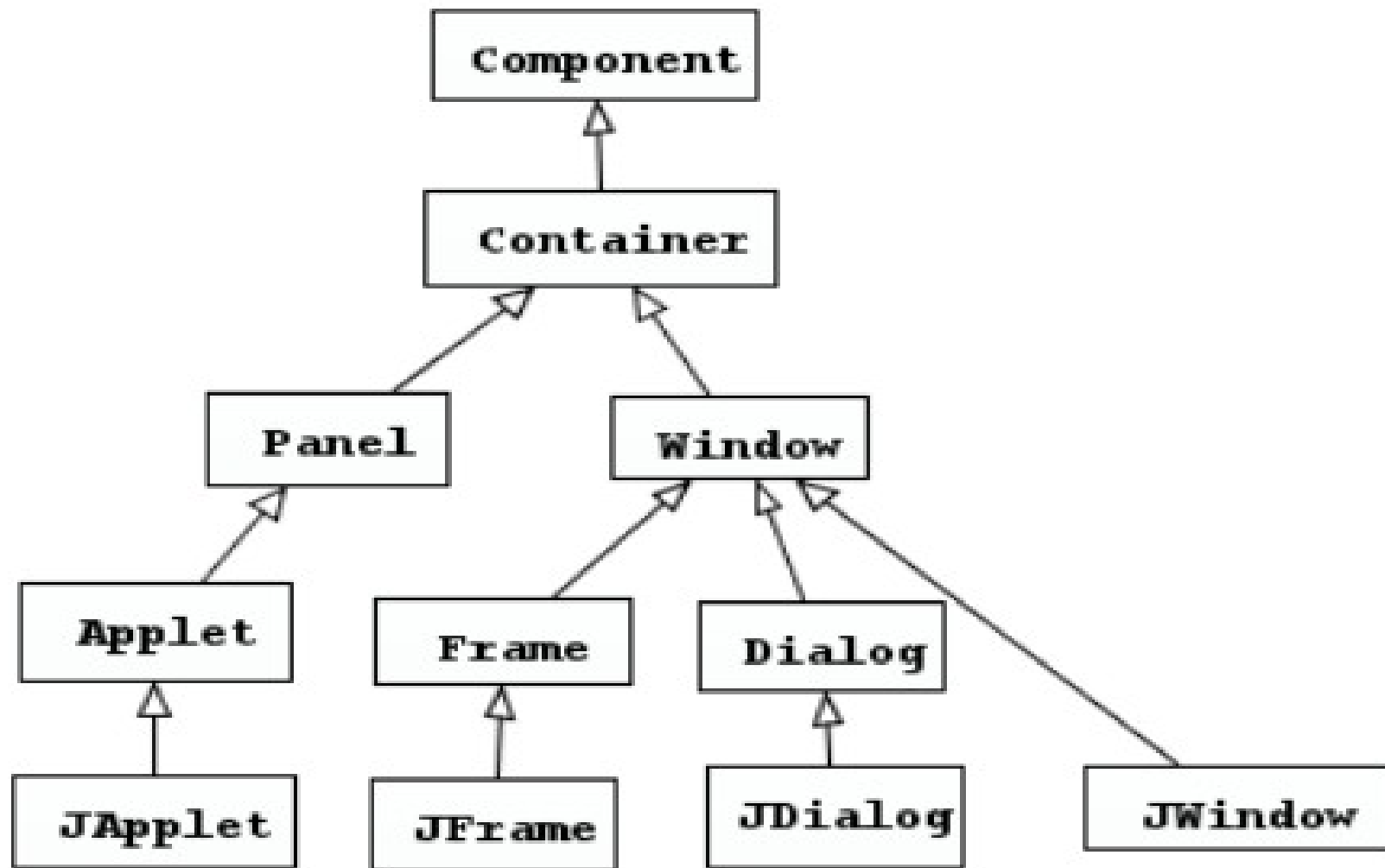


Swing Containers

Swing containers can be classified into three main categories:

- Top-level containers:
 - JFrame, JWindow, and JDialog
- General-purpose containers:
 - JPanel, JScrollPane, JToolBar, JSplitPane, and JTabbedPane
- Special-purpose containers:
 - JInternalFrame and JLayeredPane

Top-Level Containers



Swing Components

Swing components can be broadly classified as:

- Buttons
- Text components
- Uneditable information display components
- Menus
- Formatted display components
- Other basic controls

Text Components

Swing text components can be broadly divided into three categories.

- Text controls - JTextField, JPasswordField (for user input)
- Plain text areas - JTextArea (displays text in plain text, also for multi-line user input)
- Styled text areas - JEditorPane, JTextPane (displays formatted text)

Swing Component Properties



Common component properties:

- All the Swing components share some common properties because they all extend JComponent.
- Component-specific properties:
 - Each component defines more specific properties.

Common Component Properties

| Property | Methods |
|---------------------------------|--|
| Border | <code>Border getBorder()</code> <code>void setBorder(Border b)</code> |
| Background and foreground color | <code>void setBackground(Color bg)</code> <code>void setForeground(Color bg)</code> |
| Font | <code>void setFont(Font f)</code> |
| Opaque | <code>void setOpaque(boolean isOpaque)</code> |
| Maximum and minimum size | <code>void setMaximumSize(Dimension d)</code> <code>void setMinimumSize(Dimension d)</code> |
| Alignment | <code>void setAlignmentX(float ax)</code> <code>void setAlignmentY(float ay)</code> |
| Preferred size | <code>void setPreferredSize(Dimension ps)</code> |

Component-Specific Properties



Properties

Maximum row count

Model

Selected index

Selected Item

Item count

Renderer

Editable

Methods

`void setMaximumRowCount(int count)`

`void setModal(ComboBoxModel cbm)`

`int getSelectedIndex()`

`Object getSelectedItem()`

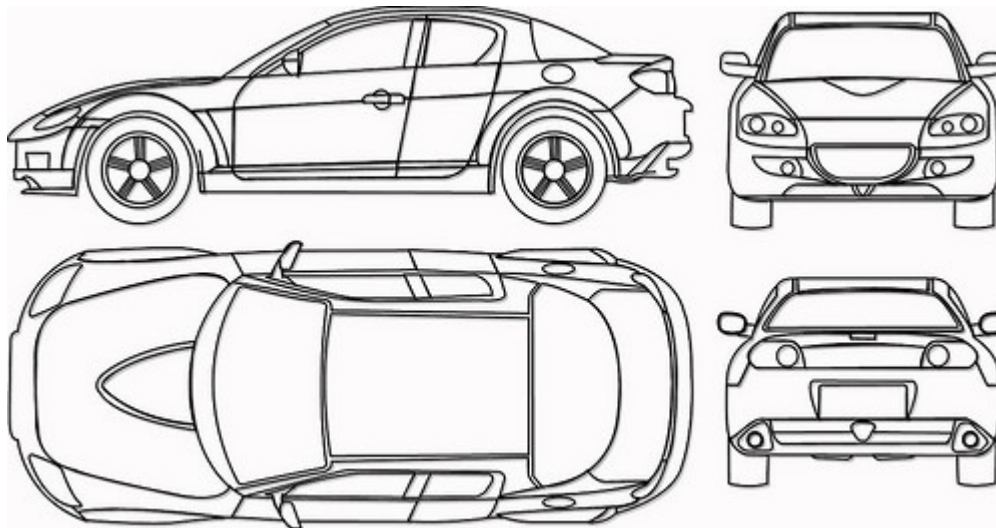
`int getItemCount()`

`void setRenderer(ListCellRenderer ar)`

`void setEditable(boolean flag)`

Layout Managers

- Handle problems caused by:
 - GUI resizing by user
 - Operating system differences in fonts
 - Locale-specific text layout requirements



Layout Managers

- Layout manager classes:
 - BorderLayout
 - FlowLayout
 - BoxLayout
 - CardLayout
 - GridLayout
 - GridBagLayout

The BorderLayout Manager

The BorderLayout manager places components in top, bottom, left, right and center locations.



BorderLayout Example

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

public class BorderExample {
    private JFrame f;
    private JButton bn, bs, bw, be, bc;
    public BorderExample() {
        f = new JFrame("Border Layout");
        bn = new JButton("Button 1");
        bc = new JButton("Button 2");
        bw = new JButton("Button 3");
        bs = new JButton("Button 4");
        be = new JButton("Button 5");}
}
```

BorderLayout Example

```
public void launchFrame() {  
    f.add(bn, BorderLayout.NORTH);  
    f.add(bs, BorderLayout.SOUTH);  
    f.add(bw, BorderLayout.WEST);  
    f.add(be, BorderLayout.EAST);  
    f.add(bc, BorderLayout.CENTER);  
    f.setSize(400,200);  
    f.setVisible(true);  
}  
  
public static void main(String args[]) {  
    BorderExample guiWindow2 = new BorderExample();  
    guiWindow2.launchFrame();  
}}
```

The FlowLayout Manager

- The FlowLayout manager places components in a row, and if the row fills, components are placed in the next row.



FlowLayout Example

```
public class LayoutExample {  
    private JFrame f;  
    private JButton b1;  
    private JButton b2;  
    private JButton b3;  
    private JButton b4;  
    private JButton b5;  
    public LayoutExample() {  
        f = new JFrame("GUI example");  
        b1 = new JButton("Button 1");  
        b2 = new JButton("Button 2");  
        b3 = new JButton("Button 3");  
        b4 = new JButton("Button 4");  
        b5 = new JButton("Button 5");  
    }  
}
```

FlowLayout Example

```
public void launchFrame() {  
    f.setLayout(new FlowLayout());  
    f.add(b1);  
    f.add(b2);  
    f.add(b3);  
    f.add(b4);  
    f.add(b5);  
    f.pack();  
    f.setVisible(true);  
}  
  
public static void main(String args[]) {  
    LayoutExample guiWindow = new LayoutExample();  
    guiWindow.launchFrame();  
}}
```

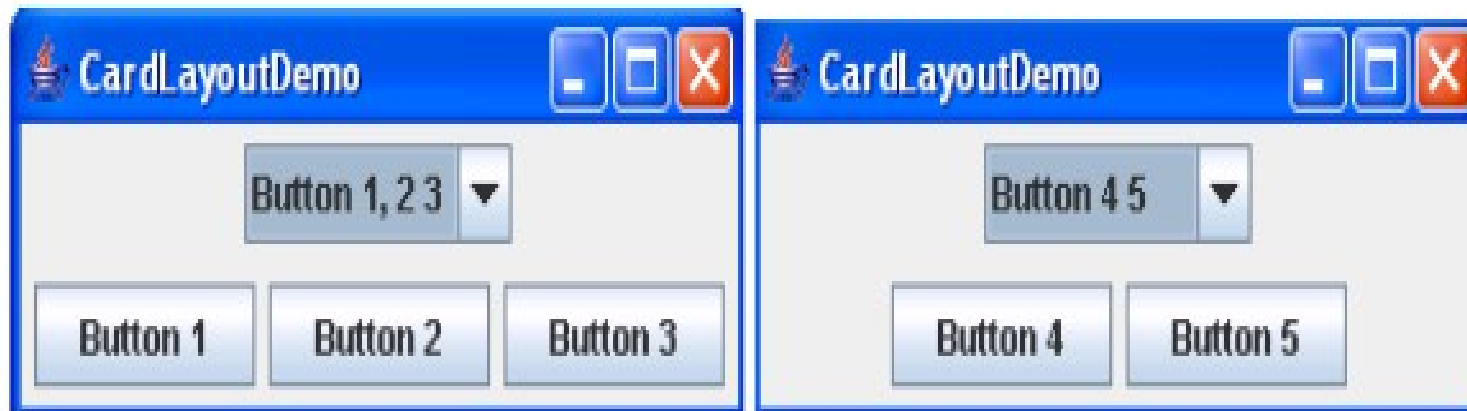

The BoxLayout Manager

- The BoxLayout manager add components from left to right, and from top to bottom in a single row of column.



The CardLayout Manager

- The CardLayout manager places the components in different cards. Cards are usually controlled by a combo box.



The GridLayout Manager

- The GridLayout manager places components in rows and columns in the form of a grid.



GridLayout Example

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

public class GridExample {
    private JFrame f;
    private JButton b1, b2, b3, b4, b5;
    public GridExample() {
        f = new JFrame("Grid Example");
        b1 = new JButton("Button 1");
        b2 = new JButton("Button 2");
        b3 = new JButton("Button 3");
        b4 = new JButton("Button 4");
        b5 = new JButton("Button 5");}
}
```

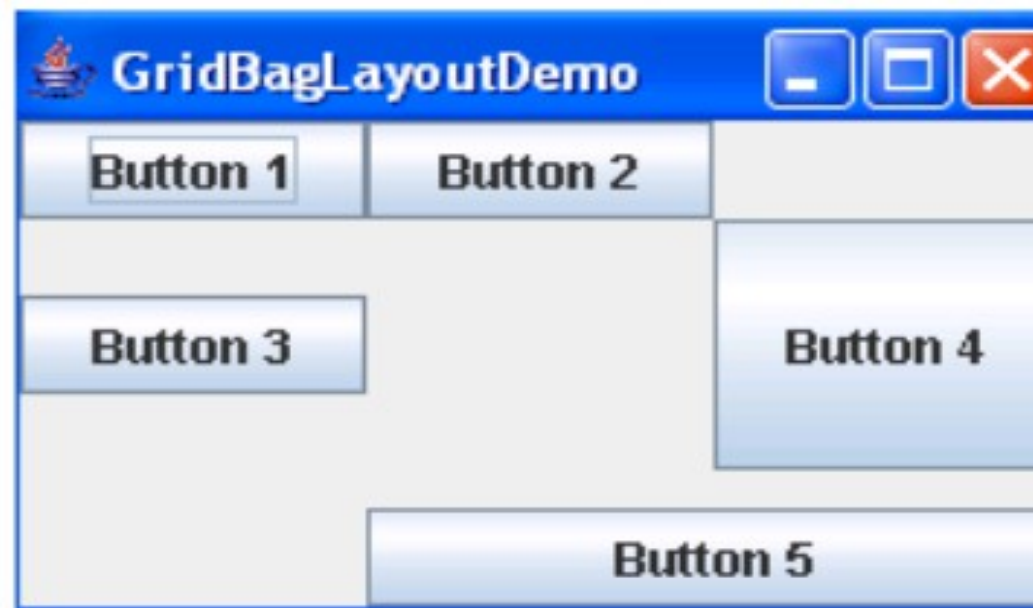
GridLayout Example

```
public void launchFrame() {  
    f.setLayout (new GridLayout(3,2));  
    f.add(b1);  
    f.add(b2);  
    f.add(b3);  
    f.add(b4);  
    f.add(b5);  
    f.pack();  
    f.setVisible(true);}  
public static void main(String args[]) {  
    GridExample grid = new GridExample();  
    grid.launchFrame();  
}}
```

The GridBagLayout Manager



- The GridBagLayout manager arranges components in rows and columns, similar to a grid layout, but provides a wide variety of options for resizing and positioning the components.



GUI Construction

- Programmatic
- GUI builder tool



Programmatic Construction

```
import javax.swing.*;

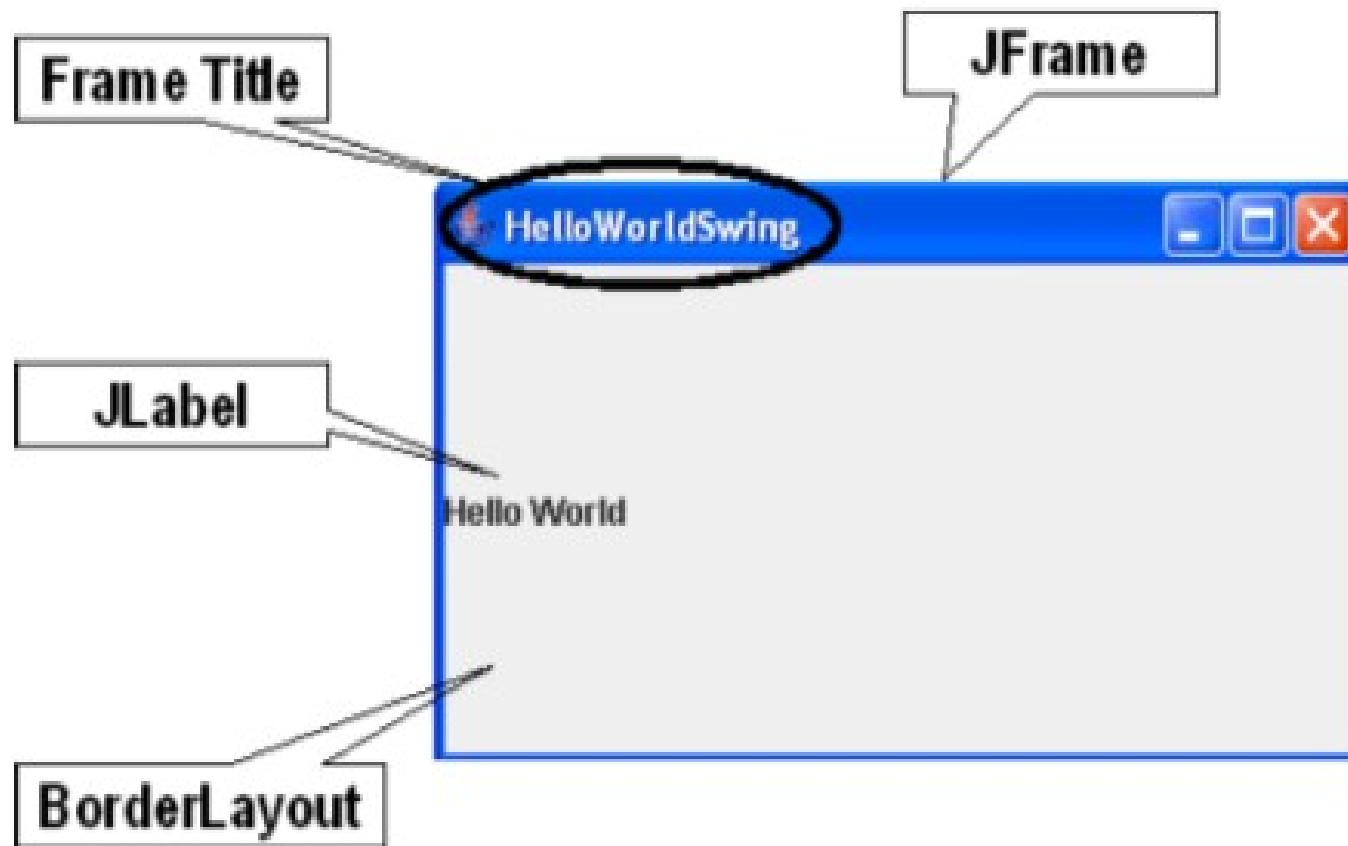
public class HelloWorldSwing {

    private static void createAndShowGUI() {
        JFrame frame = new JFrame("HelloWorldSwing");
        //Set up the window.
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel label = new JLabel("Hello World");
        // Add Label
        frame.add(label);
        frame.setSize(300,200);
        // Display Window
        frame.setVisible(true);}
}
```


Programmatic Construction

```
public static void main(String[] args) {  
    javax.swing.SwingUtilities.invokeLater(new Runnable() {  
        //Schedule for the event-dispatching thread:  
        //creating,showing this app's GUI.  
        public void run() {createAndShowGUI();}  
    });  
}
```

Programmatic Construction



Key Methods

Methods for setting up the JFrame and adding JLabel:

- `setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)`
-Creates the program to exit when the close button is clicked.
- `setVisible(true)`- Makes the JFrame visible.
- `add(label)`- JLabel is added to the content pane not to the JFrame directly.



Key Methods

- Tasks:
 - Executing GUI application code, such as rendering
 - Handling GUI events
 - Handling time consuming (background) processes
- The SwingUtilities class:
 - `SwingUtilities.invokeLater(new Runnable())`

Stay connected

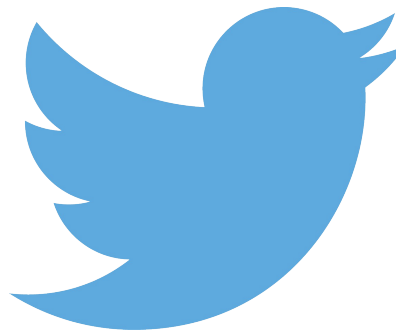


About us: Emertxe is India's one of the top IT finishing schools & self learning kits provider. Our primary focus is on Embedded with diversification focus on Java, Oracle and Android areas

Emertxe Information Technologies,
No-1, 9th Cross, 5th Main,
Jayamahal Extension,
Bangalore, Karnataka 560046
T: +91 80 6562 9666
E: training@emertxe.com



<https://www.facebook.com/Emertxe>



<https://twitter.com/EmertxeTweet>



slideshare
Present Yourself

<https://www.slideshare.net/EmertxeSlides>



Thank You