Outline

Introduction

Features

AWT vs Swings

Swing Components

- Applet
- Icons and Labels
- Buttons
- Textcomponents
- Combo Boxes

Advanced Swing Components

- Tabbed panes
- Scroll panes
- Table
- Tree
- Tooltips
- Progressbar

References

A Presentation

On

Advanced Java Programming

By

Atul S. Chaudhari

M.E.(Computer), C-DAC Pune

Department of Computer Engineering
S. S. V. P. S's B. S. Deore Polytechnic, Dhule

Outline

Introduction

Features

AWT vs Swings

Swing Components

- Applet
- Icons and Labels
- Buttons
- Textcomponents
- Combo Boxes

Advanced Swing Components

- Tabbed panes
- Scroll panes
- Table
- Tree
- Tooltips
- Progressbar

References

Unit II
Swings

(10 Marks)

Outline of Topics

1 Introduction

2 Features

3 AWT vs Swings

4 Swing Components
- Applet
- Icons and Labels
- Buttons
- Textcomponents
- Combo Boxes

5 Advanced Swing Components
- Tabbed panes
- Scroll panes
- Table
- Tree
- Tooltips
- Progressbar

6 References

Outline

Introduction

Features

AWT vs Swings

Swing Components

Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes

Advanced Swing Components

Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar

References

Swings

Outline

Introduction

Features

AWT vs Swings

Swing Components

Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes

Advanced Swing Components

Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar

References

- Swing is part of JFC (Java Foundation Classes) that is used to create window-based applications.
- It is built on the foundation of AWT API and entirely written in java.
- Swing Provides large set of classes that are used to create more powerful and flexible components than are possible with AWT.
- In addition to the AWT components like Label, Button swing provides several exiting components such as tabbed panes, trees, tables.
- Almost all Swing components are derived from a single parent called JComponent which extends the AWT Container class.
- Each AWT component has a Swing equivalent that begins with prefix "J"
- All swing classes are present in javax.swing package

Features

Outline

Introduction

Features

AWT vs Swings

Swing Components

Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes

Advanced Swing Components

Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar

References

- Lightweight components
- Pluggable look-and feels
- Borders
- Graphics Debugging
- Tooltips
- Easy mouseless operation
- Easy Scrolling
- New Layout Managers
- Advanced components

AWT vs Swings

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

Sr. No.	AWT	Swings
1	AWT components are platform-dependent.	Java swing components are platform-independent.
2	AWT components are heavyweight.	Swing components are lightweight.
3	AWT doesn't support pluggable look and feel.	Swing supports pluggable look and feel.
4	AWT provides less components than Swing.	Swing provides more powerful components such as tables, trees, scrollpanes, tabbedpane etc.
5	AWT components has less properties than swing components	Swing components has more properties than AWT components
6	AWT doesn't follows MVC(Model View Controller) architecture	Swing follows MVC architecture.

Swing Components

Outline

Introduction

Features

AWT vs Swings

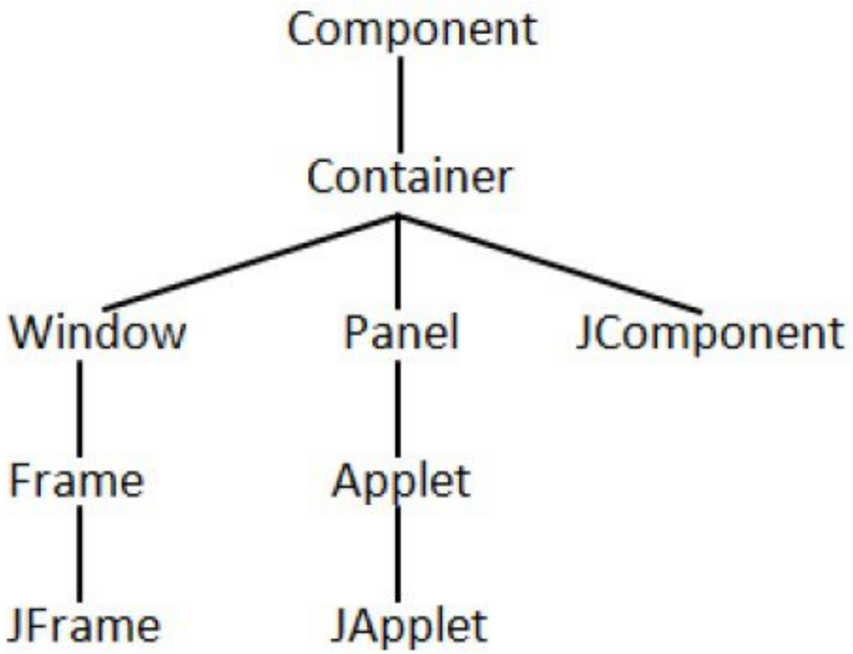
Swing Components

Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes

Advanced Swing Components

Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar

References



Swing Components contd...

Outline

Introduction

Features

AWT vs Swings

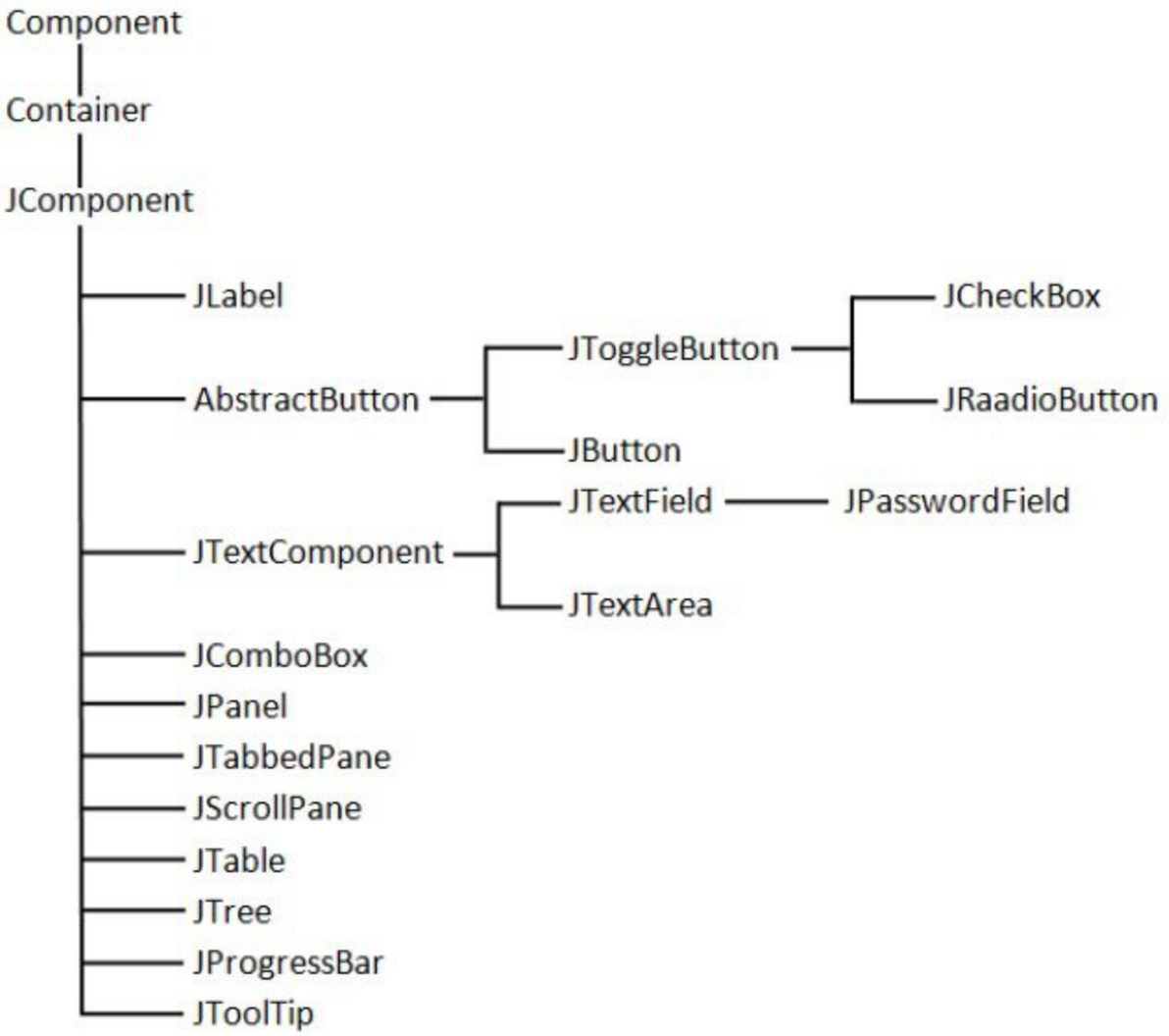
Swing Components

- Applet
- Icons and Labels
- Buttons
- Textcomponents
- Combo Boxes

Advanced Swing Components

- Tabbed panes
- Scroll panes
- Table
- Tree
- Tooltips
- Progressbar

References



ImageIcon

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- Icons are specified by objects of type Icon, which is an interface defined by Swing.
- ImageIcon implements Icon and encapsulates an image.

Constructors:

ImageIcon(String filename)
ImageIcon(URL hp)

JLabel

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- **Constructors:**

  JLabel(String str)
  JLabel(Icon i)
  JLabel(String str, Icon i, int align)

  Here, parameter align has one of the following value:
  LEFT, RIGHT, CENTER, LEADING, or TRAILING these constants are defined in SwingConstatnts interface

- **Methods:**

  String getText( )
  void setText(String str)
  Icon getIcon( )
  void setIcon(Icon i)

Swing Buttons

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- Swing defines four types of buttons: JButton, JToggleButton, JCheckBox, and JRadioButton.
- All are subclasses of the AbstractButton class, which extends JComponent.
- All buttons share a set of common properties.
- AbstractButton contains many methods that allow you to control the behavior of buttons.
- For example, we can set different icons that are displayed for the button when it is disabled, pressed, or selected.
- Another icon can be used as a rollover icon, which is displayed when the mouse is positioned over a button.

- **Methods:**

```
void setDisabledIcon(Icon di)
void setPressedIcon(Icon pi)
void setSelectedIcon(Icon si)
void setRolloverIcon(Icon ri)

String getText( )
void setText(String str)
```

JButton

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- The JButton class provides the functionality of a push button.
- JButton allows an icon, a string, or both to be associated with the push button.
- **Constructors:**

  JButton(String str)
  JButton(Icon i)
  JButton(String str, Icon i)

JCheckBox

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- The JCheckBox class provides the functionality of a check box.
- JCheckBox class is subclass of JToggleButton, which provides support for two-state buttons.

- **Constructors:**

  JCheckBox( )
  JCheckBox(String str)
  JCheckBox(String str, boolean state)
  JCheckBox(Icon i)
  JCheckBox(Icon i, boolean state)
  JCheckBox(String str, Icon i)
  JCheckBox(String str, Icon i, boolean state)

JRadioButton

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- Radio buttons are a group of mutually exclusive buttons, in which only one button can be selected at any one time.
- The JRadioButton class provides the functionality of a radio buttons.
- JRadioButton class is subclass of JToggleButton.

- **Constructors:**

  JRadioButton( )
  JRadioButton(String str)
  JRadioButton(String str, boolean state)
  JRadioButton(Icon i)
  JRadioButton(Icon i, boolean state)
  JRadioButton(String str, Icon i)
  JRadioButton(String str, Icon i, boolean state)

- In order for their mutually exclusive nature to be activated, radio buttons must be configured into a group.
- A button group is created by the ButtonGroup class.
- ButtonGroup class has only default constructor.
- Radio buttons are then added to the button group via add( ) method

  void add(AbstractButton ab)

JTextField

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- JTextField is lightweight component that allows editing of single line of text
- It is derived from JTextComponent, which provides the basic functionality common to Swing text components.

- **Constructors:**

  JTextField(int cols)
  JTextField(String str)
  JTextField(String str, int cols)

JPasswordField

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- JPasswordField is lightweight component that allows editing of single line of text
- In JPasswordField view indicates something was typed, but does not show original characters.
- It is derived from JTextField class

- **Constructors:**

  JPasswordField(int cols)
  JPasswordField(String str)
  JPasswordField(String str, int cols)

JTextArea

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- JTextArea is lightweight component that allows editing of multiple lines of text
- It is derived from JTextComponent, which provides the basic functionality common to Swing text components.
- **Constructors:**

  JTextArea(int rows, int cols)
  JTextArea(String str)
  JTextArea(String str, int rows, int cols)

JComboBox

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- Combo box is a combination of a text field and a drop-down list
- JComboBox class is used to create combo box.
- A combo box normally displays one entry.
- The user can select a value from the drop-down list, which appears at the user's request.
- If you make the combo box editable, then the combo box includes an editable field into which the user can type a value.

- **Constructors:**

  JComboBox( )
  JComboBox(Object items[ ])
  JComboBox(Vector v)

- **Methods:**

  void addItem(Object item)
  Object getSelectedItem()
  void setEditable(boolean f)

JTabbedPane

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- JTabbedPane class is used to create tabbed panes
- It manages a set of components by linking them with tabs.
- Selecting a tab causes the component associated with that tab to come to the forefront.

- **Constructor:**

  JTabbedPane( )

- **Method:**

  void addTab(String name, Component comp)

- **Steps to create tabbed pane :**
  1. Create an instance of JTabbedPane class.
  2. Add each tab by calling addTab( ).
  3. Add the tabbed pane to the content pane.

JScrollPane

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- JScrollPane is a lightweight container that automatically handles the scrolling of another component.
- The component being scrolled can either be an individual component, such as JTextArea, or a group of components contained within another lightweight container, such as a JPanel.
- If the object being scrolled is larger than the viewable area, horizontal and/or vertical scroll bars are automatically provided, and the component can be scrolled through the pane.
- **Constructors:**

  JScrollPane(Component comp)
  JScrollPane(Component comp, int vsb, int hsb)

  Here, values of parameters vsb and hsb are defined in ScrollPaneConstatnts interface:

  VERTICAL_SCROLLBAR_ALWAYS VERTICAL_SCROLLBAR_AS_NEEDED
  HORIZONTAL_SCROLLBAR_ALWAYS HORIZONTAL_SCROLLBAR_AS_NEEDED

- **Steps to create scroll pane :**
  1. Create an object of component to be scrolled.
  2. Create an instance of JScrollPane, passing to it the object to scroll.
  3. Add the scroll pane to the content pane.

JTable

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- JTable is a component that displays rows and columns of data.
- You can drag the cursor on column boundaries to resize columns.
- You can also drag a column to a new position.
- Depending on its configuration, it is also possible to select a row, column, or cell within the table, and to change the data within a cell.

- **Constructors:**

  JTable(Object data[ ][ ], Object colHeads[ ])

- **Steps to create JTable:**
  1. Create an instance of JTable class.
  2. Create a JScrollPane object, specifying the table as the object to scroll.
  3. Add the table to the scroll pane.
  4. Add the scroll pane to the content pane.

JTree

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- A tree is a component that presents a hierarchical view of data.
- User has ability to expand or collapse individual subtrees in this display.
- Trees are implemented in Swing by the JTree class.
- **Constructors:**

  JTree(Object obj[ ])
  JTree(Vector v)
  JTree(TreeNode tn)

- The TreeNode interface declares methods that obtain information about a tree node.
- The MutableTreeNode interface extends TreeNode. It declares methods that can insert and remove child nodes or change the parent node.
- The DefaultMutableTreeNode class implements the MutableTreeNode interface. It represents a node in a tree.

  DefaultMutableTreeNode(Object obj)

JTree contd...

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- To create a hierarchy of tree nodes, the add( ) method of DefaultMutable-TreeNode can be used.

  void add(MutableTreeNode node)

- **Steps to create JTree:**
  1. Create an instance of JTree.
  2. Create a JScrollPane and specify the tree as the object to be scrolled.
  3. Add the tree to the scroll pane.
  4. Add the scroll pane to the content pane.

JProgressbar

Outline
Introduction
Features
AWT vs Swings
Swing Components
Applet
Icons and Labels
Buttons
Textcomponents
Combo Boxes
Advanced Swing Components
Tabbed panes
Scroll panes
Table
Tree
Tooltips
Progressbar
References

- A component that visually displays the progress of some task.
- As the task progresses towards completion, the progress bar displays the task's percentage of completion.
- This percentage is typically represented visually by a rectangle which starts out empty and gradually becomes filled in as the task progresses.
- The progress bar can display a textual representation of percentage.

- **Constructors:**

  JProgressBar()
  JProgressBar(int orient)
  JProgressBar(int min, int max)
  JProgressBar(int orient, int min, int max)

  Here, parameter orient has one of the following value:
  SwingConstatnts.VERTICAL or SwingConstatnts.HORIZONTAL

- **Methods:**

  void setValue(int v)
  int getValue()
  void setStringPainted(boolean f)
  void setBorderPainted(boolean f)
  void setString(String ps)

References

Outline

Introduction

Features

AWT vs Swings

Swing Components

- Applet
- Icons and Labels
- Buttons
- Textcomponents
- Combo Boxes

Advanced Swing Components

- Tabbed panes
- Scroll panes
- Table
- Tree
- Tooltips
- Progressbar

References


H. Schildt, *Complete Reference Java*. McGrawHill Education, New Delhi.


Holzner and Steven, *Java 2 Programming Black Book*. Dreamtech Press, New Delhi.


K. L. Solutions, *Java Sever Programming Tutorial Java EE6 Black Book*. Dreamtech Press, New Delhi.


"Website." [Online]. Available: https://www.javatpoint.com/java-tutorial


"Website." [Online]. Available: https://www.tutorialspoint.com