Khulna Khan Bahadur Ahsanullah University

Object-oriented programming

CSE 1203

Lecture -13

Java JOptionPane

The JOptionPane class is used to provide standard dialog boxes such as message dialog box, confirm dialog box and input dialog box. These dialog boxes are used to display information or get input from the user. The JOptionPane class inherits JComponent class.

JOptionPane class declaration

1. **public class** JOptionPane **extends** JComponent **implements** Accessible

Common Constructors of JOptionPane class

Constructor	Description
JOptionPane()	It is used to create a JOptionPane with a test message.
JOptionPane(Object message)	It is used to create an instance of JOptionPane to display a message.
JOptionPane(Object message, int messageType	It is used to create an instance of JOptionPane to display a message with specified message type and default options.

Common Methods of JOptionPane class

Methods	Description
JDialog createDialog(String title)	It is used to create and return a new parentless JDialog with the specified title.
static void showMessageDialog(Component parentComponent, Object message)	It is used to create an information-message dialog titled "Message".
static void showMessageDialog(Component parentComponent, Object message, String title, int messageType)	It is used to create a message dialog with given title and messageType.
static int showConfirmDialog(Component parentComponent, Object message)	It is used to create a dialog with the options Yes, No and Cancel; with the title, Select an Option.
static String showInputDialog(Component parentComponent, Object message)	It is used to show a question-message dialog requesting input from the user parented to parentComponent.
void setInputValue(Object newValue)	It is used to set the input value that was selected or input by the user.

Java JOptionPane Example: showMessageDialog()

```
1. import javax.swing.*;
2. public class OptionPaneExample {
3. JFrame f;
4. OptionPaneExample(){
5.     f=new JFrame();
6.     JOptionPane.showMessageDialog(f,"Hello, Welcome to Javatpoint.");
7. }
8. public static void main(String[] args) {
9.     new OptionPaneExample();
10.}  }
```

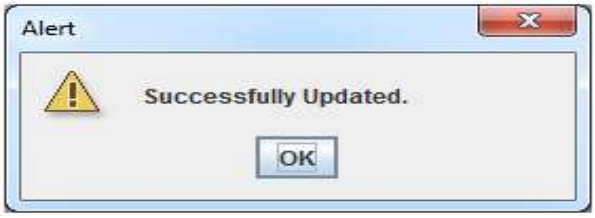
Output:

Java JOptionPane Example: showMessageDialog()

```
1. import javax.swing.*;
2. public class OptionPaneExample {
3. JFrame f;
4. OptionPaneExample(){
5.     f=new JFrame();
6.     JOptionPane.showMessageDialog(f,"Successfully Updated.","Alert",JOptionPane.WARNING_MESSAGE);
```

```
7.  }
8.  public static void main(String[] args) {
9.      new OptionPaneExample();
10. } }
```

Output:



Java JOptionPane Example: showInputDialog()

```
1.  import javax.swing.*;
2.  public class OptionPaneExample {
3.  JFrame f;
4.  OptionPaneExample(){
5.      f=new JFrame();
6.      String name=JOptionPane.showInputDialog(f,"Enter Name");     }
7.  public static void main(String[] args) {
8.      new OptionPaneExample();
9.  } }
```

Output:

Java JOptionPane Example: showConfirmDialog()

```
1.  import javax.swing.*;
2.  import java.awt.event.*;
3.  public class OptionPaneExample extends WindowAdapter{
4.  JFrame f;
5.  OptionPaneExample(){
6.      f=new JFrame();
7.      f.addWindowListener(this);
8.      f.setSize(300, 300);
9.      f.setLayout(null);
10.     f.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
11.     f.setVisible(true);  }
12.   public void windowClosing(WindowEvent e) {
13.    int a=JOptionPane.showConfirmDialog(f,"Are you sure?");
14.   if(a==JOptionPane.YES_OPTION){
15.    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
16.  }  }
17.  public static void main(String[] args) {
18.    new  OptionPaneExample();
19.  }    }
```

Output:

Java JScrollBar

The object of JScrollbar class is used to add horizontal and vertical scrollbar. It is an implementation of a scrollbar. It inherits JComponent class.

JScrollBar class declaration

Let's see the declaration for javax.swing.JScrollBar class.

1. **public class** JScrollBar **extends** JComponent **implements** Adjustable, Acce ssible

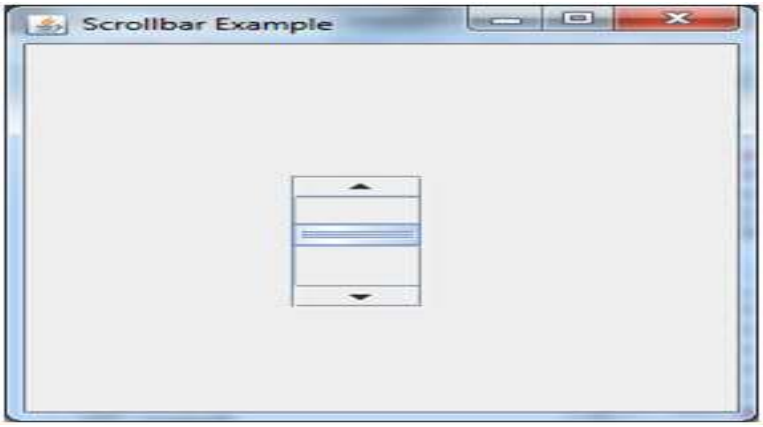
Commonly used Constructors:

Constructor	Description
JScrollBar()	Creates a vertical scrollbar with the initial values.
JScrollBar(int orientation)	Creates a scrollbar with the specified orientation and the initial values.
JScrollBar(int orientation, int value, int extent, int min, int max)	Creates a scrollbar with the specified orientation, value, extent, minimum, and maximum.

Java JScrollBar Example

```
1. import javax.swing.*;
2. class ScrollBarExample
3. {
4. ScrollBarExample(){
5.     JFrame f= new JFrame("Scrollbar Example");
6.  JScrollBar s=new JScrollBar();
7. s.setBounds(100,100, 50,100);
8. f.add(s);
9. f.setSize(400,400);
10.f.setLayout(null);
11.f.setVisible(true);
12.}
13.public static void main(String args[])
14.{
15.new ScrollBarExample();
16.}}
```

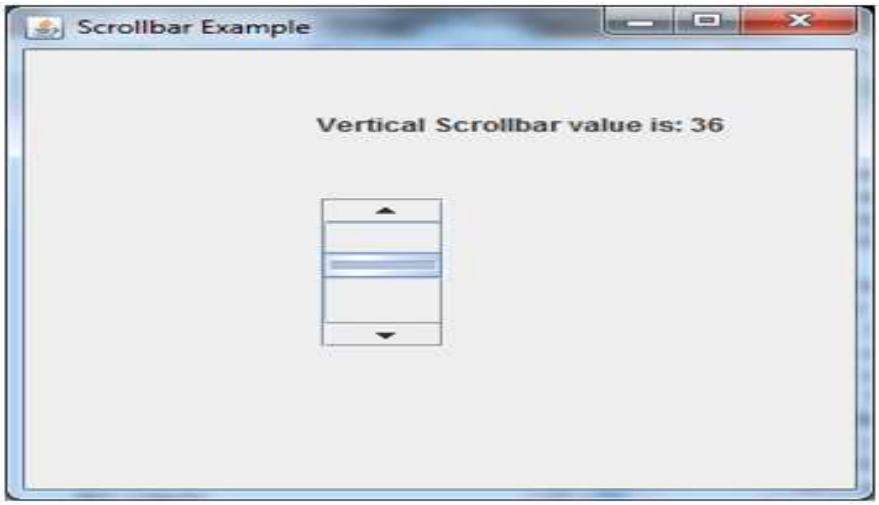
Output:



Java JScrollBar Example with AdjustmentListener

```
1. import javax.swing.*;
2. import java.awt.event.*;
3. class ScrollBarExample
4. {
5. ScrollBarExample(){
6.     JFrame f= new JFrame("Scrollbar Example");
7.     final JLabel label = new JLabel();
8.     label.setHorizontalAlignment(JLabel.CENTER);
9.     label.setSize(400,100);
10.    final JScrollBar s=new JScrollBar();
11.    s.setBounds(100,100, 50,100);
12.    f.add(s); f.add(label);
13.    f.setSize(400,400);
14.   f.setLayout(null);
15.   f.setVisible(true);
16.   s.addAdjustmentListener(new AdjustmentListener() {
17.    public void adjustmentValueChanged(AdjustmentEvent e) {
18.       label.setText("Vertical Scrollbar value is:"+ s.getValue());
19.    }
20. });
21.}
22.public static void main(String args[])
23.{
24.   new ScrollBarExample();
25.}}
```

Output:



Java JFrame

The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI.

Unlike Frame, JFrame has the option to hide or close the window with the help of setDefaultCloseOperation(int) method.

Nested Class

Modifier and Type	Class	Description
protected class	JFrame.AccessibleJFrame	This class implements accessibility support for the JFrame class.

Fields

Modifier and Type	Field	Description
protected AccessibleContext	accessibleContext	The accessible context property.
static int	EXIT_ON_CLOSE	The exit application default window close operation.
protected JRootPane	rootPane	The JRootPane instance that manages the contentPane and optional menuBar for this frame, as well as the glassPane.
protected boolean	rootPaneChecking Enabled	If true then calls to add and setLayout will be forwarded to the contentPane.

Constructors

Constructor	Description
JFrame()	It constructs a new frame that is initially invisible.
JFrame(GraphicsConfiguration gc)	It creates a Frame in the specified GraphicsConfiguration of a screen device and a blank title.
JFrame(String title)	It creates a new, initially invisible Frame with the specified title.

JFrame(String title, GraphicsConfiguration gc)	It creates a JFrame with the specified title and the specified GraphicsConfiguration of a screen device.

Useful Methods

Modifier and Type	Method	Description
protected void	addImpl(Component comp, Object constraints, int index)	Adds the specified child Component.
protected JRootPane	createRootPane()	Called by the constructor methods to create the default rootPane.
protected void	frameInit()	Called by the constructors to init the JFrame properly.
void	setContentPane(Containe contentPane)	It sets the contentPane property

static void	setDefaultLookAndFeel Decorated (boolean default LookAndFeelDecorated)	Provides a hint as to whether or not newly created JFrames should have their Window decorations (such as borders, widgets to close the window, title...) provided by the current look and feel.
void	setIconImage(Image image)	It sets the image to be displayed as the icon for this window.
void	setJMenuBar(JMenuBar menubar)	It sets the menubar for this frame.
void	setLayeredPane(JLayered Pane layeredPane)	It sets the layeredPane property.
JRootPane	getRootPane()	It returns the rootPane object for this frame.
TransferHandler	getTransferHandler()	It gets the transferHandler property.

JFrame Example

```
1. import java.awt.FlowLayout;
2. import javax.swing.JButton;
3. import javax.swing.JFrame;
4. import javax.swing.JLabel;
5. import javax.swing.JPanel;
6. public class JFrameExample {
7.     public static void main(String s[]) {
8.         JFrame frame = new JFrame("JFrame Example");
9.         JPanel panel = new JPanel();
10.        panel.setLayout(new FlowLayout());
11.        JLabel label = new JLabel("JFrame By Example");
12.        JButton button = new JButton();
13.        button.setText("Button");
14.        panel.add(label);
15.        panel.add(button);
16.        frame.add(panel);
17.        frame.setSize(200, 300);
18.        frame.setLocationRelativeTo(null);
19.        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20.        frame.setVisible(true);
21.    } }
```