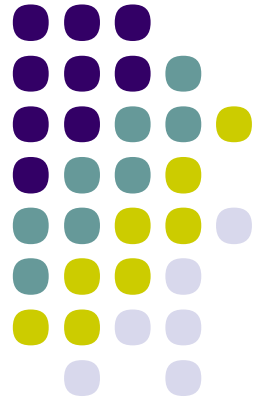


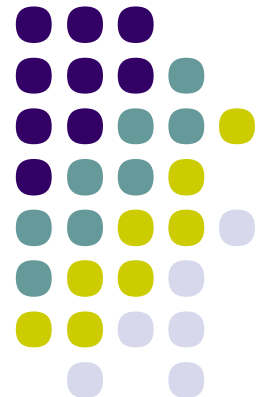
GUI containers

Nguyễn Thị Tú Mi
Email: nttmi@hcmuaf.edu.vn



JPanel

Nguyễn Thị Tú Mi
Email: nttmi@hcmuaf.edu.vn





JPanel

- Dùng để nhóm các component lại với nhau
- Dùng phương thức add để thêm 1 component vào JPanel
- Các thuộc tính của JPanel

Property	Data type	get	is	set	Default value
<code>accessibleContext</code> ^o	AccessibleContext	•			JPanel.AccessibleJPanel()
<code>doubleBuffered</code> ^o	boolean		•	•	true
<code>layout</code> ^o	LayoutManager	•		•	FlowLayout()
<code>opaque</code> ^{o, b}	boolean	•	•		true
<code>UI</code> ^{1,4}	PaneUI	•		•	From L&F
<code>UIClassID</code>	String	•			"PanelUI"
^{1,4} since 1.4, ^b bound, ^o overridden					



Constructor

- `public JPanel()`
- `public JPanel(boolean isDoubleBuffered)`
- `public JPanel(LayoutManager layout)`
- `public JPanel(LayoutManager layout, boolean isDoubleBuffered)`

Opacity



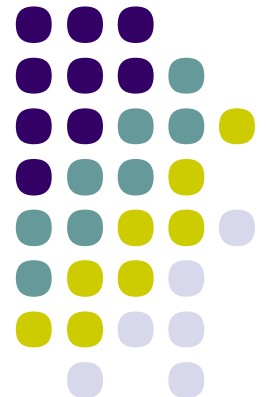
Opacity



```
public class OpaqueExample extends JFrame {
    public OpaqueExample() {
        super("Opaque JPanel Demo");
        setSize(400, 200);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        JPanel opaque = createNested(true);
        JPanel notOpaque = createNested(false);
        getContentPane().setLayout(new FlowLayout());
        getContentPane().add(opaque);
        getContentPane().add(notOpaque);
    }
    // Create a JPanel containing another JPanel. The inner JPanel's opacity is set according
    // to the parameter. A JButton is placed inside the inner JPanel to give it some content.
    public JPanel createNested(boolean opaque) {
        JPanel outer = new JPanel(new FlowLayout());
        JPanel inner = new JPanel(new FlowLayout());
        outer.setBackground(Color.white);
        inner.setBackground(Color.black);
        inner.setOpaque(opaque);
        inner.setBorder(BorderFactory.createLineBorder(Color.gray));
        inner.add(new JButton("Button"));
        outer.add(inner);
        return outer;
    }
    public static void main(String[] args) {
        OpaqueExample oe = new OpaqueExample();
        oe.setVisible(true);
    }
}
```

JFrame

Nguyễn Thị Tú Mi
Email: nttmi@hcmuaf.edu.vn





Cấu trúc cây kế thừa

Object

Component

Container

JComponent

Window

AbstractButton

JLabel

JPanel

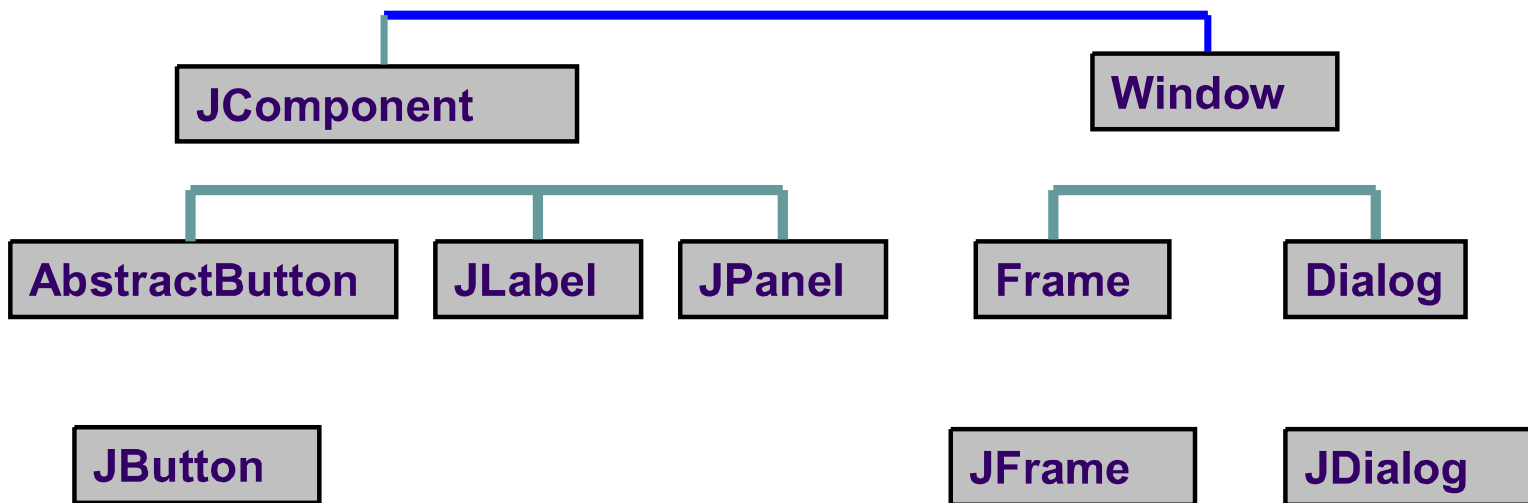
Frame

Dialog

JButton

JFrame

JDialog





Các thuộc tính của JFrame

Property	Data type	get	is	set	Default value
<code>accessibleContext^o</code>	<code>AccessibleContext</code>	.			<code>JFrame.Accessible-JFrame()</code>
<code>background^o</code>	<code>Color</code>	.		.	<code>UIManager.getColor ("control")</code>
<code>contentPane^o</code>	<code>Container</code>	.		.	From <code>rootPane</code>
<code>defaultCloseOperation</code>	<code>int</code>	.		.	<code>HIDE_ON_CLOSE</code>
<code>glassPane^o</code>	<code>Component</code>	.		.	From <code>rootPane</code>
<code>JMenuBar^o</code>	<code>JMenuBar</code>	.		.	From <code>rootPane</code>
<code>layeredPane^o</code>	<code>JLayeredPane</code>	.		.	From <code>rootPane</code>
<code>layout^o</code>	<code>LayoutManager</code>	.		.	<code>BorderLayout()</code>
<code>rootPane^{o, *}</code>	<code>JRootPane</code>	.		.	<code>JRootPane()</code>
<code>rootPaneCheckingEnabled^p</code>	<code>boolean</code>		.	.	<code>true</code>
<code>title^o</code>	<code>String</code>	.		.	<code>""</code>



Constructor

- `JFrame()`
- `JFrame(String title)`



Các phương thức

`public void setDefaultCloseOperation(int operation)`

- Thiết lập phản ứng của chương trình khi nhấn nút “close” trên frame này
- Một số lựa chọn Java cung cấp:
 - `DO_NOTHING_ON_CLOSE`
 - `HIDE_ON_CLOSE`
 - `DISPOSE_ON_CLOSE`
 - `EXIT_ON_CLOSE` (default)



Các phương thức

- `public int getDefaultCloseOperation()`
- `public void setJMenuBar(JMenuBar menubar)`
- `public JMenuBar getJMenuBar()`
- `public Container getContentPane()`
- `public JLayeredPane getLayeredPane()`
- `public void setLayeredPane(JLayeredPane layeredPane)`
- `public Component getGlassPane()`
- `public void setGlassPane(Component glassPane)`



Các phương thức

- `public String getTitle()`
- `public void setTitle(String title)`
- `public boolean isResizable()`
- `public void setResizable(boolean resizable)`
- `public void setCursor(Cursor cursor)`
 - **Java cung cấp 1 số cursor sau:** `CROSSHAIR_CURSOR`,
`TEXT_CURSOR`, `WAIT_CURSOR`, `HAND_CURSOR`,
`MOVE_CURSOR`



Các phương thức

- `public void setLocation(int x, int y)`
- `public void setLocation(Point p)`
 - Chuyển JFrame đến vị trí mới
 - Góc trên bên trái của JFrame sẽ ứng với tọa độ mới đưa vào.
- `public void setLocationRelativeTo(Component c)`
 - Thiết lập vị trí của JFrame này dựa trên component chỉ định.
 - Nếu component là null thì JFrame sẽ được định vị ở giữa màn hình.



Các phương thức

- `public void setBounds(int x, int y, int width, int height)`
 - Di chuyển và định kích thước của JFrame
 - Vị trí mới của góc trên bên trái frame sẽ là (x, y)
 - Kích thước mới của JFrame sẽ là width, height



Các phương thức

- `public void setExtendedState(int state)`
 - Thiết lập trạng thái cho JFrame
 - Một số trạng thái mà Java cung cấp:
 - `NORMAL, ICONIFIED, MAXIMIZED_HORIZ, MAXIMIZED_VERT, MAXIMIZED_BOTH, MAXIMIZED_HORIZ, MAXIMIZED_VERT`
 - Nếu trạng thái đó không được hỗ trợ bởi HĐH thì sẽ không có hiệu ứng xảy ra
 - Ứng dụng sẽ xác định xem trạng thái có được hỗ trợ không thông qua phương thức
`java.awt.Toolkit #isFrameStateSupported(int state)`



Lớp Toolkit

- public static Toolkit getDefaultToolkit()
- public abstract Dimension getScreenSize()
- public abstract Image getImage(String filename)

Ví dụ



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

public class DemoFrame extends JFrame {
    Toolkit kit;
    JButton locationButton, cursorButton, iconButton;

    public DemoFrame() {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            e.printStackTrace();
        }
        addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        kit = Toolkit.getDefaultToolkit();
        Dimension screenSize = kit.getScreenSize();
        int screenHeight = screenSize.height;
        int screenWidth = screenSize.width;
        setSize(screenWidth / 2, screenHeight / 2);
        setTitle("untitled Frame");
        setResizable(false);
        Container pane = getContentPane();
        pane.setLayout(new FlowLayout());
    }
}
```

Ví dụ



```
locationButton = new JButton(" Center a Frame");
locationButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setLocationRelativeTo(null);
        setTitle("a Centered Frame");
    }
});

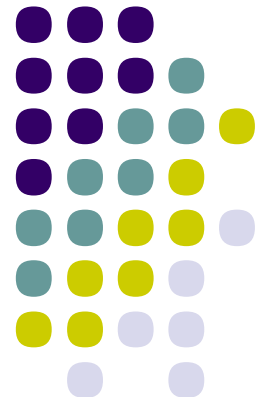
cursorButton = new JButton(" Set Cursor");
cursorButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setCursor(new Cursor(Cursor.CROSSHAIR_CURSOR));
        setTitle("a Cross-Hair cursor");
    }
});

iconButton = new JButton(" set Frame Icon");
iconButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Image img = kit.getImage("hand.jpg");
        setIconImage(img);
        setTitle("a hand icon");
    }
});

pane.add(locationButton);
pane.add(cursorButton);
pane.add(iconButton);
}
```

JWindow

Nguyễn Thị Tú Mi
Email: nttmi@hcmuaf.edu.vn





JWindow

- Là lớp mở rộng của `java.awt.Window`
- Trong AWT, lý do để JWindow tồn tại là tạo ra các popup menu
- Trong Swing đã có JPopupMenu làm nhiệm vụ đó
- Chỉ sử dụng trong trường hợp cần hiển thị 1 component đơn giản nào đó mà không muốn dùng JFrame: splash screen



Thuộc tính của JWindow

Property	Data type	get	is	set	Default value
<code>accessibleContext</code> ^o	<code>AccessibleContext</code>	.			<code>JWindow.AccessibleJWindow()</code>
<code>contentPane</code> ^o	<code>Container</code>	.		.	<code>From rootPane</code>
<code>glassPane</code> ^o	<code>Component</code>	.		.	<code>From rootPane</code>
<code>layeredPane</code> ^o	<code>JLayeredPane</code>	.		.	<code>From rootPane</code>
<code>layout</code> ^o	<code>LayoutManager</code>	.		.	<code>BorderLayout()</code>
<code>rootPane</code> ^{o, *}	<code>JRootPane</code>	.		.	<code>JRootPane()</code>
<code>rootPaneCheckingEnabled</code> ^p	<code>boolean</code>		.	.	<code>true</code>

^ooverridden, ^pprotected



Constructors

- `public JWindow()`
- `public JWindow(JFrame frame)`
- `public JWindow(Window window)`
- `public JWindow(GraphicsConfiguration gc)`
- `public JWindow(Window window, GraphicsConfiguration gc)`



Ví dụ sử dụng JWindow

```
public class SplashScreen extends JWindow {
    private int duration;
    public SplashScreen(int d) {
        duration = d;
    }
    // A simple little method to show a title screen in the center of the screen
    // for the amount of time given in the constructor
    public void showSplash() {
        JPanel content = (JPanel) getContentPane();
        content.setBackground(Color.white);

        // Set the window's bounds, centering the window.
        int width = 450;
        int height = 115;
        Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
        int x = (screen.width - width) / 2;
        int y = (screen.height - height) / 2;
        setBounds(x, y, width, height);

        // Build the splash screen.
        JLabel label = new JLabel(new ImageIcon("sen.gif"));
        JLabel copyrt = new JLabel("this is a demo of splash screen",
            JLabel.CENTER);
        copyrt.setFont(new Font("Sans-Serif", Font.BOLD, 12));
        content.add(label, BorderLayout.CENTER);
        content.add(copyrt, BorderLayout.SOUTH);
        Color oraRed = new Color(156, 20, 20, 255);
        content.setBorder(BorderFactory.createLineBorder(oraRed, 10));
    }
}
```




Ví dụ sử dụng JWindow

```
// Display it.
setVisible(true);

// Wait a little while, maybe while loading resources.
try {
    Thread.sleep(duration);
} catch (Exception e) {
}

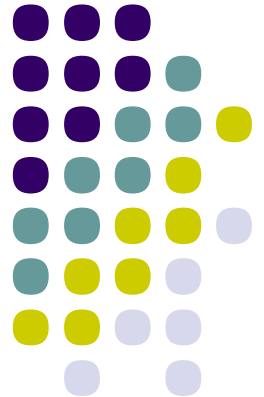
setVisible(false);
}

public void showSplashAndExit() {
    showSplash();
    System.exit(0);
}

public static void main(String[] args) {
    // Throw a nice little title page up on the screen first.
    SplashScreen splash = new SplashScreen(5000);
    // Normally, we'd call splash.showSplash( ) and get on with the program.
    // But, since this is only a test...
    splash.showSplashAndExit();
}
}
```

JScrollPane

Nguyễn Thị Tú Mi
Email: nttmi@hcmuaf.edu.vn





JScrollPane

- JScrollPane giúp bạn hiển thị các component với kích thước động, có thể scroll được.
- Thường được dùng cho các component có kích thước thật lớn hơn kích thước được phép hiển thị.

Ví dụ



JScrollPane Demonstration

	0-1	2-5	6-10
Household	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Office	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Extended Family	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Company (US)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Company (World)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Team	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Will	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Birthday Card List	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
High School	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Country	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Ví dụ



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

public class ScrollDemo extends JFrame {

    JScrollPane scrollpane;

    public ScrollDemo( ) {
        super("JScrollPane Demonstration");
        setSize(300, 200);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        init( );
        setVisible(true);
    }

    public void init( ) {
        JRadioButton form[][] = new JRadioButton[12][5];
        String counts[] = { "", "0-1", "2-5", "6-10", "11-100", "101+" };
        String categories[] = { "Household", "Office", "Extended Family",
                                "Company (US)", "Company (World)", "Team",
                                "Will", "Birthday Card List", "High School",
                                "Country", "Continent", "Planet" };

        JPanel p = new JPanel( );
        p.setSize(600, 400);
        p.setLayout(new GridLayout(13, 6, 10, 0));
```

Ví dụ



```
for (int row = 0; row < 13; row++) {
    ButtonGroup bg = new ButtonGroup( );
    for (int col = 0; col < 6; col++) {
        if (row == 0) {
            p.add(new JLabel(counts[col]));
        }
        else {
            if (col == 0) {
                p.add(new JLabel(categories[row - 1]));
            }
            else {
                form[row - 1][col - 1] = new JRadioButton( );
                bg.add(form[row - 1][col - 1]);
                p.add(form[row - 1][col - 1]);
            }
        }
    }
}
scrollpane = new JScrollPane(p);
getContentPane( ).add(scrollpane, BorderLayout.CENTER);
}

public static void main(String args[]) {
    new ScrollDemo( );
}
}
```



Thuộc tính của JScrollPane

Property	Data type	get	is	set	Default value
accessibleContext ^o	AccessibleContext	•			JScrollPane.AccessibleJScrollPane()
columnHeader ^b	JViewport	•			null
columnHeaderView	Component			•	
componentOrientation ^{b, o}	int	•		•	ComponentOrientation.UNKNOWN
horizontalScrollBar ^b	JScrollBar	•			null
horizontalScrollBarPolicy ^b	int	•		•	HORIZONTAL_SCROLLBAR_AS_NEEDED
layout ^{b, o}	LayoutManager	•		•	new JScrollPaneLayout()
opaque	boolean		•		false
rowHeader ^b	JViewport	•			null
rowHeaderView	Component			•	
UI	ScrollPane-UI	•		•	From L&F
UIClassID ^o	String				"ScrollPaneUI"
validateRoot	boolean		•		true
verticalScrollBar ^b	JScrollBar	•			null
verticalScrollBarPolicy ^b	int	•		•	VERTICAL_SCROLLBAR_AS_NEEDED
viewport ^b	JViewport	•			null
viewportBorder ^b	Border	•		•	null
viewportBorderBounds	Rectangle	•			
viewportView	Component			•	
wheelScrollingEnabled ^{b, 1.4}	boolean		•	•	true
1.4since 1.4, ^b bound, ^o overridden					



Constructors

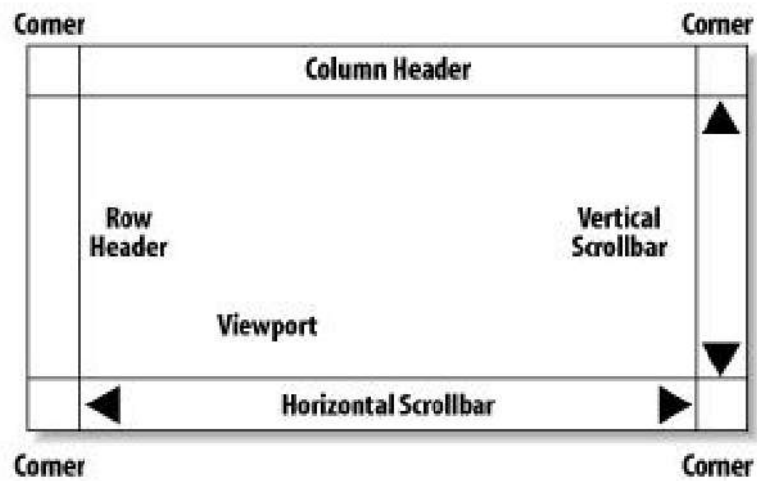
- `public JScrollPane()`
- `public JScrollPane(Component view)`
- `public JScrollPane(Component view, int verticalScrollBarPolicy, int horizontalScrollBarPolicy)`
- `public JScrollPane(int verticalScrollBarPolicy, int horizontalScrollBarPolicy)`



Pane Component Methods

- `public JScrollBar createHorizontalScrollBar()`
- `public JScrollBar createVerticalScrollBar()`
- `public JViewport createViewport()`
- `public Component getCorner(String whichCorner)`
- `public void setCorner(String whichCorner, Component corner)`

ScrollPaneLayout



ScrollPaneLayout constant values



Location string from ScrollPaneConstants	Component location
VIEWPORT	Main viewing area, typically a <code>JViewport</code> component
COLUMN_HEADER	The column header (a row), typically a <code>JViewport</code> component
ROW_HEADER	The row header (a column), typically a <code>JViewport</code> component
HORIZONTAL_SCROLLBAR	The horizontal scrollbar for the viewport; must be a <code>JScrollbar</code> component
VERTICAL_SCROLLBAR	The vertical scrollbar for the viewport; must be a <code>JScrollbar</code> component
LOWER_LEFT_CORNER	The southwest corner, typically empty
LOWER_RIGHT_CORNER	The southeast corner, typically empty
UPPER_LEFT_CORNER	The northwest corner, typically empty
UPPER_RIGHT_CORNER	The northeast corner, typically empty

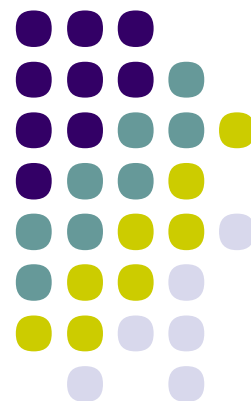
ScrollPaneLayout policy constants



ScrollPaneConstants constant	Type	Effect on Scrollbar component
HORIZONTAL_SCROLLBAR_ALWAYS	int	Always keeps a horizontal scrollbar around, even if the viewport extent area is wide enough to display the entire component
HORIZONTAL_SCROLLBAR_AS_NEEDED	int	Shows a horizontal scrollbar whenever the extent area is smaller than the full component
HORIZONTAL_SCROLLBAR_NEVER	int	Never shows a horizontal scrollbar, even if the component is wider than the viewport extent area
VERTICAL_SCROLLBAR_ALWAYS	int	Always keeps a vertical scrollbar around, even if the viewport extent area is tall enough to display the entire component
VERTICAL_SCROLLBAR_AS_NEEDED	int	Shows a vertical scrollbar whenever the extent area is smaller than the full component
VERTICAL_SCROLLBAR_NEVER	int	Never shows a vertical scrollbar, even if the component is taller than the viewport extent area
HORIZONTAL_SCROLLBAR_POLICY	String	The name of the horizontal scrollbar policy property for use with property change events
VERTICAL_SCROLLBAR_POLICY	String	The name of the vertical scrollbar policy property for use with property change events

JSplitPane

Nguyễn Thị Tú Mi
Email: nttmi@hcmuaf.edu.vn





JSplitPane

- JSplitPane cho phép bạn chia các component ra làm 2 phần (theo chiều ngang hoặc dọc)

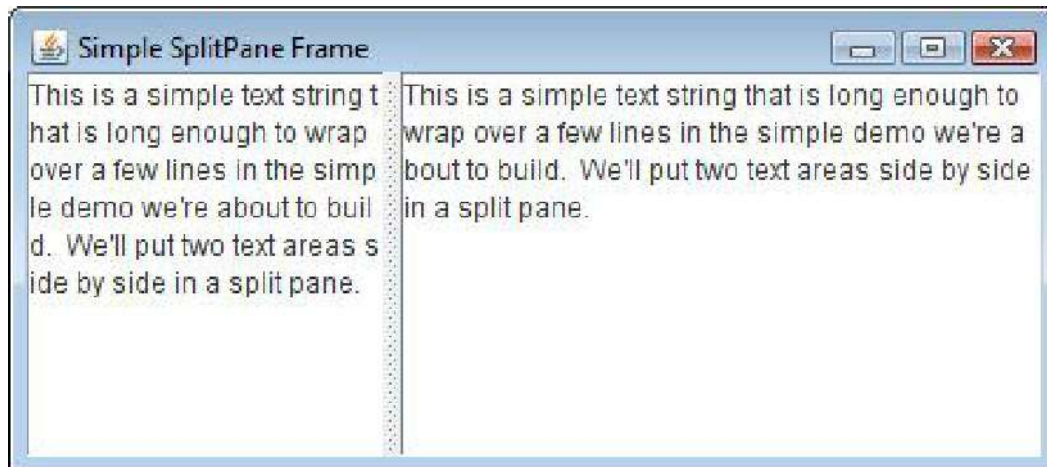


Thuộc tính của JSplitPane

Property	Data type	get	is	set	Default value
accessibleContext ^o	AccessibleContext	·			JSplitPane.AccessibleJSplitPane ()
bottomComponent	Component	·		·	null
continuousLayout ^b	boolean		·	·	false
dividerLocation ^{b,*}	int	·		·	-1
dividerSize ^b	int	·		·	5
lastDividerLocation	int	·		·	0
leftComponent	Component	·		·	null
maximumDividerLocation*	int	·			-1
minimumDividerLocation*	int	·			-1
oneTouchExpandable ^b	boolean		·	·	false
orientation ^b	int	·		·	HORIZONTAL_SPLIT
resizeWeight ^{b, 1.3}	double	·		·	0.0
rightComponent	Component	·		·	null
topComponent	Component	·		·	null
UI ^b	SplitPaneUI	·			From L&F
UIClassID ^o	String	·			"SplitPaneUI"

^{1.3}since 1.3, ^bbound, ^ooverridden

Ví dụ



Ví dụ



```
public class SimpleSplitPane extends JFrame {  
    static String sometext = "This is a simple text string that is long enough "  
        + "to wrap over a few lines in the simple demo we're about to build.  We'll "  
        + "put two text areas side by side in a split pane.";  
  
    public SimpleSplitPane() {  
        super("Simple SplitPane Frame");  
        setSize(450, 200);  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
        JTextArea jt1 = new JTextArea(sometext);  
        JTextArea jt2 = new JTextArea(sometext);  
  
        // Make sure our text boxes do line wrapping and have reasonable minimum  
        // sizes.  
        jt1.setLineWrap(true);  
        jt2.setLineWrap(true);  
        jt1.setMinimumSize(new Dimension(150, 150));  
        jt2.setMinimumSize(new Dimension(150, 150));  
        jt1.setPreferredSize(new Dimension(250, 200));  
        JSplitPane sp = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT, jt1, jt2);  
        getContentPane().add(sp, BorderLayout.CENTER);  
    }  
    public static void main(String args[]) {  
        SimpleSplitPane ssb = new SimpleSplitPane();  
        ssb.setVisible(true);  
    }  
}
```

JSplitPane constants



Constant	Type	Description
BOTTOM	String	Add a component to the bottom of a vertically split pane.
CONTINUOUS_LAYOUT_PROPERTY	String	Used in property change events to specify that the <code>continuousLayout</code> property has been changed.
DIVIDER	String	Add a component as the divider for the pane.
DIVIDER_LOCATION_PROPERTY ^{1..3}	String	Used in property change events to specify that the <code>dividerLocation</code> property has changed.
DIVIDER_SIZE_PROPERTY	String	Used in property change events to specify that the <code>dividerSize</code> property has changed.
HORIZONTAL_SPLIT	int	One of the valid values for the <code>orientation</code> property of a <code>JSplitPane</code> object. This type of split creates a vertical divider, resulting in a set of left/right components.
LAST_DIVIDER_LOCATION_PROPERTY	String	Used in property change events to specify that the <code>lastDividerLocation</code> property has changed.
LEFT	String	Add a component to the left of a horizontally split pane.
ONE_TOUCH_EXPANDABLE_PROPERTY	String	Used in property change events to specify that the <code>oneTouchExpandable</code> property has changed.
ORIENTATION_PROPERTY	String	Used in property change events to specify that the <code>orientation</code> property has changed.
RESIZE_WEIGHT_PROPERTY ^{1..3}	String	Used in property change events to specify that the <code>resizeWeight</code> property has changed.
RIGHT	String	Add a component to the right of a horizontally split pane.
TOP	String	Add a component to the top of a vertically split pane.
VERTICAL_SPLIT	int	One of the valid values for the <code>orientation</code> property of a <code>JSplitPane</code> object. This type of split creates a horizontal divider, resulting in a set of top/bottom components.
^{1..3} since 1.3		



Constructors

- `public JSplitPane()`
- `public JSplitPane(int orientation)`
- `public JSplitPane(int orientation, boolean continuousLayout)`
- `public JSplitPane(int orientation, Component leftOrTop, Component bottomOrRight)`
- `public JSplitPane(int orientation, boolean continuousLayout, Component leftOrTop, Component bottomOrRight)`

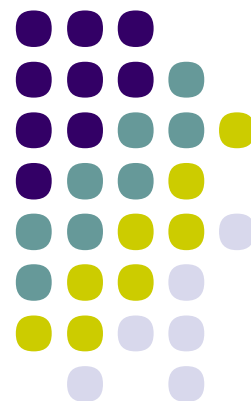


Các phương thức điều khiển

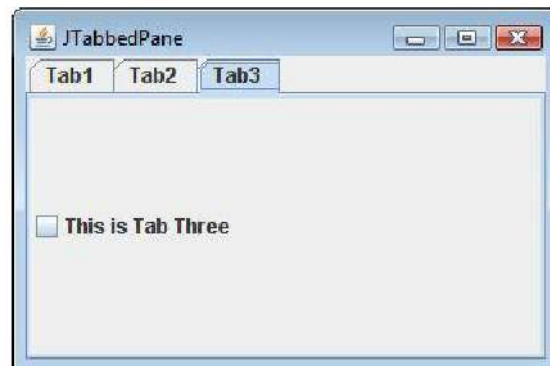
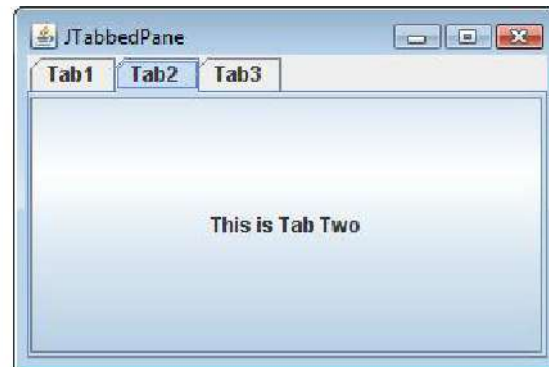
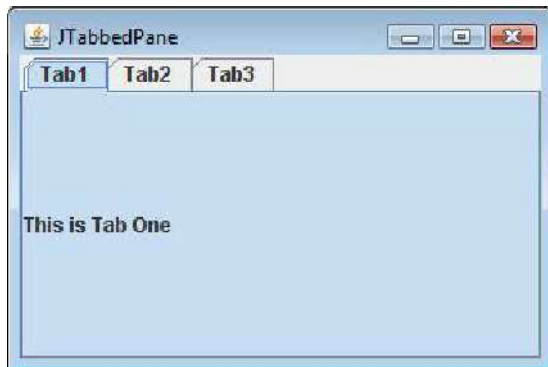
- `public void remove(Component comp)`
- `public void remove(int index)`
- `public void removeAll()`
- `public void resetToPreferredSizes()`
- `public void setDividerLocation(double position)`

JTabbedPane

Nguyễn Thị Tú Mi
Email: nttmi@hcmuaf.edu.vn



Ví dụ



Ví dụ



```
public class SimpleTab extends JFrame {  
  
    JTabbedPane jtp;  
  
    public SimpleTab() {  
        super("JTabbedPane");  
        setSize(200, 200);  
        Container contents = getContentPane();  
        jtp = new JTabbedPane();  
        jtp.addTab("Tab1", new JLabel("This is Tab One"));  
        jtp.addTab("Tab2", new JButton("This is Tab Two"));  
        jtp.addTab("Tab3", new JCheckBox("This is Tab Three"));  
        contents.add(jtp);  
  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
        setVisible(true);  
    }  
  
    public static void main(String args[]) {  
        new SimpleTab();  
    }  
}
```



Thuộc tính của JTabbedPane

Property	Data type	get	is	set	Default value
accessibleContext ^{b, °}	AccessibleContext	*			JTabbedPane.AccessibleJTabbedPane()
backgroundAt ⁱ	Color	*		*	L&F-dependent
boundsAt ⁱ	Rectangle	*			
changeListeners ^{1,4}	ChangeListener[]	*			Empty array
componentAt ⁱ	Component	*		*	
disabledIconAt ⁱ	Icon	*		*	
displayed-MnemonicAt ^{b, i, 1,4}	int	*		*	-1
enabledAt ⁱ	boolean		*	*	
foregroundAt ⁱ	Color	*		*	L&F-dependent
iconAt ⁱ	Icon	*		*	
mnemonicAt ^{b, i, 1,4}	int	*		*	0 (no mnemonic)
model ^b	SingleSelectionModel	*		*	DefaultSingleSelectionModel()
selectedComponent	Component	*		*	null
selectedIndex	int	*		*	-1
tabCount	int	*			0
tabLayoutPolicy ^{b, 1,4}	int	*		*	WRAP_TAB_LAYOUT
tabPlacement ^b	int	*		*	SwingConstants.TOP
tabRunCount	int	*			0
titleAt ^{b, i}	String	*		*	
toolTipTextAt ^{i, 1,3}	String	*		*	
UI	TabbedPaneUI	*		*	null
UIClassID [°]	String				"TabbedPaneUI"
1,3since 1.3, 1,4since 1.4, ^b bound, ⁱ indexed, [°] overridden					



Events

- public void
addChangeListener(ChangeListener l)
- public void
removeChangeListener(ChangeListener l)
- protected ChangeListener
createChangeListener()



Constructors

- `public JTabbedPane()`
- `public JTabbedPane(int tabPlacement)`
- `public JTabbedPane(int tabPlacement, int tabLayoutPolicy)`



Các phương thức

- `public void addTab(String title, Component comp)`
- `public void addTab(String title, Icon tabIcon, Component comp)`
- `public void addTab(String title, Icon tabIcon, Component comp, String tip)`
- `public int indexAtLocation(int x, int y)`
- `public void insertTab(String title, Icon tabIcon, Component comp, String tip, int index)`
- `public Component add(Component component)`
- `public Component add(String title, Component component)`



Các phương thức

- `public Component add(Component component, int index)`
- `public void add(Component component, Object constraints)`
- `public void add(Component component, Object constraints, int index)`
- `public void remove(Component component)`
- `public void removeAll()`
- `public void removeTabAt(int index)`
- `public int indexOfComponent(Component comp)`
- `public int indexOfTab(String title)`
- `public int indexOfTab(Icon icon)`

Summary

