

Swing

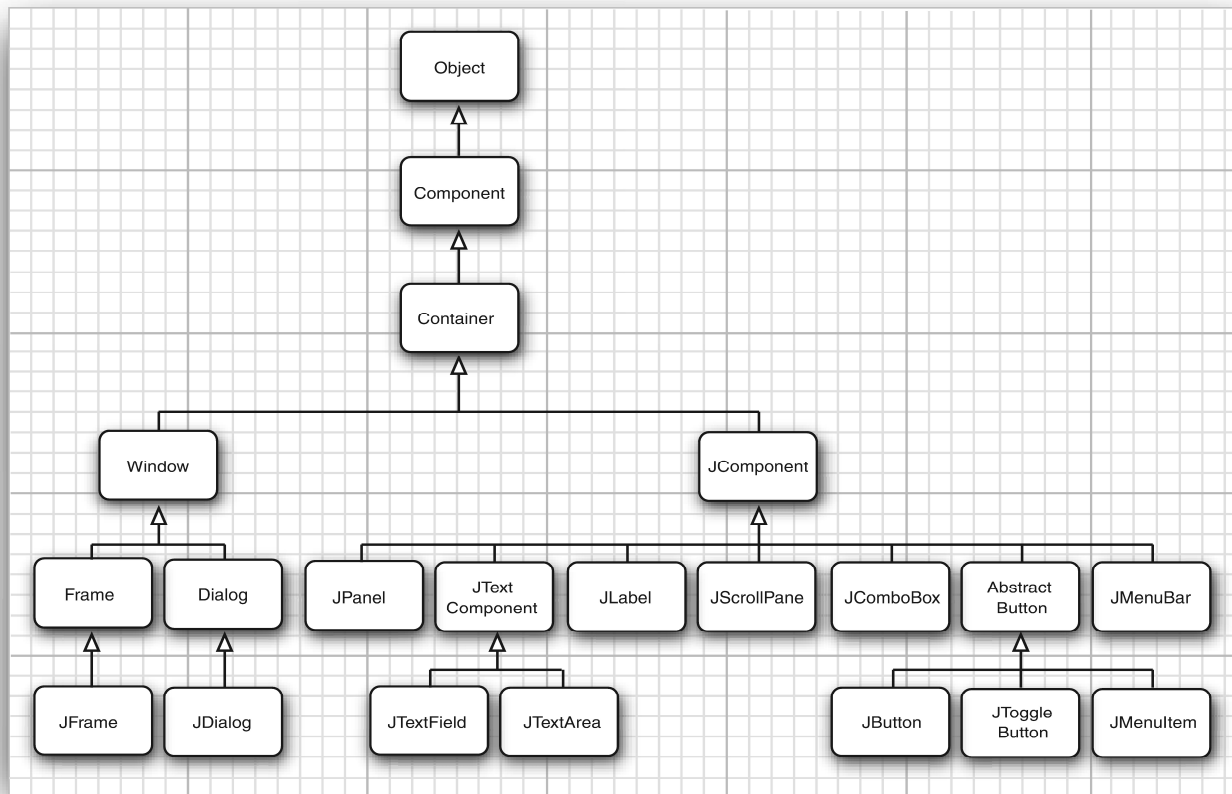
Part 2 : Basic Components

Chapter 11, Core Java, Volume I

Contents

- Swing Components
- Text Fields
- Text Areas
- Example: Text
- Check Boxes
- Example: Check Box
- Radio Buttons
- Example: Radio Button
- Combo Boxes
- Example: Combo Box

Swing Components



Text Fields

- Construct `TextField` with default text and size:
`TextField textField = new TextField("Default input", 20);`
`TextField textField = new TextField(20);`
- Get/set user input:
`String input = textField.getText();`
`textField.setText("Hello!");`
- Use a `JPasswordField` to mask user input. The input is returned as a `char[]` array:
`char[] input = passwordField.getPassword();`
`// Think why the String class is not used as the return type!`
- Use labels to identify components (e.g. to label input fields):
`JLabel label = new JLabel("Password: ", SwingConstants.RIGHT); //== JLabel.RIGHT`
- Tip: Label text can contain HTML for markup.

Text Areas

- Construct `JTextArea` with number of rows and columns:
`textArea = new JTextArea(8, 40); // 8 lines of 40 columns each`
- Call `getText/setText` to read and write contents.
- Enable line wrapping:
`textArea.setLineWrap(true); // long lines are wrapped`
- Or enclose in scroll pane:
`textArea = new JTextArea(8, 40);`
`JScrollPane scrollPane = new JScrollPane(textArea);`
`// Add scrollPane to container`



Example: Text (Listing 11.1)

```
public class TextComponentFrame extends JFrame
{
    public static final int TEXTAREA_ROWS = 8;
    public static final int TEXTAREA_COLUMNS = 20;
    public TextComponentFrame()
    {
        JTextField textField = new JTextField();
        JPasswordField passwordField = new JPasswordField();

        JPanel northPanel = new JPanel();
        northPanel.setLayout(new GridLayout(2, 2));
        northPanel.add(new JLabel("User name: ", SwingConstants.RIGHT));
        northPanel.add(textField);
        northPanel.add(new JLabel("Password: ", SwingConstants.RIGHT));
        northPanel.add(passwordField);
        add(northPanel, BorderLayout.NORTH);
    }
}
```

Example: Text (Listing 11.1)

```
JTextArea textArea = new JTextArea(TEXTAREA_ROWS, TEXTAREA_COLUMNS);
JScrollPane scrollPane = new JScrollPane(textArea);
add(scrollPane, BorderLayout.CENTER);

// add button to append text into the text area
JPanel southPanel = new JPanel();
JButton insertButton = new JButton("Insert");
southPanel.add(insertButton);
insertButton.addActionListener(event ->
    textArea.append("User name: " + textField.getText() + " Password: "
        + new String(passwordField.getPassword()) + "\n"));

add(southPanel, BorderLayout.SOUTH);
pack();
}
```

Check Boxes

- Checkbox can be unchecked or checked.
- Provide label in constructor:
`boldBox = new JCheckBox("Bold");`
- Call `isSelected/setSelected` to read/write state:
`boldBox.setSelected(true);`
- Install an action listener:
`boldBox.addActionListener(event -> {
 if (boldBox.isSelected()) ...
});`



Example: CheckBox (Listing 11.2)

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

/**
 * A frame with a sample text label and check boxes for selecting font
 * attributes.
 */
public class CheckBoxFrame extends JFrame
{
    private JLabel label;
    private JCheckBox bold;
    private JCheckBox italic;
    private static final int FONTSIZE = 24;
```

Example: CheckBox (Listing 11.2)

```
public CheckBoxFrame()
{
    // add the sample text label

    label = new JLabel("The quick brown fox jumps over the lazy dog.");
    label.setFont(new Font("Serif", Font.BOLD, FONTSIZE));
    add(label, BorderLayout.CENTER);

    // this listener sets the font attribute of
    // the label to the check box state

    ActionListener listener = event -> {
        int mode = 0;
        if (bold.isSelected()) mode += Font.BOLD;
        if (italic.isSelected()) mode += Font.ITALIC;
        label.setFont(new Font("Serif", mode, FONTSIZE));
    };
```

Example: CheckBox (Listing 11.2)

```
// add the check boxes

JPanel buttonPanel = new JPanel();

bold = new JCheckBox("Bold");
bold.addActionListener(listener);
bold.setSelected(true); // initially bold checked
buttonPanel.add(bold);

italic = new JCheckBox("Italic");
italic.addActionListener(listener);
buttonPanel.add(italic);

add(buttonPanel, BorderLayout.SOUTH);
pack();
}
```

Radio Buttons

- Use radio buttons for choosing one of several options.
- Buttons must be inside a `ButtonGroup`:

```
ButtonGroup group = new ButtonGroup();
JRadioButton smallButton = new JRadioButton("Small", false);
group.add(smallButton);
JRadioButton mediumButton = new JRadioButton("Medium", true);
group.add(mediumButton);
```
- Button group only controls the behavior that selects only one button. For layout, add radio buttons to a panel.
- Add action listener to each button. It is called when the option is selected.

```
smallButton.addActionListener(event -> label.setFont(new Font("Serif", Font.PLAIN, 8)));
```
- Note: There is no convenient way to get the currently selected button from the `ButtonGroup`.



Example: RadioButton (Listing 11.3)

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

/**
 * A frame with a sample text label and radio buttons for selecting font sizes.
 */
public class RadioButtonFrame extends JFrame
{
    private JPanel buttonPanel;
    private ButtonGroup group;
    private JLabel label;
    private static final int DEFAULT_SIZE = 36;
```

Example: RadioButton (Listing 11.3)

```
public RadioButtonFrame()
{
    // add the sample text label
    label = new JLabel("The quick brown fox jumps over the lazy dog.");
    label.setFont(new Font("Serif", Font.PLAIN, DEFAULT_SIZE));
    add(label, BorderLayout.CENTER);

    // add the radio buttons
    buttonPanel = new JPanel();
    group = new ButtonGroup();
    addRadioButton("Small", 8);
    addRadioButton("Medium", 12);
    addRadioButton("Large", 18);
    addRadioButton("Extra large", 36);

    add(buttonPanel, BorderLayout.SOUTH);
    pack();
}
```

Example: RadioButton (Listing 11.3)

```
public void addRadioButton(String name, int size)
{
    boolean selected = size == DEFAULT_SIZE;
    JRadioButton button = new JRadioButton(name, selected);
    group.add(button);
    buttonPanel.add(button);

    // this listener sets the label font size
    ActionListener listener = event -> label.setFont(new Font("Serif", Font.PLAIN, size));
    button.addActionListener(listener);
}
}
```

Combo Boxes

- Use combo boxes when there are more than a handful of choices.

- Call `setEditable` to allow editing of the options.
 - A combo box is a combination of a text field and a list of choices.
 - Editing affects only the selected item, not the list of choices.

- `JComboBox` is a generic class:
 - `JComboBox<String>`, `JComboBox<Integer>`, ...

- Get the selected item like this:

```
JComboBox<String> faceCombo = ...;
```

```
Object item = faceCombo.getSelectedItem(); // return type is Object
```

```
String item = (String) faceCombo.getSelectedItem();
```

or

```
String item = faceCombo.getItemAt(faceCombo.getSelectedIndex()); // if not editable
```



Combo Boxes

- An `ActionEvent` is fired when an item is selected:

```
faceCombo.addActionListener(event ->
    label.setFont(new Font(
        comboBox.getItemAt(faceCombo.getSelectedIndex()), Font.PLAIN, DEFAULT_SIZE)));
```

Example : ComboBox (Listing 11.4)

```
import java.awt.BorderLayout;
import java.awt.Font;

import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class ComboBoxFrame extends JFrame
{
    private JComboBox<String> faceCombo;
    private JLabel label;
    private static final int DEFAULT_SIZE = 24;
```

Example : ComboBox (Listing 11.4)

```
public ComboBoxFrame()
{
    // add the sample text label
    label = new JLabel("The quick brown fox jumps over the lazy dog.");
    label.setFont(new Font("Serif", Font.PLAIN, DEFAULT_SIZE));
    add(label, BorderLayout.CENTER);

    // make a combo box and add face names

    faceCombo = new JComboBox<>();
    faceCombo.addItem("Serif");
    faceCombo.addItem("SansSerif");
    faceCombo.addItem("Monospaced");
    faceCombo.addItem("Dialog");
    faceCombo.addItem("DialogInput");
}
```

Example : ComboBox (Listing 11.4)

```
// the combo box listener changes the label font to the selected face name
```

```
faceCombo.addActionListener(event ->  
    label.setFont(  
        new Font(faceCombo.getItemAt(faceCombo.getSelectedIndex()),  
            Font.PLAIN, DEFAULT_SIZE)));
```

```
// add combo box to a panel at the frame's southern border
```

```
JPanel comboPanel = new JPanel();  
comboPanel.add(faceCombo);  
add(comboPanel, BorderLayout.SOUTH);  
pack();  
}  
}
```