# swing

PresentationPoint

## swing

- swing is a set of classes that provides more powerful and flexible GUI component than the AWT.
- swing is built on the foundation of the AWT.
- swing also uses the same event handling mechanism as the AWT.

## Components and Containers

A swing GUI consists of two key items : components and containers.

All containers are also components.

**Components**: is an independent visual control, such as push button or slider.

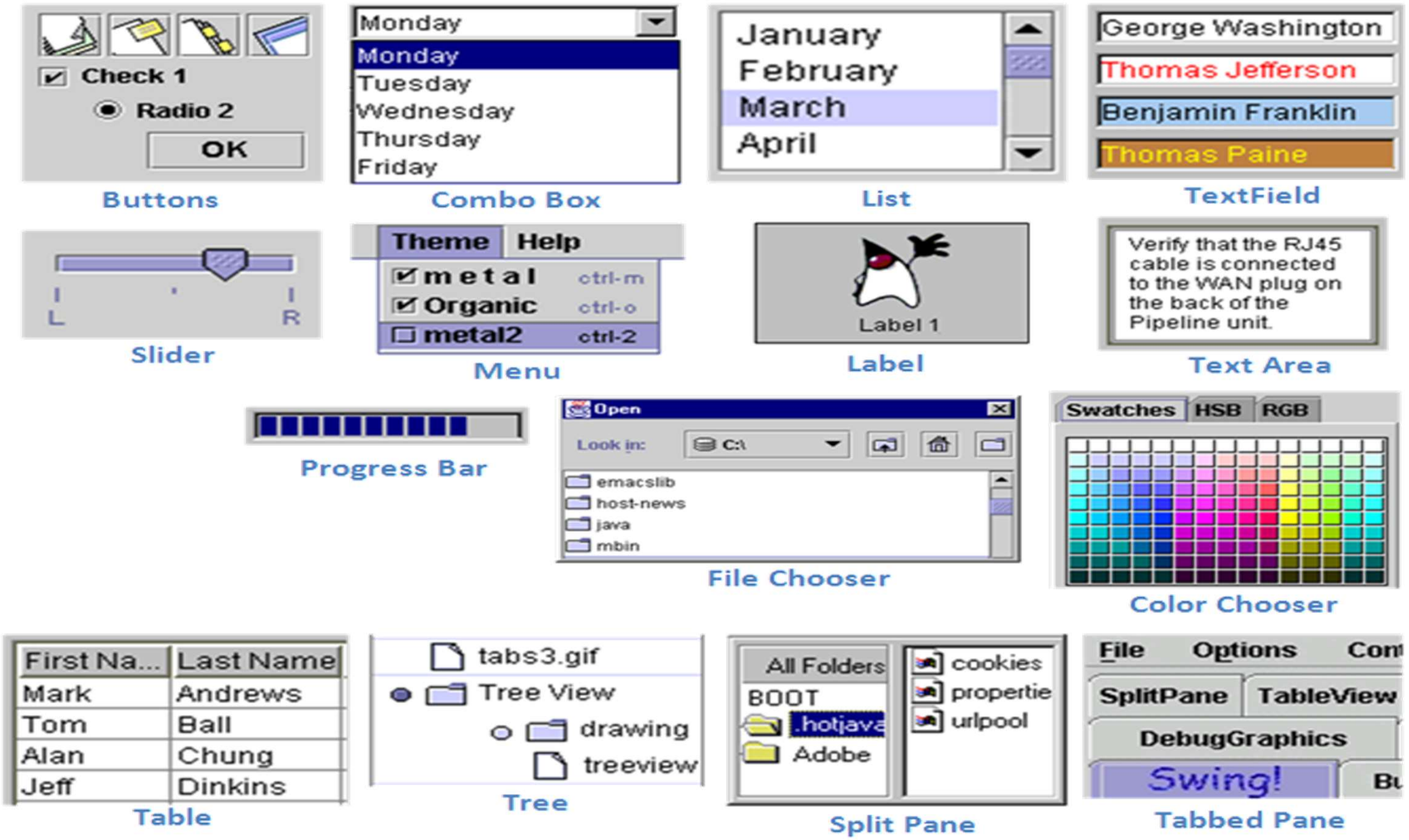
**Container**:

- holds the group of components.
- Container is a special type of components that is designed to hold other components.
- Thus all swing GUI will have a least one container.
- Because containers are components, a container can also hold other containers.

**Example for Container** :  JFrame

**Example for Component :** JLabel, JTextField,Jbutton, …

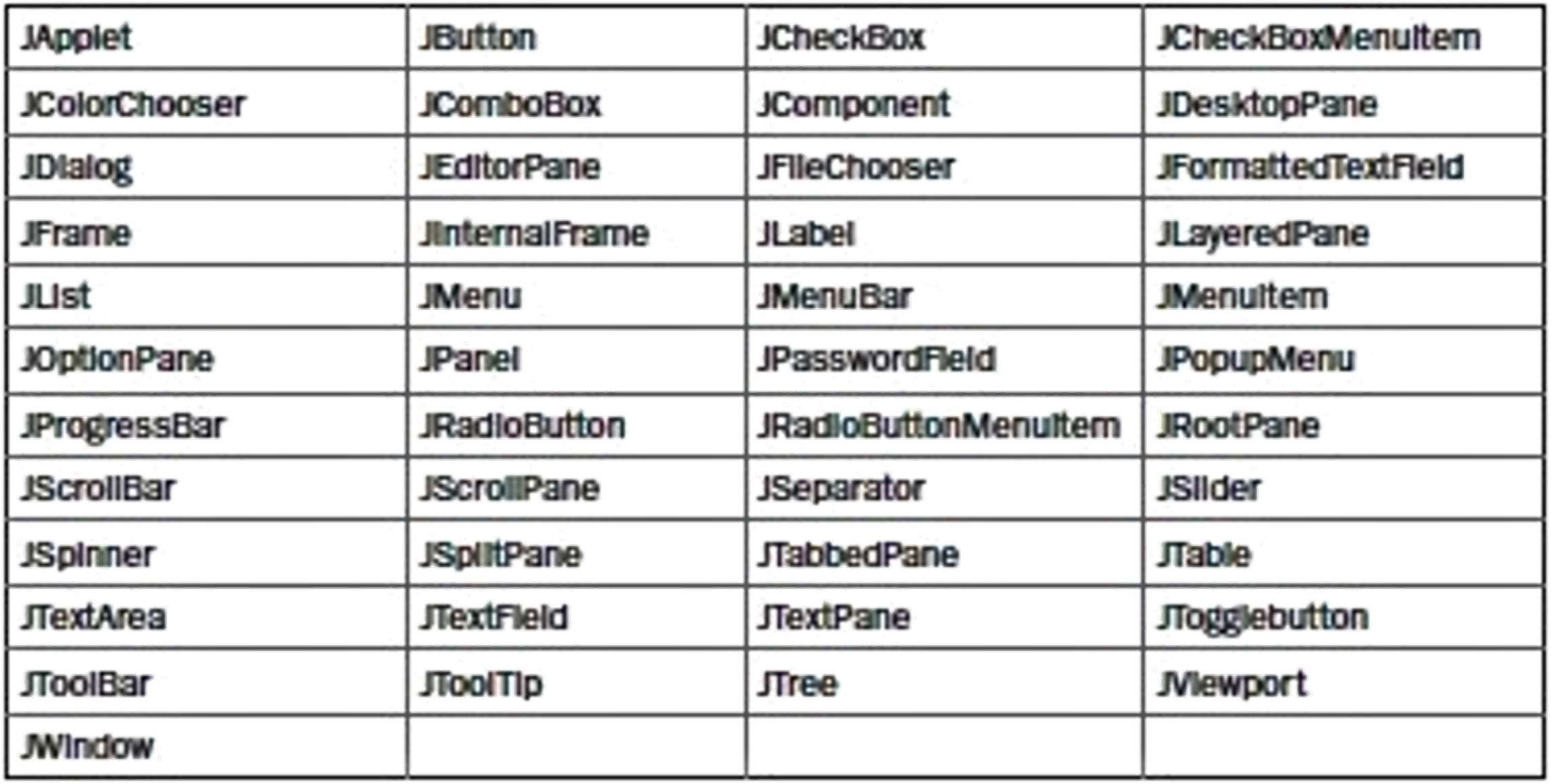
## Components:



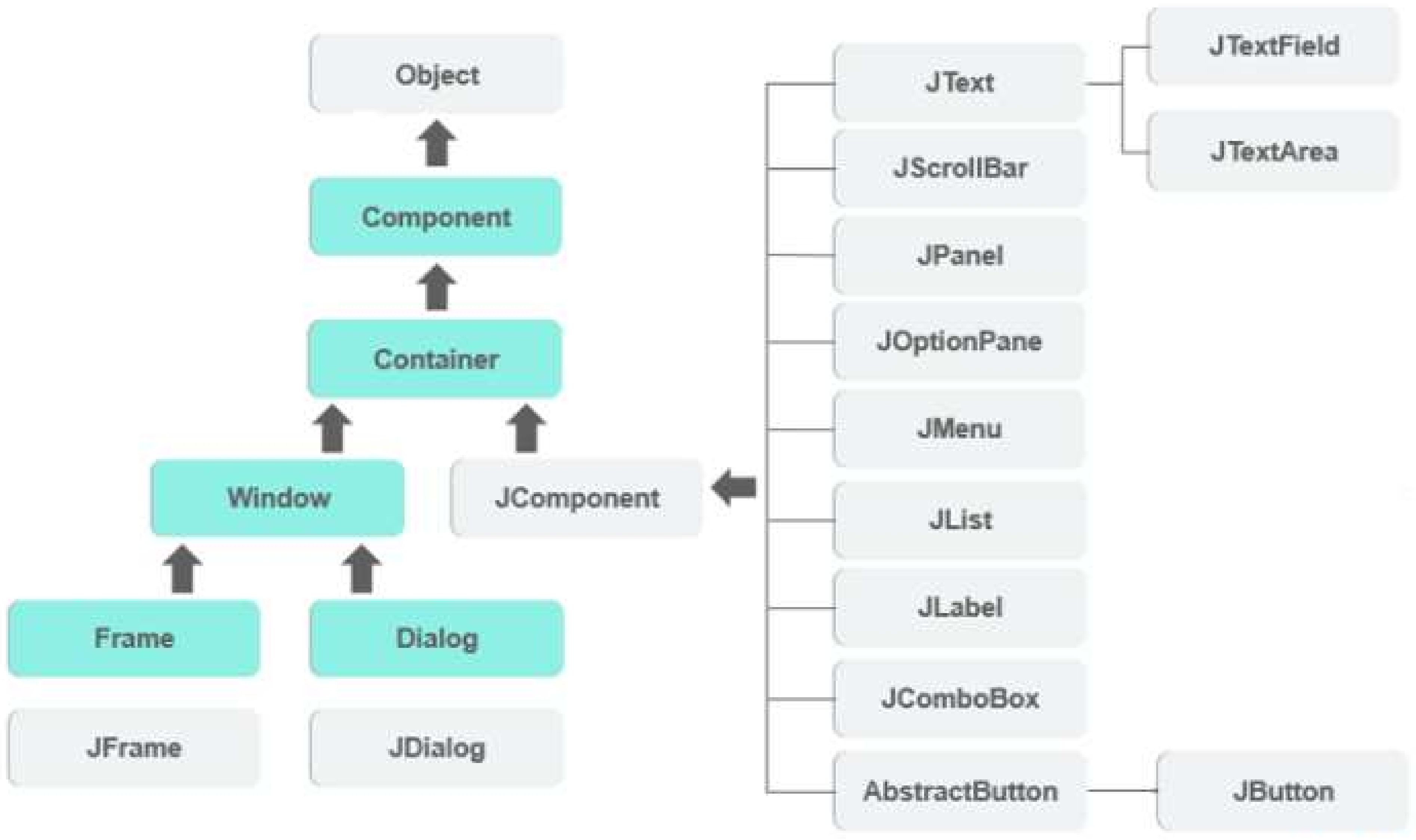
| First Na... | Last Name |
|---|---|
| Mark | Andrews |
| Tom | Ball |
| Alan | Chung |
| Jeff | Dinkins |

## **Components** (Continued …)

- All Swing components are represented as classes within the package javax.swing
  - All begin with letter J



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


Object
Component
Container
Window
JComponent
Frame
Dialog
JFrame
JDialog
JText
JTextField
JTextArea
JScrollBar
JPanel
JOptionPane
JMenu
JList
JLabel
JComboBox
AbstractButton
JButton

## JFrame

Constructor :

```
    JFrame();
    JFrame(Strng s);
```

Example :

```
    JFrame f = new Jframe();
    JFrame f = new Jframe(“First Jframe”);
```

## Jframe methods

```
setSize(275,100);

setVisible(true)

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

add(component c)

setLayout()
```

## Jframe methods

```
JFrame f = new JFrame();
f.setSize(275,100);
f.setVisible(true)
f.setDefaultCloseOperation(Jframe.EXIT_ON_CLOSE);
add(Component c);
```

## Swing components

JLabel,

JTextField,

JButton

## JLabel

- Creates a label

  JLabel (String *str*)

  Use a text for the label

  String getText ( )

  Get the text associated with the label

  void setText (String *str*)

  Set the text for the label

## JTextField

constructors

```
JTextField();
JTextField(int size);
JTextField(String s)
```

Example :

```
JTextField  t1 = new JTextField(12);
```

methods :  To set or obtain the text displayed by the text field.

**String getText ( )**

**void setText (String str)**

## JButton

constructors

```
JButton(String s);
```

example

```
JButton b1 = new JButton(“OK”);
```

add components to the container using add method.

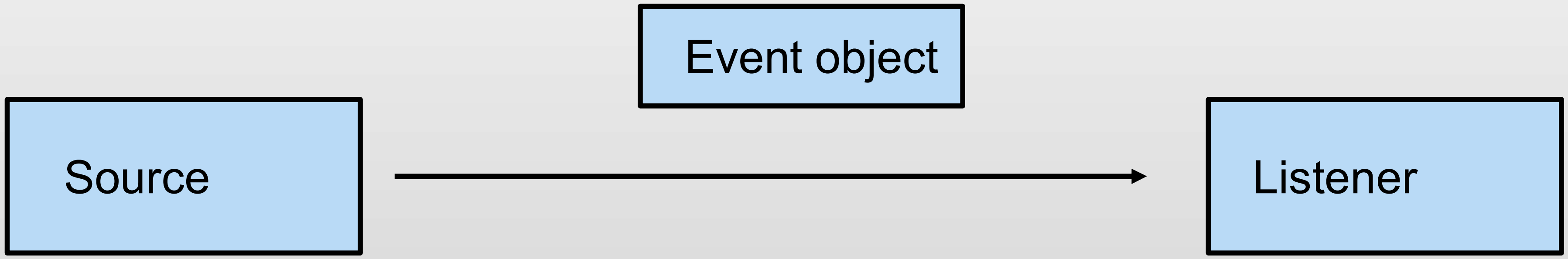
```
JFrame  f = new JFrame(“A simple swing application”);
  f.setSize(275,100);
  f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
Jlabel l1 = new Jlabel(“Enter name”);
JTextField t1 = new JTextField(12);
JTextField t2 = new JTextField(12);
Jbutton b1 = new Jbutton(“OK”);
 f.setLayout(new FlowLayout());
 f.add(l1);
 f.add(t1);
 f.add(t2);
 f.add(b1);
f. setVisible(true);
```

## The delegation event model

defines  standard and consistent mechanisms to generate and process  events.

**Concept :**

*Event sources have methods that allow you to register event listeners with them. When an event happens to the source, the source sends a notification of that event to all the listener objects that were registered for that event.*

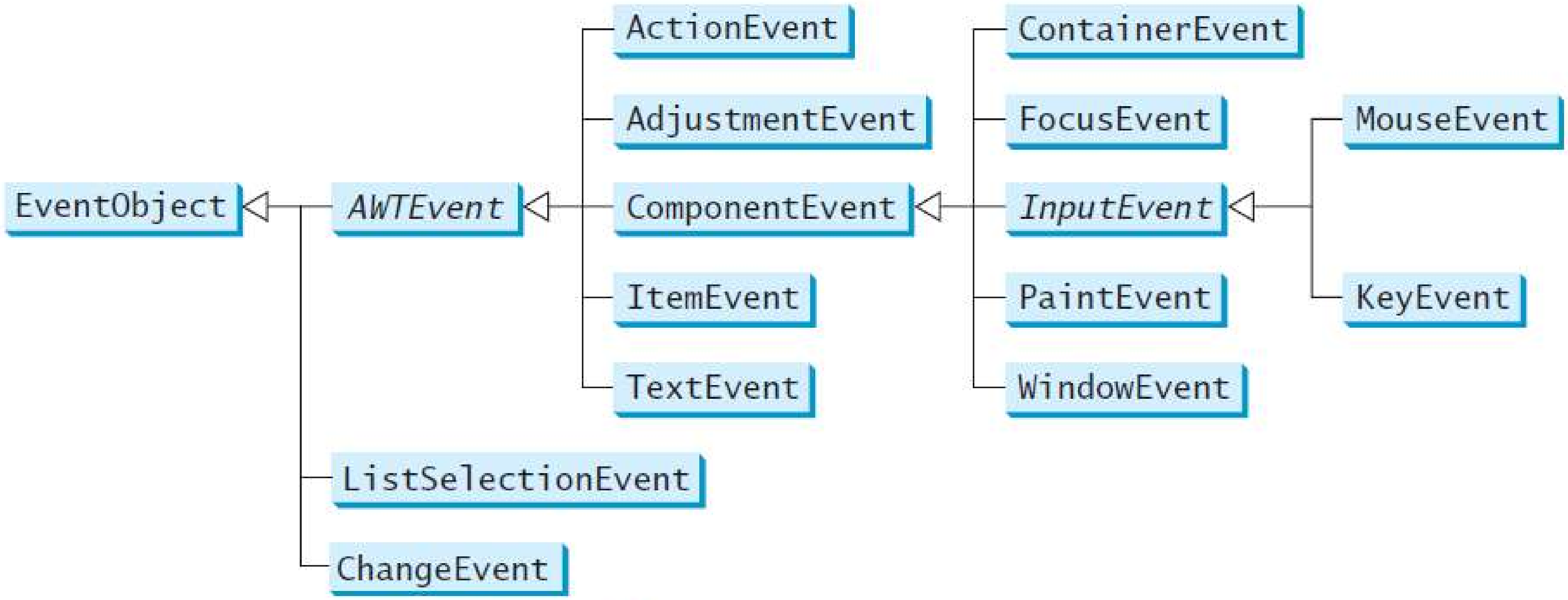


| Event Source | Description |
|---|---|
| Button | Generates action events when the button is pressed. |
| Check box | Generates item events when the check box is selected or deselected. |
| Choice | Generates item events when the choice is changed. |
| List | Generates action events when an item is double-clicked; generates item events when an item is selected or deselected. |
| Menu item | Generates action events when a menu item is selected; generates item events when a checkable menu item is selected or deselected. |
| Scroll bar | Generates adjustment events when the scroll bar is manipulated. |
| Text components | Generates text events when the user enters a character. |
| Window | Generates window events when a window is activated, closed, deactivated, deiconified, iconified, opened, or quit. |

PresentationPoint

| Event Class | Description |
|---|---|
| ActionEvent | Generated when a button is pressed, a list item is double-clicked, or a menu item is selected. |
| AdjustmentEvent | Generated when a scroll bar is manipulated. |
| ComponentEvent | Generated when a component is hidden, moved, resized, or becomes visible. |
| ContainerEvent | Generated when a component is added to or removed from a container. |
| FocusEvent | Generated when a component gains or loses keyboard focus. |
| InputEvent | Abstract superclass for all component input event classes. |
| ItemEvent | Generated when a check box or list item is clicked; also occurs when a choice selection is made or a checkable menu item is selected or deselected. |
| KeyEvent | Generated when input is received from the keyboard. |
| MouseEvent | Generated when the mouse is dragged, moved, clicked, pressed, or released; also generated when the mouse enters or exits a component. |
| MouseWheelEvent | Generated when the mouse wheel is moved. |
| TextEvent | Generated when the value of a text area or text field is changed. |
| WindowEvent | Generated when a window is activated, closed, deactivated, deiconified, iconified, opened, or quit. |

PresentationPoint



An event is an object of the `EventObject` class.

PresentationPoint

| *User Action* | *Source Object* | *Event Type Fired* |
|---|---|---|
| Click a button | `JButton` | `ActionEvent` |
| Press return on a text field | `JTextField` | `ActionEvent` |
| Select a new item | `JComboBox` | `ItemEvent`, `ActionEvent` |
| Select item(s) | `JList` | `ListSelectionEvent` |
| Click a check box | `JCheckBox` | `ItemEvent`, `ActionEvent` |
| Click a radio button | `JRadioButton` | `ItemEvent`, `ActionEvent` |
| Select a menu item | `JMenuItem` | `ActionEvent` |
| Move the scroll bar | `JScrollBar` | `AdjustmentEvent` |
| Move the scroll bar | `JSlider` | `ChangeEvent` |
| Window opened, closed, iconified, deiconified, or closing | `Window` | `WindowEvent` |
| Mouse pressed, released, clicked, entered, or exited | `Component` | `MouseEvent` |
| Mouse moved or dragged | `Component` | `MouseEvent` |
| Key released or pressed | `Component` | `KeyEvent` |

PresentationPoint

| *Event Class (Handlers)* | *Listener Interface* | *Listener Methods* |
|---|---|---|
| `ActionEvent` | `ActionListener` | `actionPerformed(ActionEvent)` |
| `ItemEvent` | `ItemListener` | `itemStateChanged(ItemEvent)` |
| `MouseEvent` | `MouseListener` | `mousePressed(MouseEvent)` |
| | | `mouseReleased(MouseEvent)` |
| | | `mouseEntered(MouseEvent)` |
| | | `mouseExited(MouseEvent)` |
| | | `mouseClicked(MouseEvent)` |
| | `MouseMotionListener` | `mouseDragged(MouseEvent)` |
| | | `mouseMoved(MouseEvent)` |
| `KeyEvent` | `KeyListener` | `keyPressed(KeyEvent)` |
| | | `keyReleased(KeyEvent)` |
| | | `keyTyped(KeyEvent)` |

PresentationPoint

## ActionListener interface

- To implement the ActionListener interface, the listener class must have a method called actionPerformed that receives an ActionEvent object as a parameter.

```
class MyListener  implements  ActionListener
{
            public void actionPerformed( ActionEvent  event )
            {
                        // reaction to button click goes here
                        . . .
            }
}
```

PresentationPoint

## More swing components...

- JCheckBox
- JList
- JComboBox

PresentationPoint

## CheckBox

```
JCheckBox(String st);
```

When the user selects or deselects a checkbox, an ItemEvent is generated.

Following method is used to determine the state of checkbox.

```
boolean  isSelected();
```

## JList

The List class provides a compact, multiple-choice,  selection list.

Unlike the Choice object, which shows only the single selected item in the menu, a List object can be constructed to show any number of choices in the visible window.

- Methods
  - int getSelectedIndex ( )
    - Returns the index of the item that is selected
  - Object getSelectedValue ( )
    - Returns the object that is selected

PresentationPoint

## JComboBox

Swing provides a combo box (a combination of a text field and a drop-down list) through the JComboBox class.

- Methods
  - int getItemCount ( )
    - Returns the number of items in the list
  - int getSelectedIndex ( )
    - Returns the index of the selected item
  - Object getSelectedItem ( )
    - Returns the selected item
  - void removeItem (Object obj)
    - Removes obj from the list
  - void removeItemAt (int index)
    - Removes the object at the specified index