

Graphical I/O



Overview of topics related to Java's graphical I/O

- > Multiple toolkits exist to support a graphical user interface (GUI)
 - Abstract Window Toolkit (AWT)
 - Swing Toolkit
 - JavaFX Toolkit

- > Two types of GUI programs:
 - Application programs
 - Applets (short programs, typically run within a browser)

Swing GUI consists of:

Components

Containers

Containers can also be components

Components are derived from JComponent class

- JComponent provides functionality to all components
- JC inherits AWT classes Container & Component
- All of Swings components are represented by classes defined within the package javax.swing

Swing Components

JApplet	<u>JButton</u>	<u>JCheckBox</u>	JCheckBoxMenuItem
JColorChooser	JComboBox	JComponent	JDesktopPane
JDialog	JEditorPane	JFileChooser	JFormattedTextField
<u>JFrame</u>	JInternalFrame	<u>JLabel</u>	JLayer
JLayeredPane	<u>JList</u>	JMenu	JMenuBar
JMenuItem	JOptionPane	JPanel	JPasswordField
JPopupMenu	JProgressBar	JRadioButton	JRadioButtonMenuItem
JRootPane	JScrollBar	JScrollPane	JSeparator
JSlider	JSpinner	JSplitPane	JTabbedPane
JTable	JTextArea	<u>JTextField</u>	JTextPane
JToggleButton	JToolBar	JToolTip	JTree
JViewport	JWindow		

Swing defines two types of containers

- > Top-level containers
 - JFrame
 - JApplet
 - JWindow
 - JDialog

Top level containers do not inherit JComponent
They inherit AWT classes Component & Container
They are thus “heavyweight”

- > Lightweight containers
 - Do inherit JComponent

Swing Layout Managers

Control position of components within a container

FlowLayout	A simple layout that positions components left-to-right, top-to-bottom. (Positions components right-to-left for some cultural settings.)
BorderLayout	Positions components within the center or the borders of the container. This is the default layout for a content pane.
GridLayout	Lays out components within a grid.
GridBagLayout	Lays out different size components within a flexible grid.
BoxLayout	Lays out components vertically or horizontally within a box.
SpringLayout	Lays out components subject to a set of constraints.

A simple first example

```
import javax.swing.*;

class SwingDemo {

    SwingDemo() {

        // Create a new JFrame container.
        JFrame jfrm = new JFrame("A Simple Swing Application");

        // Give the frame an initial size.
        jfrm.setSize(275, 100);

        // Terminate the program when the user closes the application.
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create a text-based label.
        JLabel jlab = new JLabel(" Swing defines the modern Java GUI.");

        // Add the label to the content pane.
        jfrm.add(jlab);

        // Display the frame.
        jfrm.setVisible(true);
    }

    public static void main(String args[]) {
        // Create the frame on the event dispatching thread.
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new SwingDemo();
            }
        });
    }
}
```

```
class SwingDemo {  
  
    SwingDemo() {                // a constructor  
  
        JFrame jfrm = new JFrame("A Simple Swing Application");  
  
        jfrm.setSize(275, 100);  
  
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        JLabel jlab = new JLabel(" Swing defines the modern Java GUI.");  
  
        jfrm.add(jlab);  
  
        jfrm.setVisible(true);  
    }  
  
    public static void main(String args[]) {  
        // Create the frame on the event dispatching thread.  
        SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                new SwingDemo();  
            }  
        });  
    }  
}
```



```
import javax.swing.*;

class SwingDemo {

    SwingDemo() {                // this is a constructor

        // Create a new JFrame container.

        // Give the frame an initial size.

        // Terminate the program when the user closes the application.

        // Create a text-based label.

        // Add the label to the content pane.

        // Display the frame.
    }
}
```

```
import javax.swing.*;

class SwingDemo {

    SwingDemo() {                // this is a constructor

        // Create a new JFrame container.
        JFrame jfrm = new JFrame("A Simple Swing Application");

        // Give the frame an initial size.
        jfrm.setSize(275, 100);

        // Terminate the program when the user closes the application.
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        .....
    }
}
```

```
import javax.swing.*;

class SwingDemo {

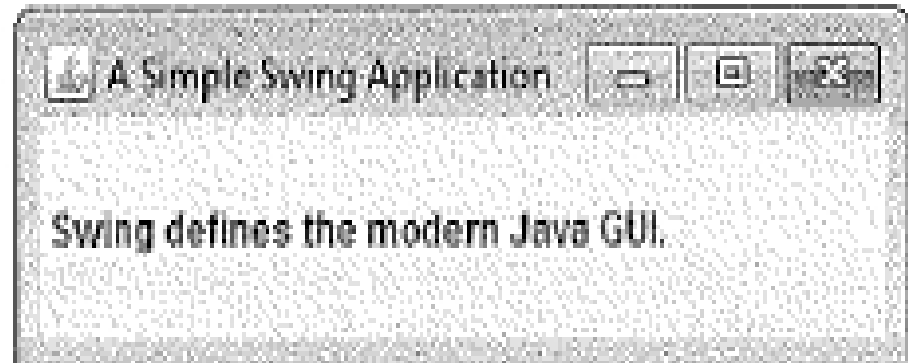
    SwingDemo() {                // a constructor

        . . . (continuing)

        // Create a text-based label.
        JLabel jlab = new JLabel(" Swing defines the modern Java GUI.");

        // Add the label to the content pane.
        jfrm.add(jlab);

        // Display the frame.
        jfrm.setVisible(true);
    }
}
```



The main routine

```
class SwingDemo {  
  
    // Constructor goes here  
  
    public static void main(String args[]) {  
        // Create the frame on the event dispatching thread.  
        SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                new SwingDemo();  
            }  
        });  
    }  
}
```

```
class SwingDemo {

    SwingDemo() {

        // Create a new JFrame container.
        JFrame jfrm = new JFrame("A Simple Swing Application");

        // Give the frame an initial size.
        jfrm.setSize(275, 100);

        // Terminate the program when the user closes the application.
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create a text-based label.
        JLabel jlab = new JLabel(" Swing defines the modern Java GUI.");

        // Add the label to the content pane.
        jfrm.add(jlab);

        // Display the frame.
        jfrm.setVisible(true);
    }
}
```