l’API Swing
Création d’interfaces graphiques

Cours Java - F. Michel

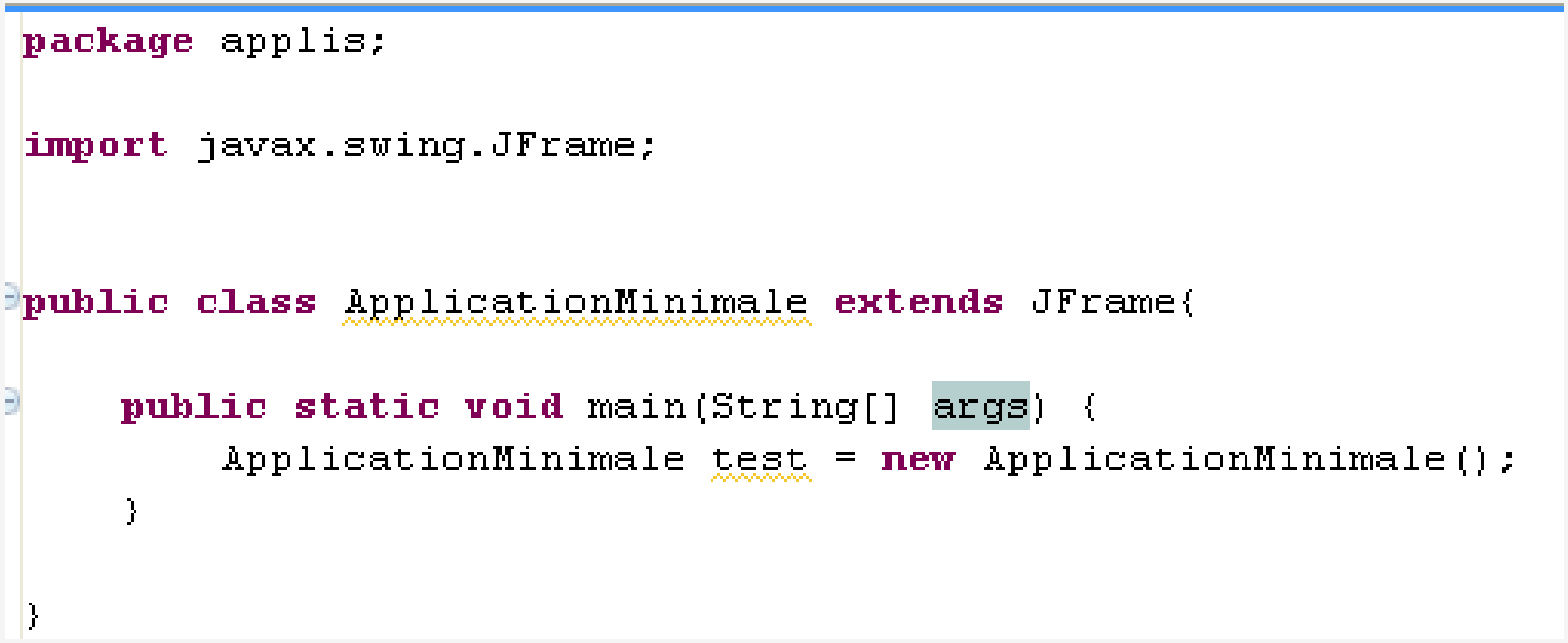
L'API Swing

- javax.accessibility
- javax.swing
- javax.swing.border
- javax.swing.colorchooser
- javax.swing.event
- javax.swing.filechooser
- javax.swing.text
- javax.swing.text.html.parser
- javax.swing.undo
- javax.swing.plaf
- javax.swing.plaf.basic
- javax.swing.plaf.metal
- javax.swing.plaf.multi
- javax.swing.plaf.synth
- javax.swing.table
- javax.swing.text.html
- javax.swing.text.rtf
- javax.swing.tree

javax.swing.JFrame

La base : javax.swing.JFrame

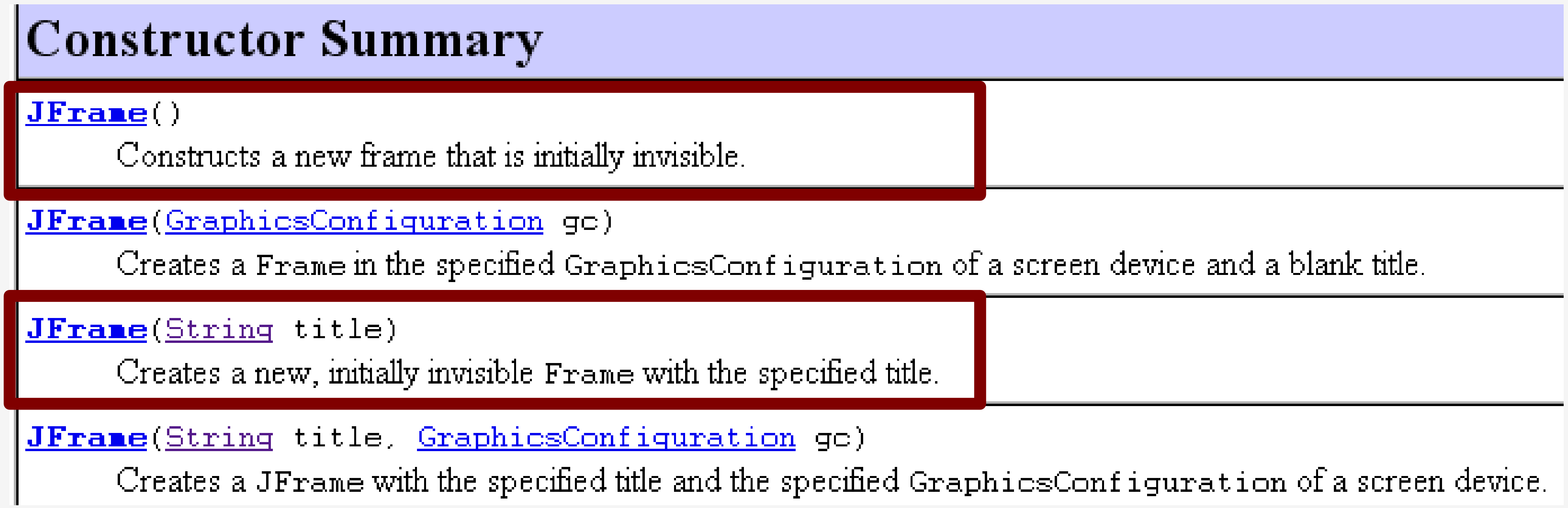
- Cette classe permet de créer une nouvelle fenêtre, autrement dit une nouvelle application:



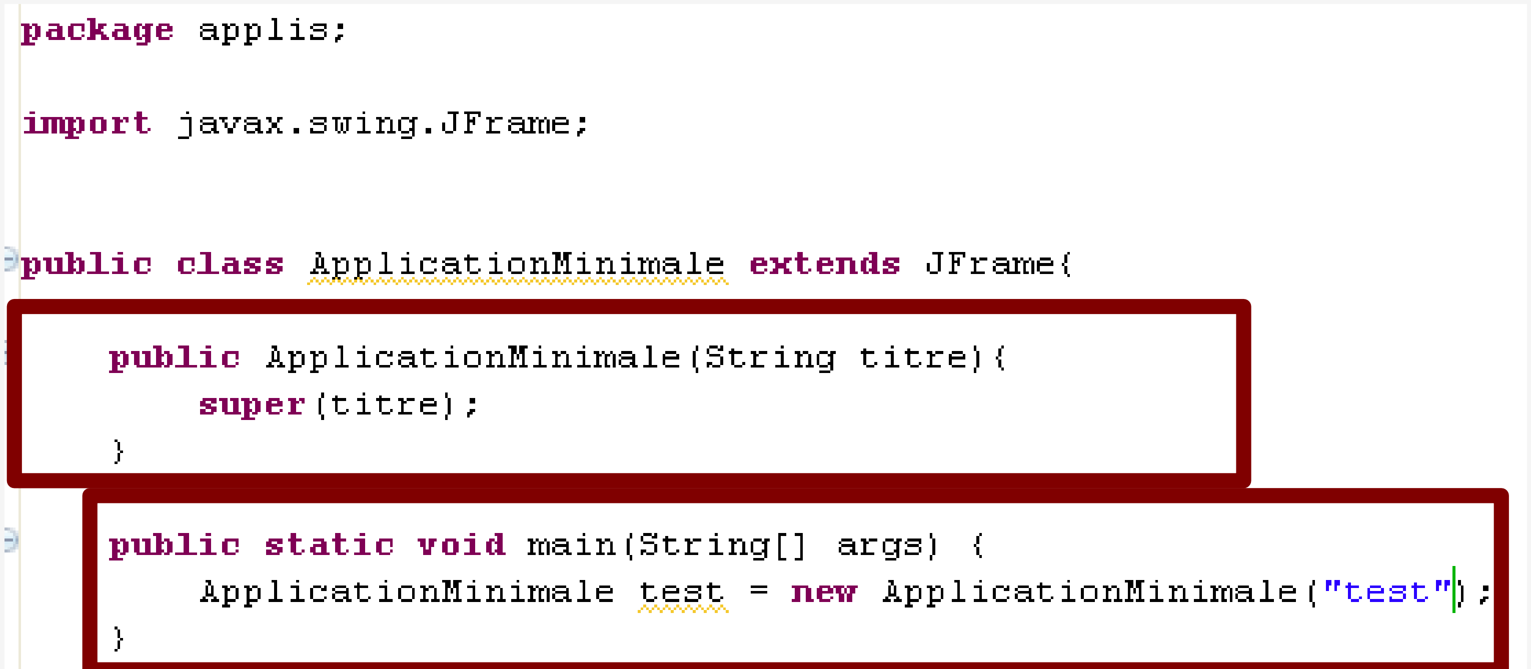
```java
package applis;

import javax.swing.JFrame;


public class ApplicationMinimale extends JFrame{

    public static void main(String[] args) {
        ApplicationMinimale test = new ApplicationMinimale();
    }

}
```

On l'exécute : rien, pas de fenêtre.

Analyse de la classe JFrame



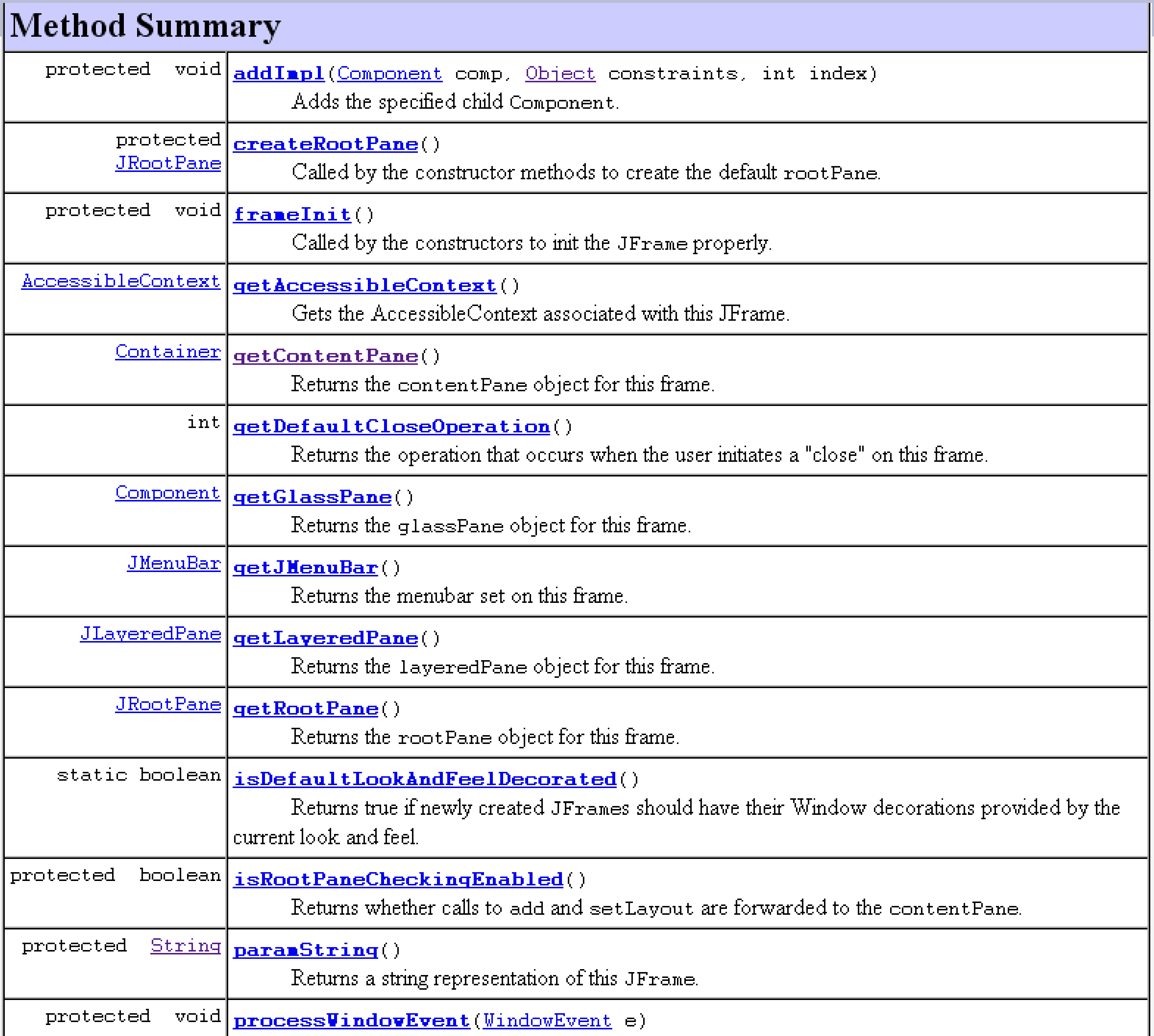
Constructor Summary
`JFrame()` Constructs a new frame that is initially invisible.
`JFrame(GraphicsConfiguration gc)` Creates a Frame in the specified GraphicsConfiguration of a screen device and a blank title.
`JFrame(String title)` Creates a new, initially invisible Frame with the specified title.
`JFrame(String title, GraphicsConfiguration gc)` Creates a JFrame with the specified title and the specified GraphicsConfiguration of a screen device.



```java
package applis;

import javax.swing.JFrame;


public class ApplicationMinimale extends JFrame{

    public ApplicationMinimale(String titre){
        super(titre);
    }

    public static void main(String[] args) {
        ApplicationMinimale test = new ApplicationMinimale("test");
    }
```

Toujours rien

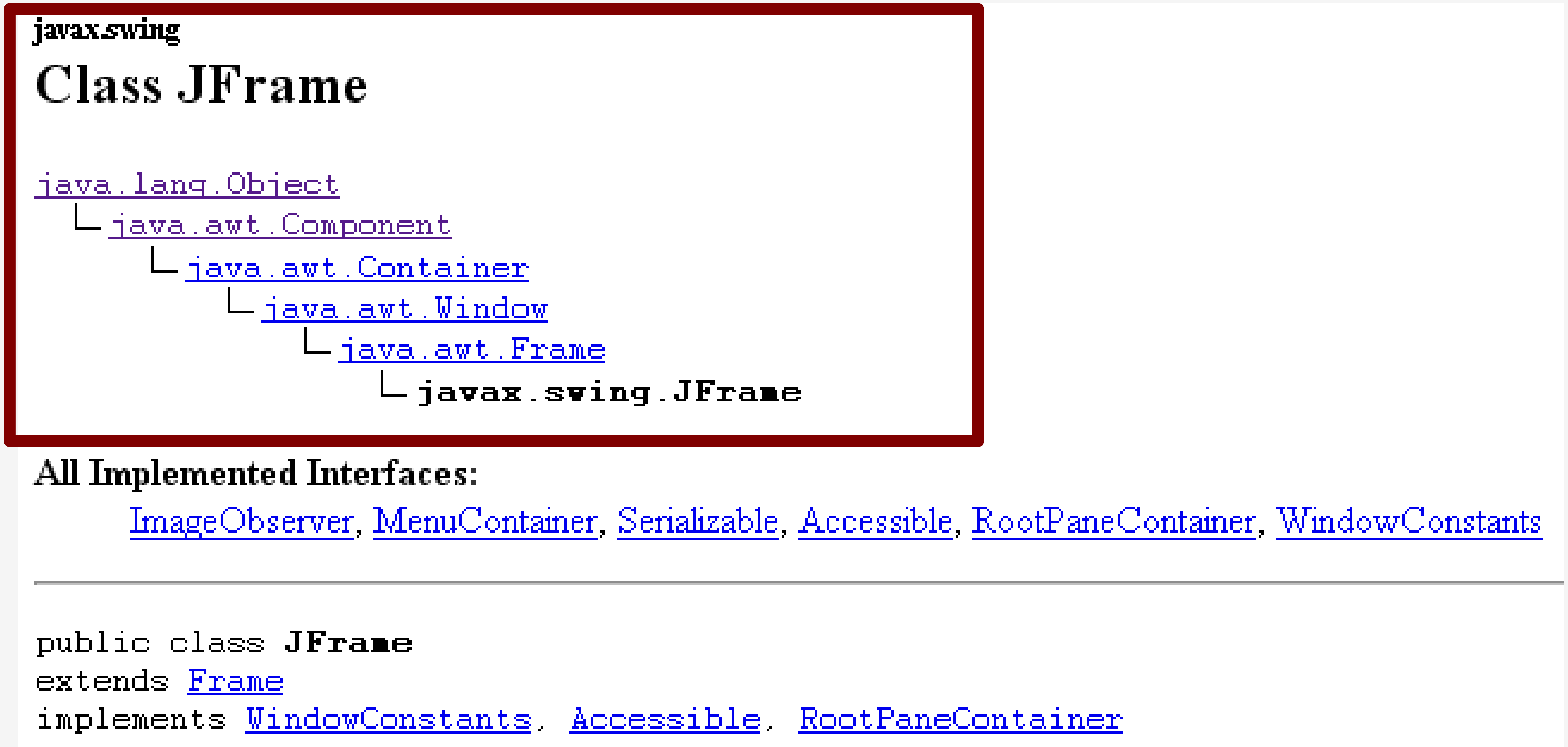
Analyse de la classe JFrame



Method Summary	
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.
AccessibleContext	getAccessibleContext() Gets the AccessibleContext associated with this JFrame.
Container	getContentPane() Returns the contentPane object for this frame.
int	getDefaultCloseOperation() Returns the operation that occurs when the user initiates a "close" on this frame.
Component	getGlassPane() Returns the glassPane object for this frame.
JMenuBar	getJMenuBar() Returns the menubar set on this frame.
JLayeredPane	getLayeredPane() Returns the layeredPane object for this frame.
JRootPane	getRootPane() Returns the rootPane object for this frame.
static boolean	isDefaultLookAndFeelDecorated() Returns true if newly created JFrames should have their Window decorations provided by the current look and feel.
protected boolean	isRootPaneCheckingEnabled() Returns whether calls to add and setLayout are forwarded to the contentPane.
protected String	paramString() Returns a string representation of this JFrame.
protected void	processWindowEvent(WindowEvent e)

Où trouver cette méthode

- Toute classe Java hérite fatalement d'une autre classe (excepté Object)
- Peut aussi implémenter un certain nombre d'interfaces



javax.swing

Class JFrame

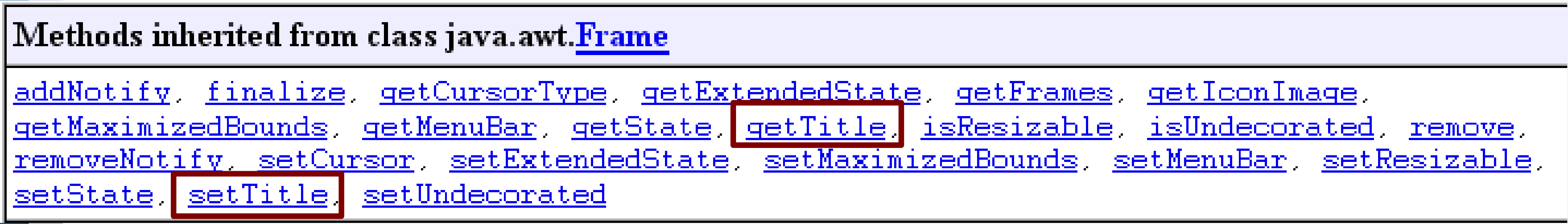
```
java.lang.Object
  └java.awt.Component
      └java.awt.Container
          └java.awt.Window
              └java.awt.Frame
                  └javax.swing.JFrame
```

All Implemented Interfaces:

ImageObserver, MenuContainer, Serializable, Accessible, RootPaneContainer, WindowConstants

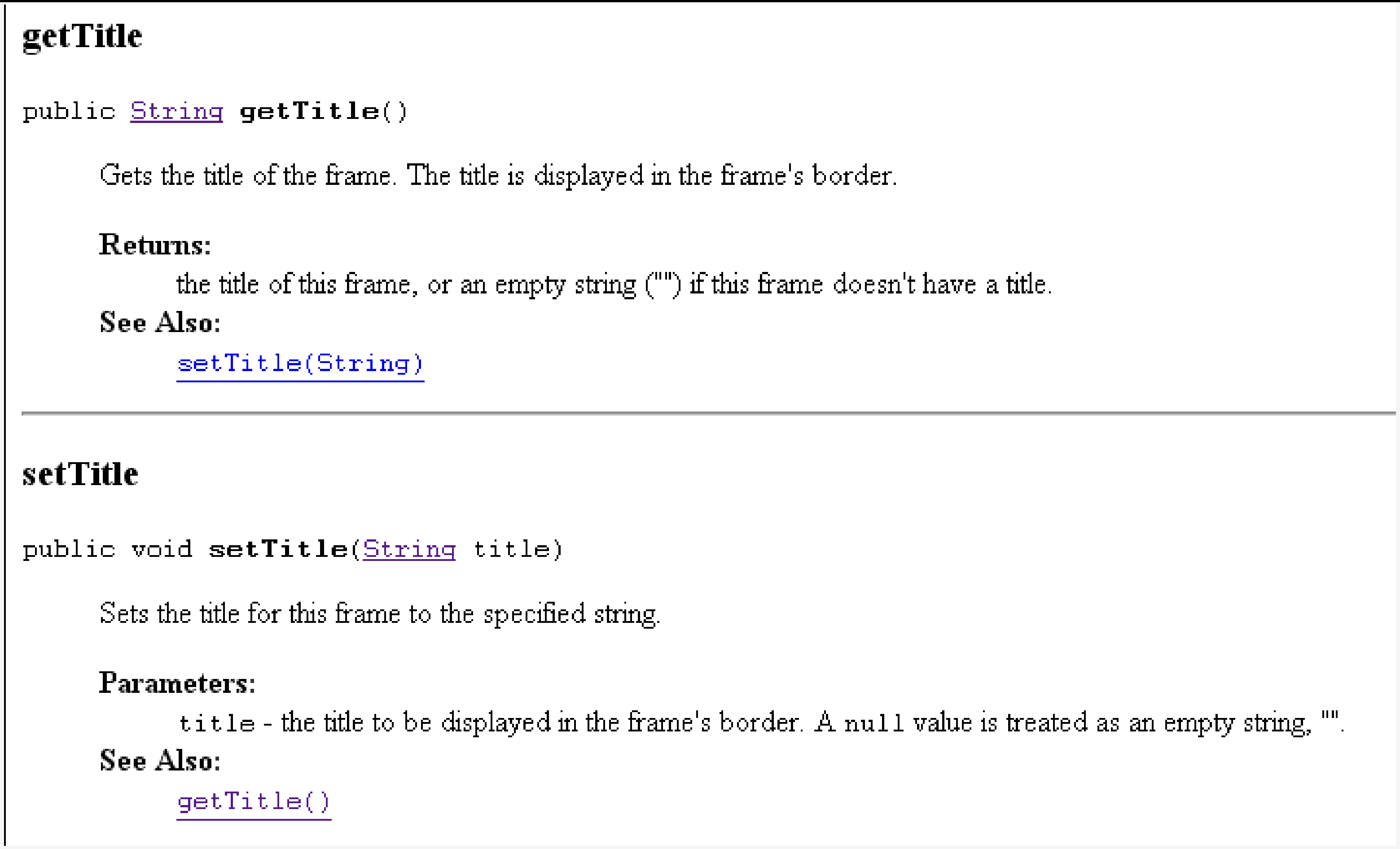
```
public class JFrame
extends Frame
implements WindowConstants, Accessible, RootPaneContainer
```

Héritée de la classe Frame ?



Methods inherited from class java.awt.Frame

addNotify, finalize, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds, getMenuBar, getState, getTitle, isResizable, isUndecorated, remove, removeNotify, setCursor, setExtendedState, setMaximizedBounds, setMenuBar, setResizable, setState, setTitle, setUndecorated



getTitle

```
public String getTitle()
```

Gets the title of the frame. The title is displayed in the frame's border.

Returns:
the title of this frame, or an empty string ("") if this frame doesn't have a title.

See Also:
`setTitle(String)`

setTitle

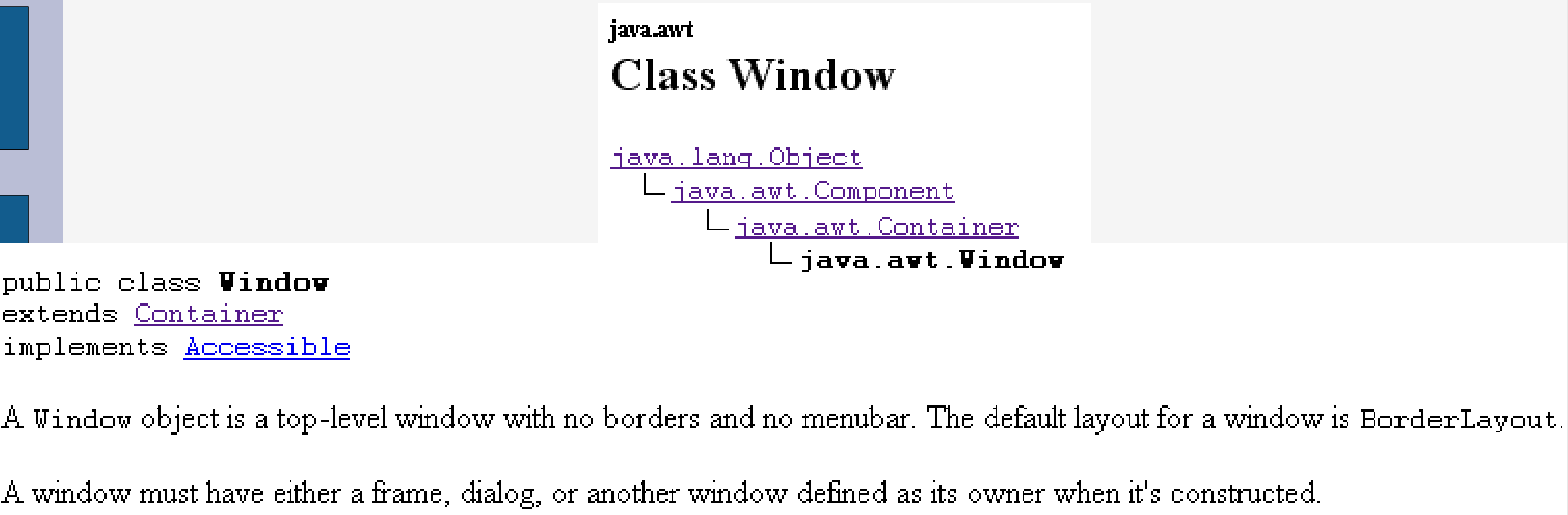
```
public void setTitle(String title)
```

Sets the title for this frame to the specified string.

Parameters:
`title` - the title to be displayed in the frame's border. A `null` value is treated as an empty string, "".

See Also:
`getTitle()`

Héritée de la classe Window ?



java.awt

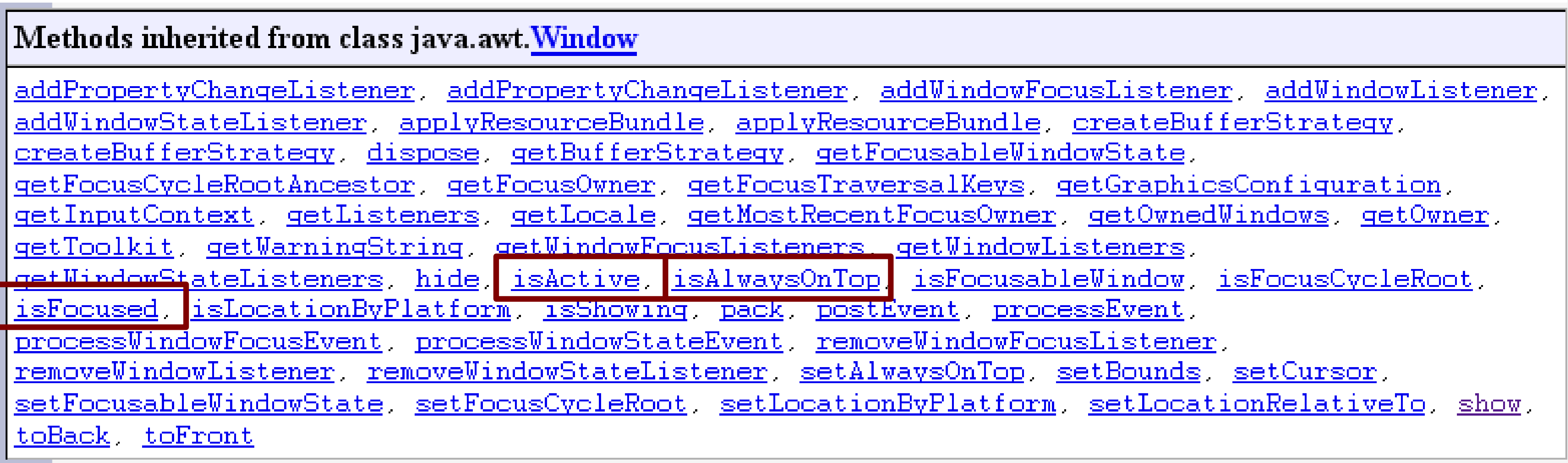
Class Window

```
java.lang.Object
  └ java.awt.Component
      └ java.awt.Container
          └ java.awt.Window
```

```
public class Window
extends Container
implements Accessible
```

A `Window` object is a top-level window with no borders and no menubar. The default layout for a window is `BorderLayout`.

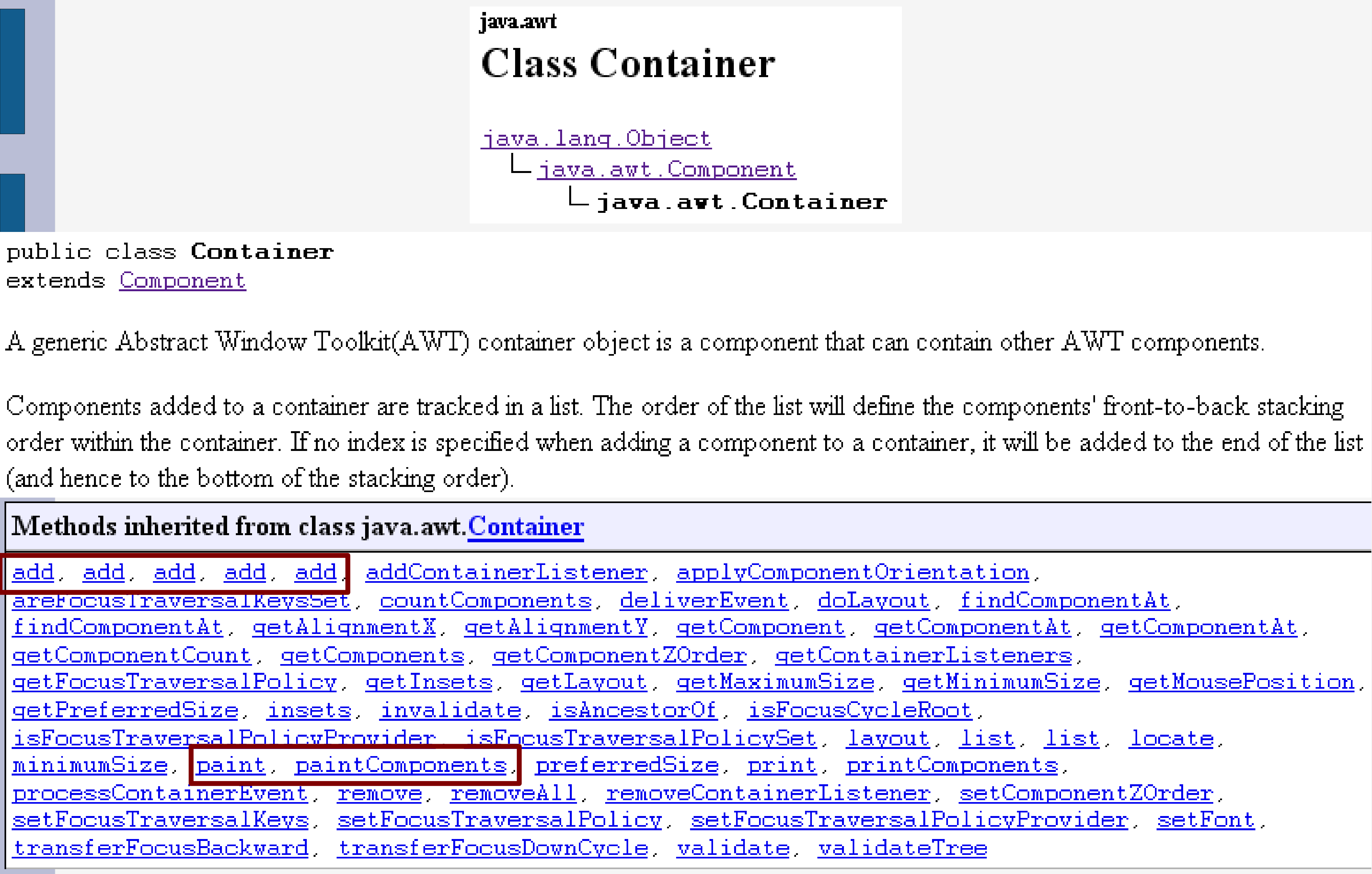
A window must have either a frame, dialog, or another window defined as its owner when it's constructed.



Methods inherited from class java.awt.Window

addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getGraphicsConfiguration, getInputContext, getListeners, getLocale, getMostRecentFocusOwner, getOwnedWindows, getOwner, getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners, getWindowStateListeners, hide, isActive, isAlwaysOnTop, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isShowing, pack, postEvent, processEvent, processWindowFocusEvent, processWindowStateEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, setAlwaysOnTop, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot, setLocationByPlatform, setLocationRelativeTo, show, toBack, toFront

Héritée de la classe Container ?



java.awt

Class Container

```
java.lang.Object
  └ java.awt.Component
      └ java.awt.Container
```

```
public class Container
extends Component
```

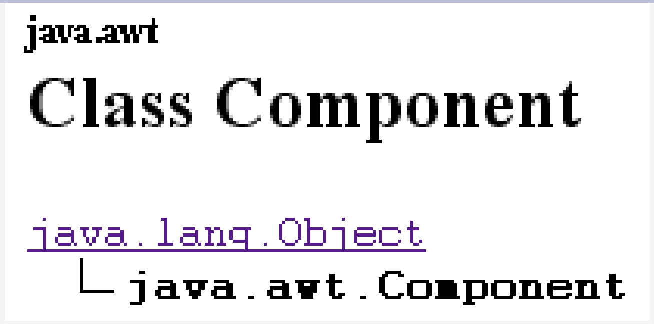
A generic Abstract Window Toolkit(AWT) container object is a component that can contain other AWT components.

Components added to a container are tracked in a list. The order of the list will define the components' front-to-back stacking order within the container. If no index is specified when adding a component to a container, it will be added to the end of the list (and hence to the bottom of the stacking order).

Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalPolicy, getInsets, getLayout, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, processContainerEvent, remove, removeAll, removeContainerListener, setComponentZOrder, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, transferFocusBackward, transferFocusDownCycle, validate, validateTree

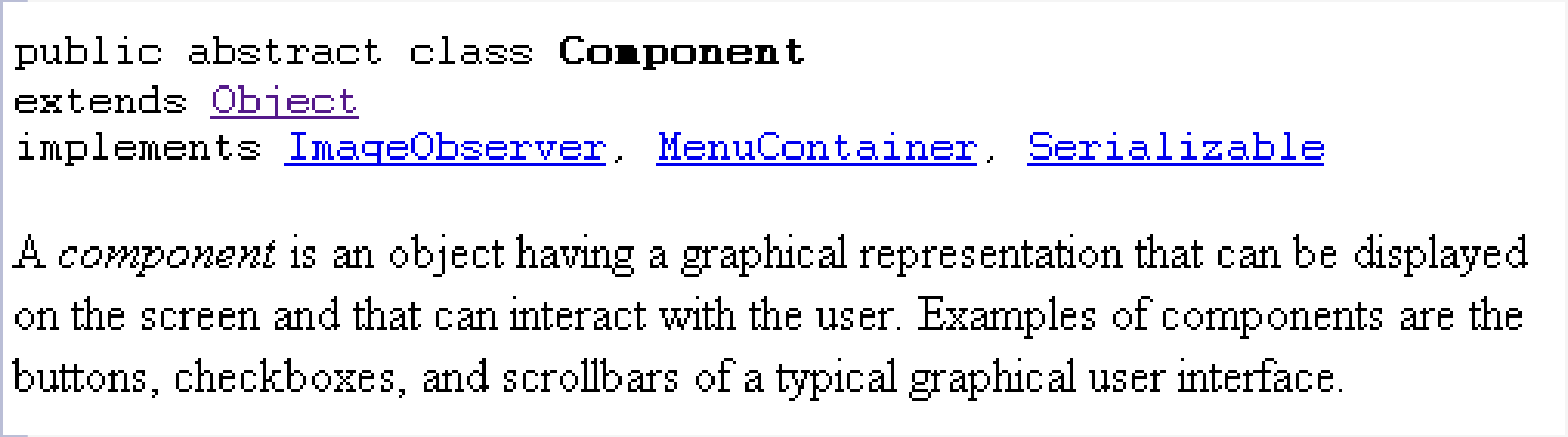
Héritée de la classe Component ?



java.awt

Class Component

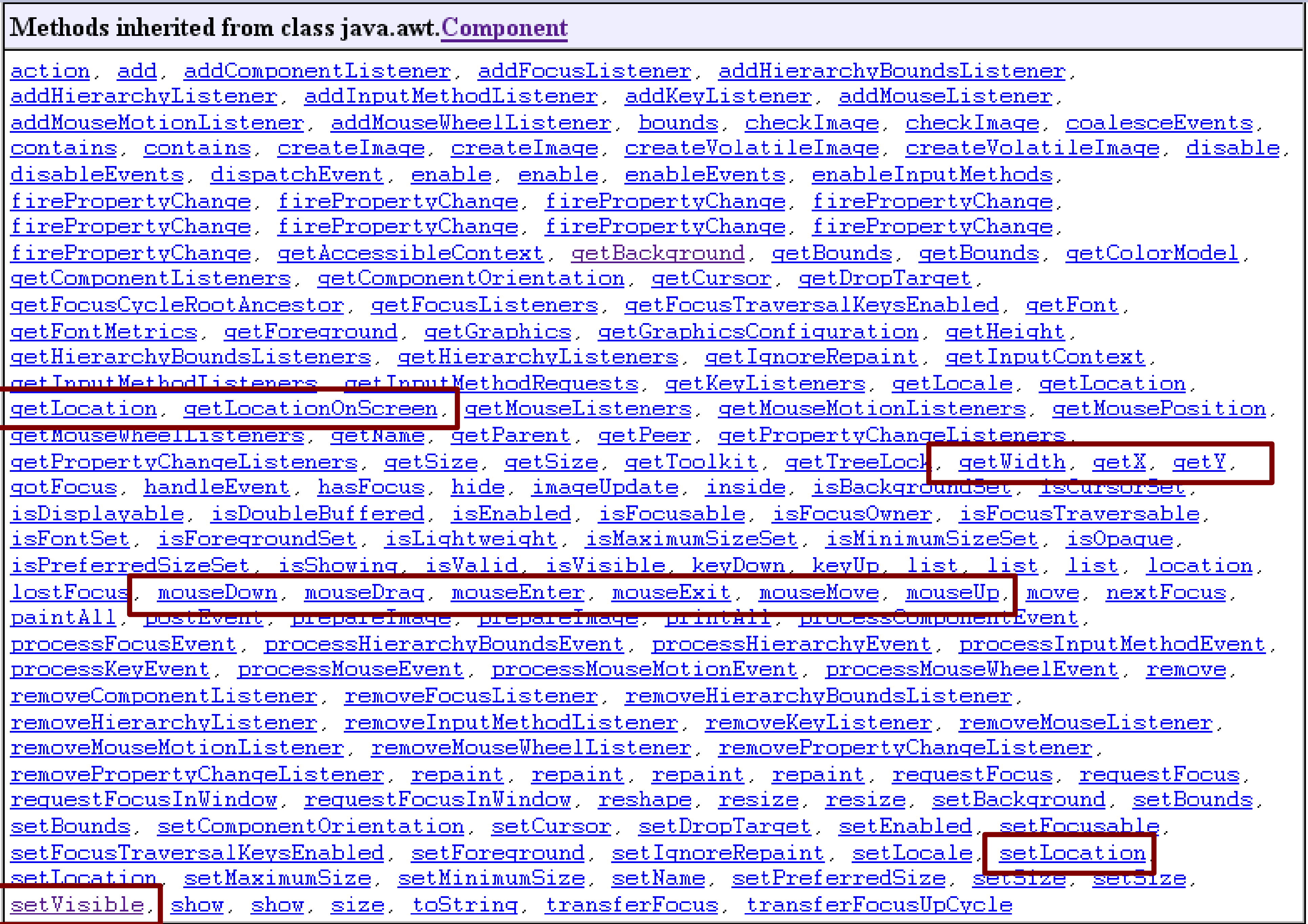
```
java.lang.Object
  └ java.awt.Component
```



```
public abstract class Component
extends Object
implements ImageObserver, MenuContainer, Serializable
```

A *component* is an object having a graphical representation that can be displayed on the screen and that can interact with the user. Examples of components are the buttons, checkboxes, and scrollbars of a typical graphical user interface.

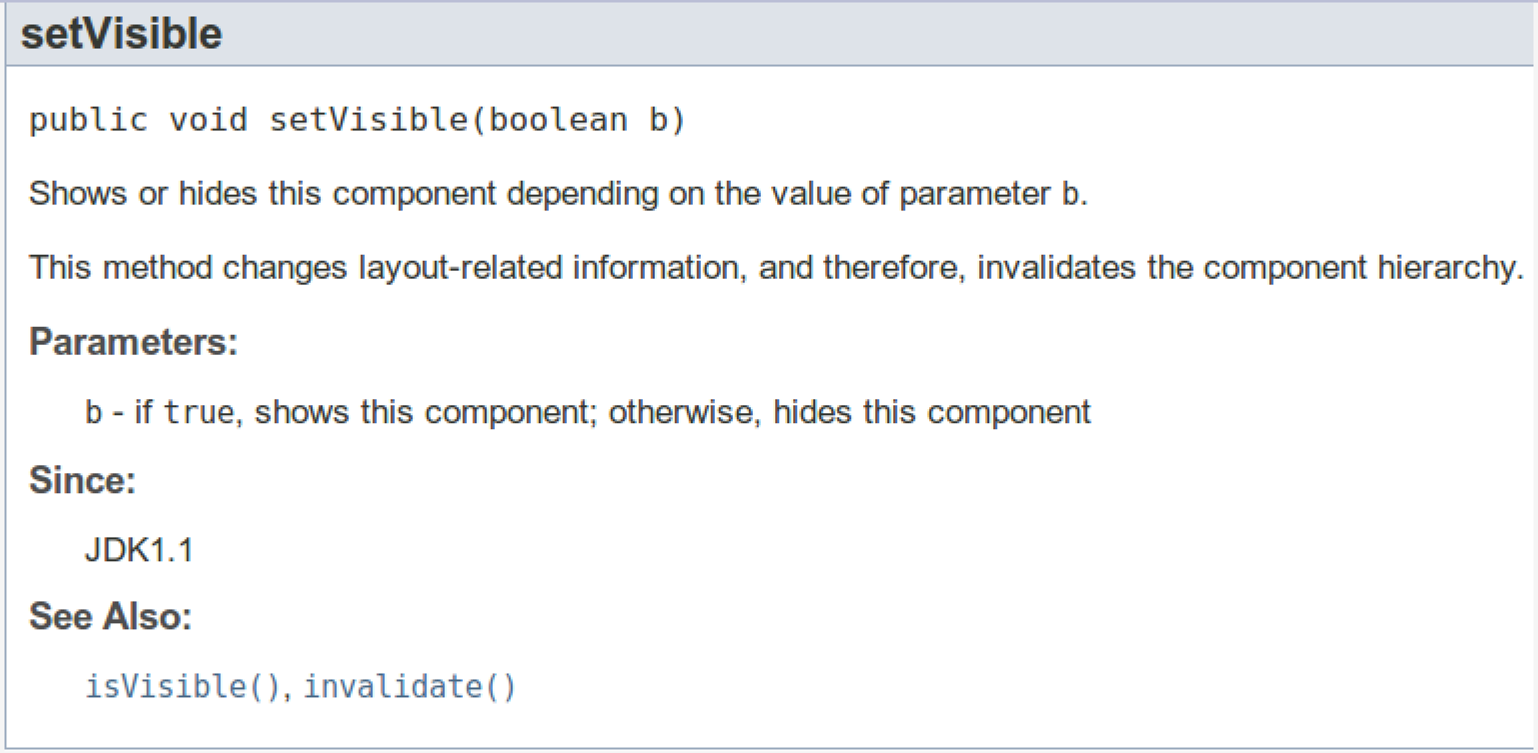
Héritée de la classe Component ?



Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents,
contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, disable,
disableEvents, dispatchEvent, enable, enable, enableEvents, enableInputMethods,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, getAccessibleContext, getBackground, getBounds, getBounds, getColorModel,
getComponentListeners, getComponentOrientation, getCursor, getDropTarget,
getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeysEnabled, getFont,
getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext,
getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocale, getLocation,
getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMousePosition,
getMouseWheelListeners, getName, getParent, getPeer, getPropertyChangeListeners,
getPropertyChangeListeners, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY,
gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet, isCursorSet,
isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusOwner, isFocusTraversable,
isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque,
isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list, location,
lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus,
paintAll, postEvent, prepareImage, prepareImage, printAll, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent,
processKeyEvent, processMouseEvent, processMouseMotionEvent, processMouseWheelEvent, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener,
removeMouseMotionListener, removeMouseWheelListener, removePropertyChangeListener,
removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocus,
requestFocusInWindow, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds,
setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable,
setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint, setLocale, setLocation,
setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize,
setVisible, show, show, size, toString, transferFocus, transferFocusUpCycle

Dans la classe java.awt.Component



setVisible

`public void setVisible(boolean b)`

Shows or hides this component depending on the value of parameter b.

This method changes layout-related information, and therefore, invalidates the component hierarchy.

Parameters:

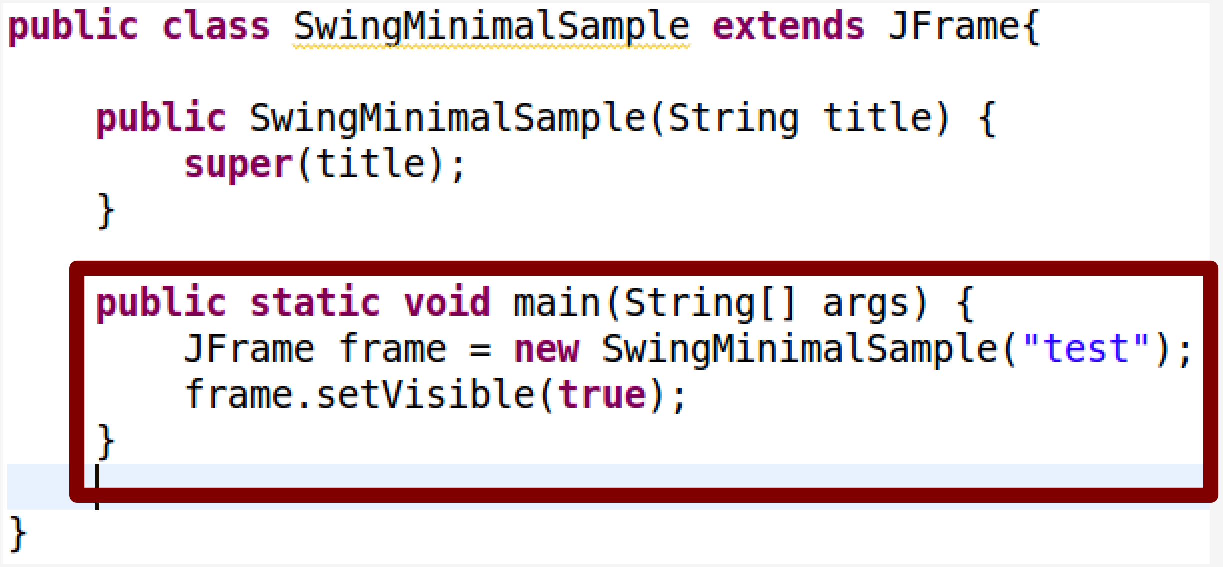
b - if `true`, shows this component; otherwise, hides this component

Since:

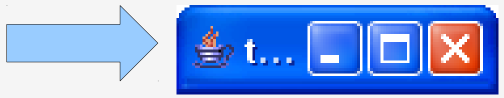
JDK1.1

See Also:

`isVisible()`, `invalidate()`

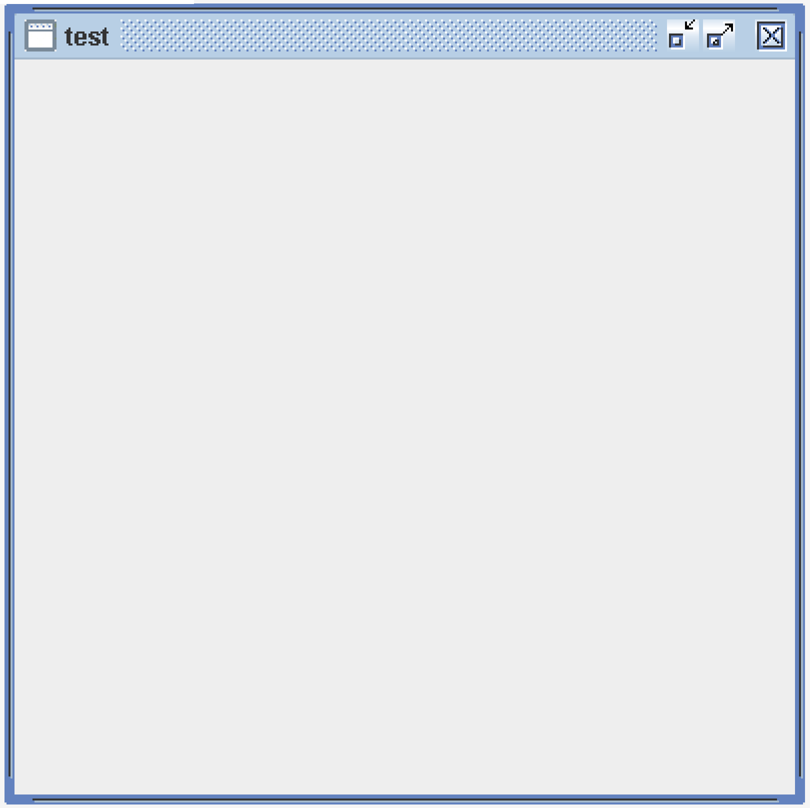


```java
public class SwingMinimalSample extends JFrame{

    public SwingMinimalSample(String title) {
        super(title);
    }

    public static void main(String[] args) {
        JFrame frame = new SwingMinimalSample("test");
        frame.setVisible(true);
    }

}
```



Code minimal

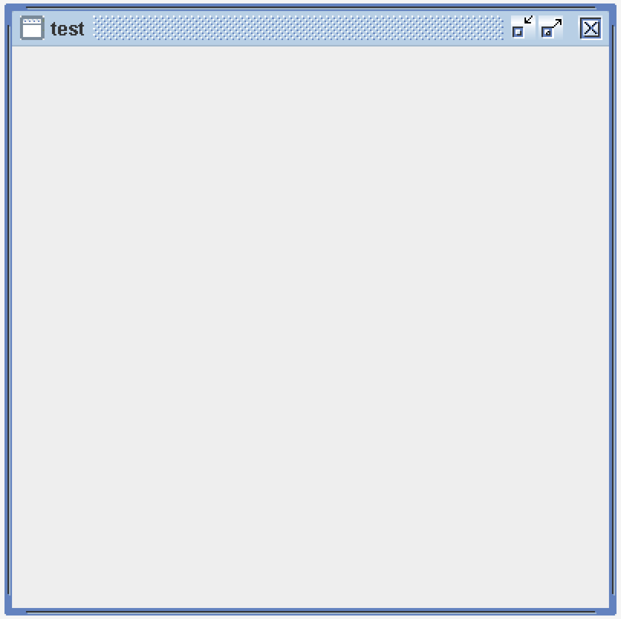
```
public SwingMinimalSample(String title) {
    super(title);
}

public static void main(String[] args) {
    JFrame frame = new SwingMinimalSample("test");
    frame.setSize(400, 400);
    frame.setLocation(200, 200);
    frame.setVisible(true);
}
```



java.awt.Window.setLocationRelativeTo(Component)

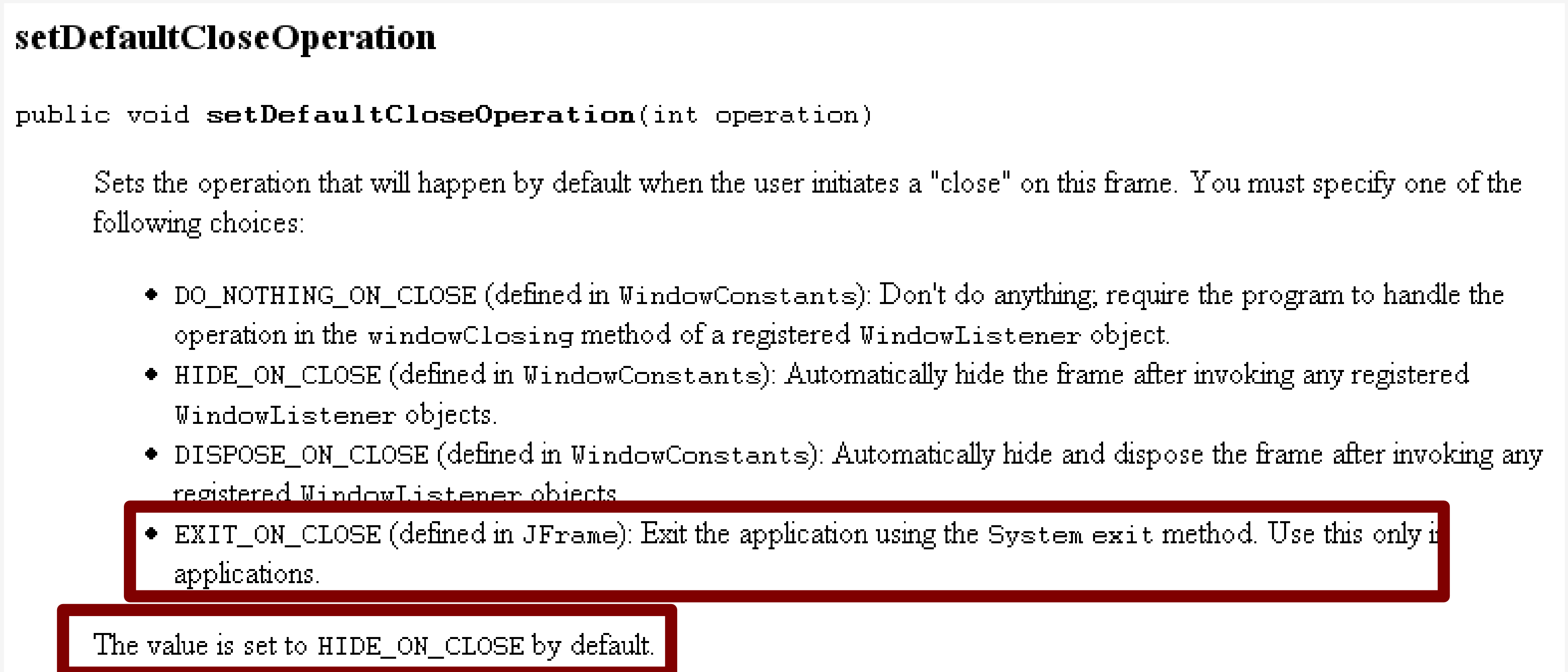
- Centré relativement à un autre composant
- ***null*** → centré au milieu de l'écran

```java
public static void main(String[] args) {
    JFrame frame = new SwingMinimalSample("test");
    frame.setSize(400, 400);
    frame.setLocationRelativeTo(null);
    frame.setVisible(true);
}
```



Clique sur le bouton fermer

- La fenêtre disparaît
- Mais la machine virtuelle est toujours active!
  - (l'application n'est pas terminée) : la fenêtre est simplement cachée



setDefaultCloseOperation

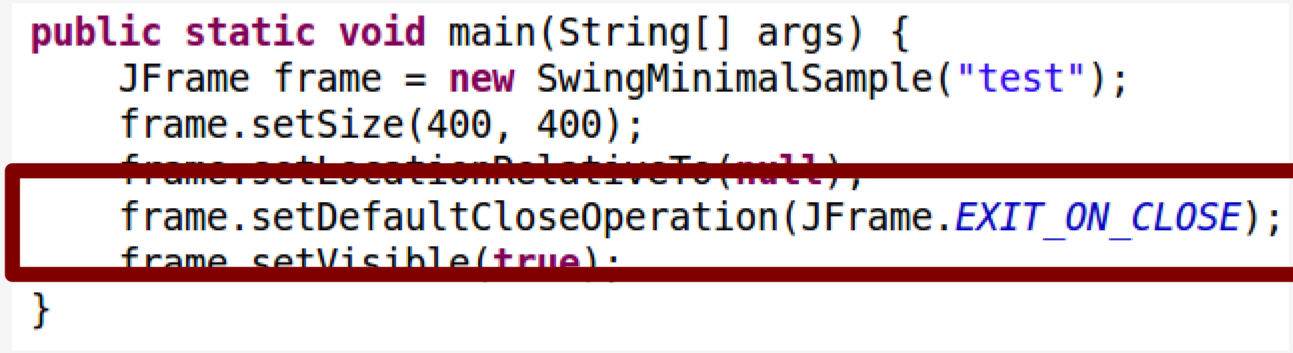
```
public void setDefaultCloseOperation(int operation)
```

Sets the operation that will happen by default when the user initiates a "close" on this frame. You must specify one of the following choices:

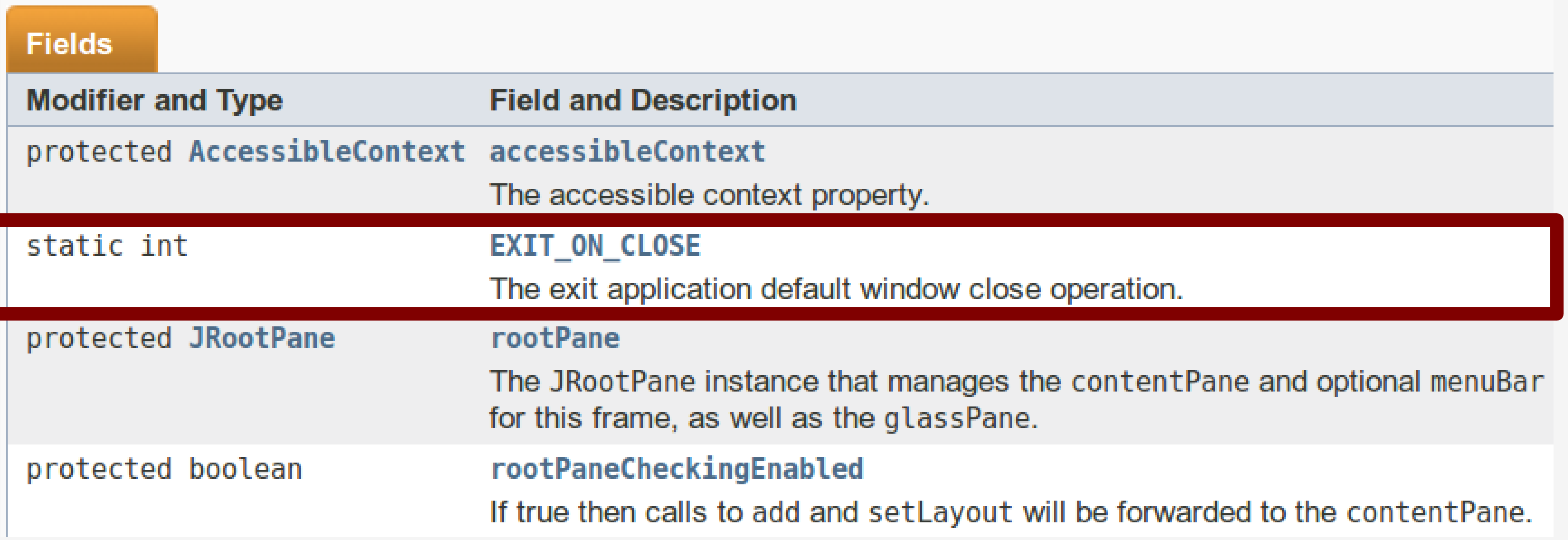
- `DO_NOTHING_ON_CLOSE` (defined in `WindowConstants`): Don't do anything; require the program to handle the operation in the `windowClosing` method of a registered `WindowListener` object.
- `HIDE_ON_CLOSE` (defined in `WindowConstants`): Automatically hide the frame after invoking any registered `WindowListener` objects.
- `DISPOSE_ON_CLOSE` (defined in `WindowConstants`): Automatically hide and dispose the frame after invoking any registered `WindowListener` objects.
- `EXIT_ON_CLOSE` (defined in `JFrame`): Exit the application using the `System` `exit` method. Use this only i applications.

The value is set to `HIDE_ON_CLOSE` by default.

Fermeture de type « application »



```
public static void main(String[] args) {
    JFrame frame = new SwingMinimalSample("test");
    frame.setSize(400, 400);
    frame.setLocationRelativeTo(null);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true);
}
```

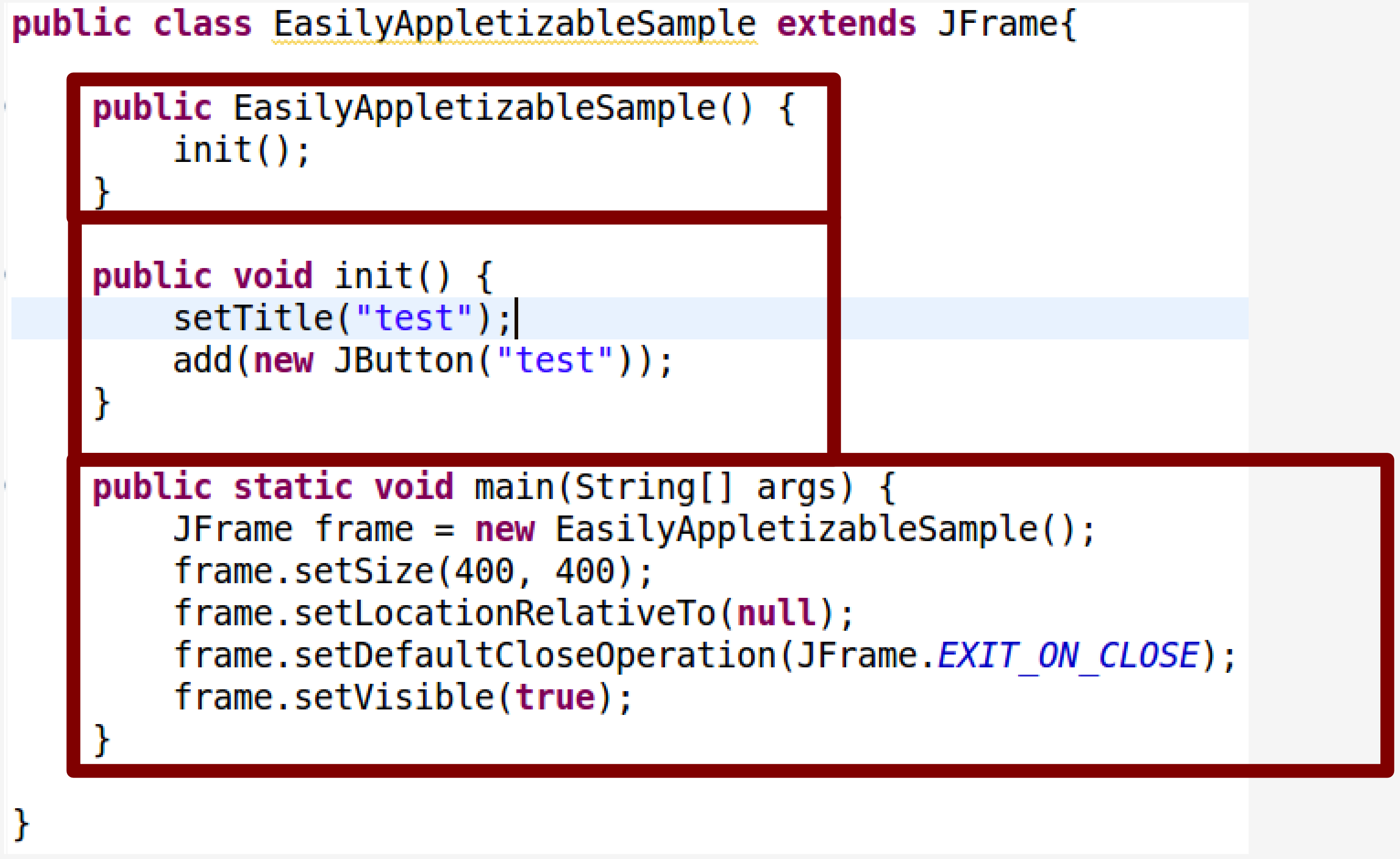


Fields

Modifier and Type	Field and 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`	`rootPaneCheckingEnabled` If true then calls to `add` and `setLayout` will be forwarded to the `contentPane`.

Faciliter la conversion en applet

- Mettre les opérations (compatibles) du constructeur dans une méthode **init**(): facilite la transformation en applet de l'application



```
public class EasilyAppletizableSample extends JFrame{

    public EasilyAppletizableSample() {
        init();
    }

    public void init() {
        setTitle("test");
        add(new JButton("test"));
    }

    public static void main(String[] args) {
        JFrame frame = new EasilyAppletizableSample();
        frame.setSize(400, 400);
        frame.setLocationRelativeTo(null);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

}
```

Élaboration d'une JFrame

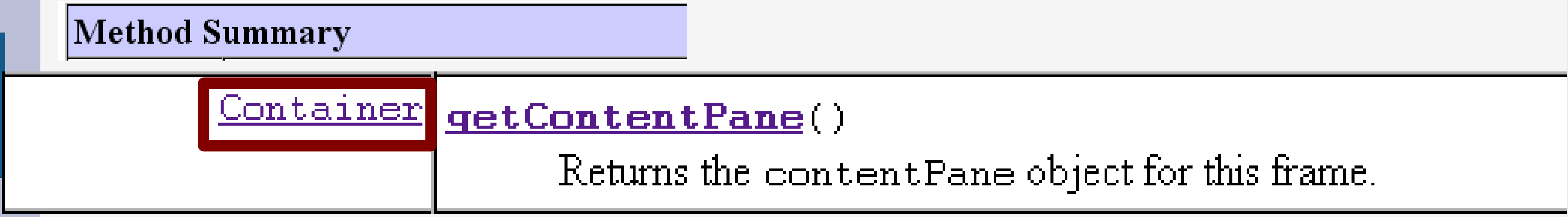
- Une JFrame comporte essentiellement deux parties:
  - Les menus et barres d'outils, situés par défaut sous le titre
  - La zone contenant les éléments graphiques de l'application
- Commençons par décrire la manière dont fonctionne la **zone des éléments graphiques**

javax.swing.JPanel

Le ContentPane de JFrame

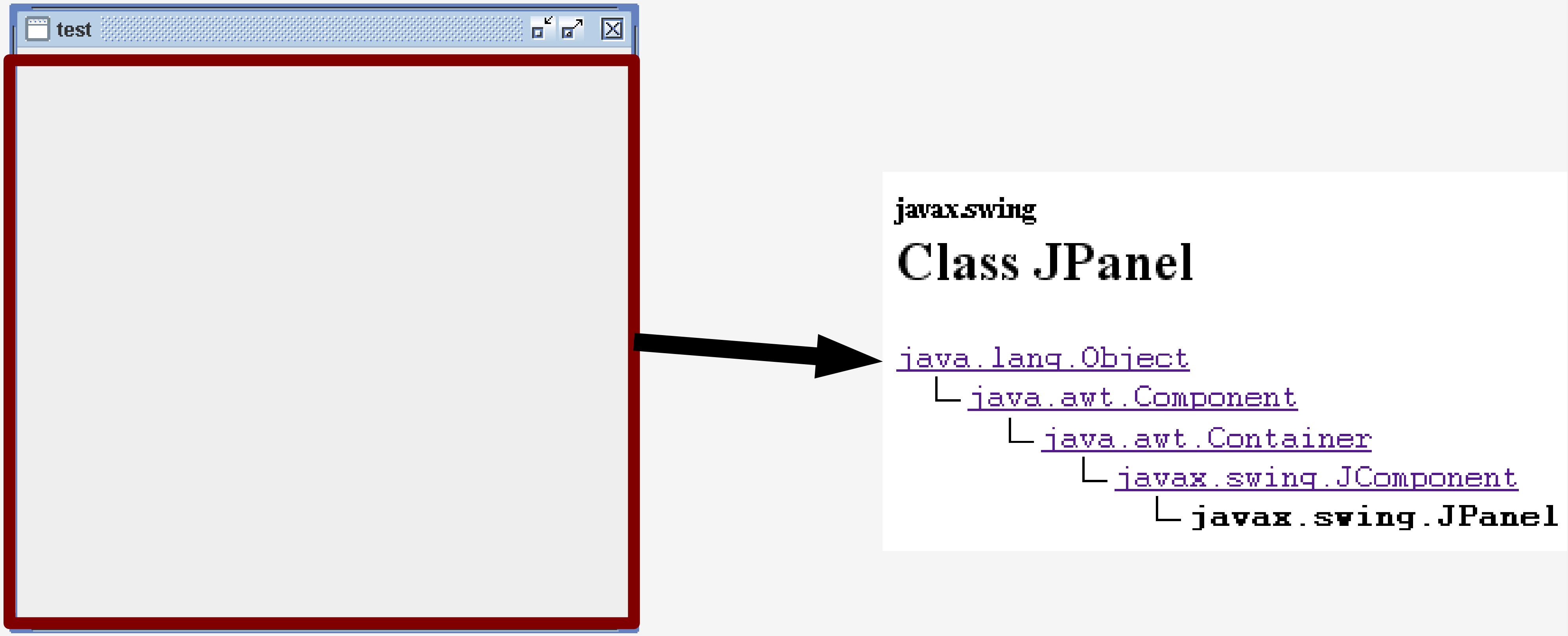
javax.swing

Class JFrame



Method Summary

Container	**getContentPane**() Returns the contentPane object for this frame.

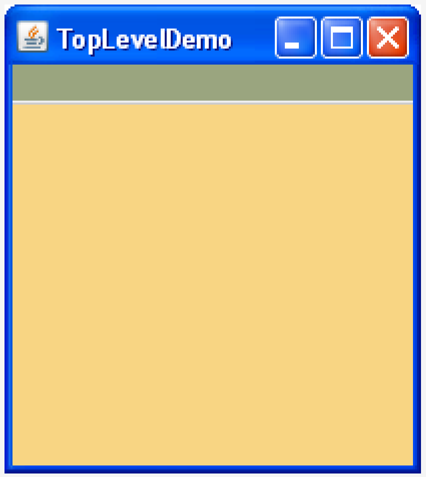


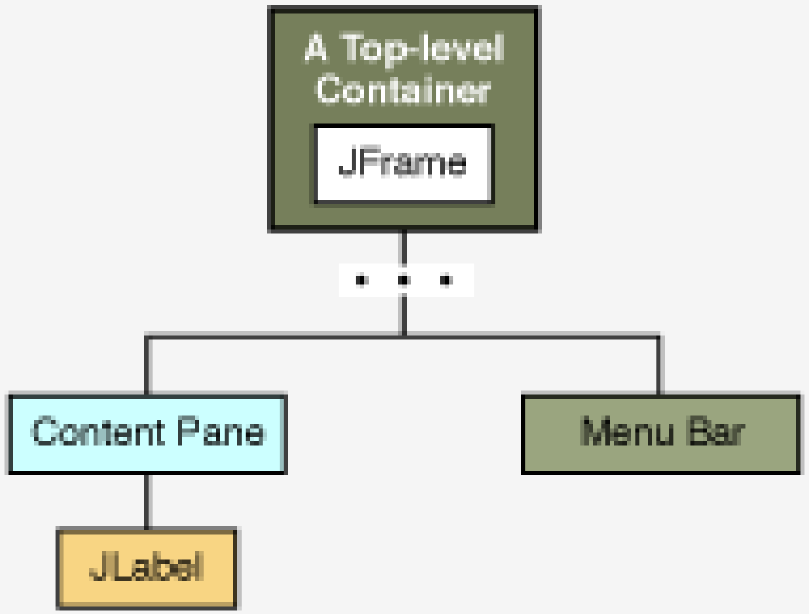
javax.swing

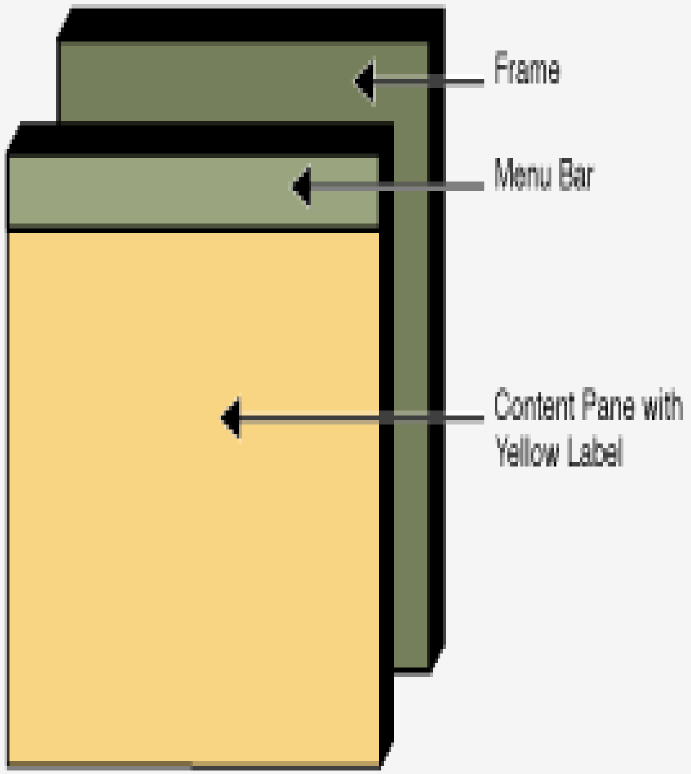
Class JPanel

```
java.lang.Object
  └ java.awt.Component
      └ java.awt.Container
          └ javax.swing.JComponent
              └ javax.swing.JPanel
```

JFrame

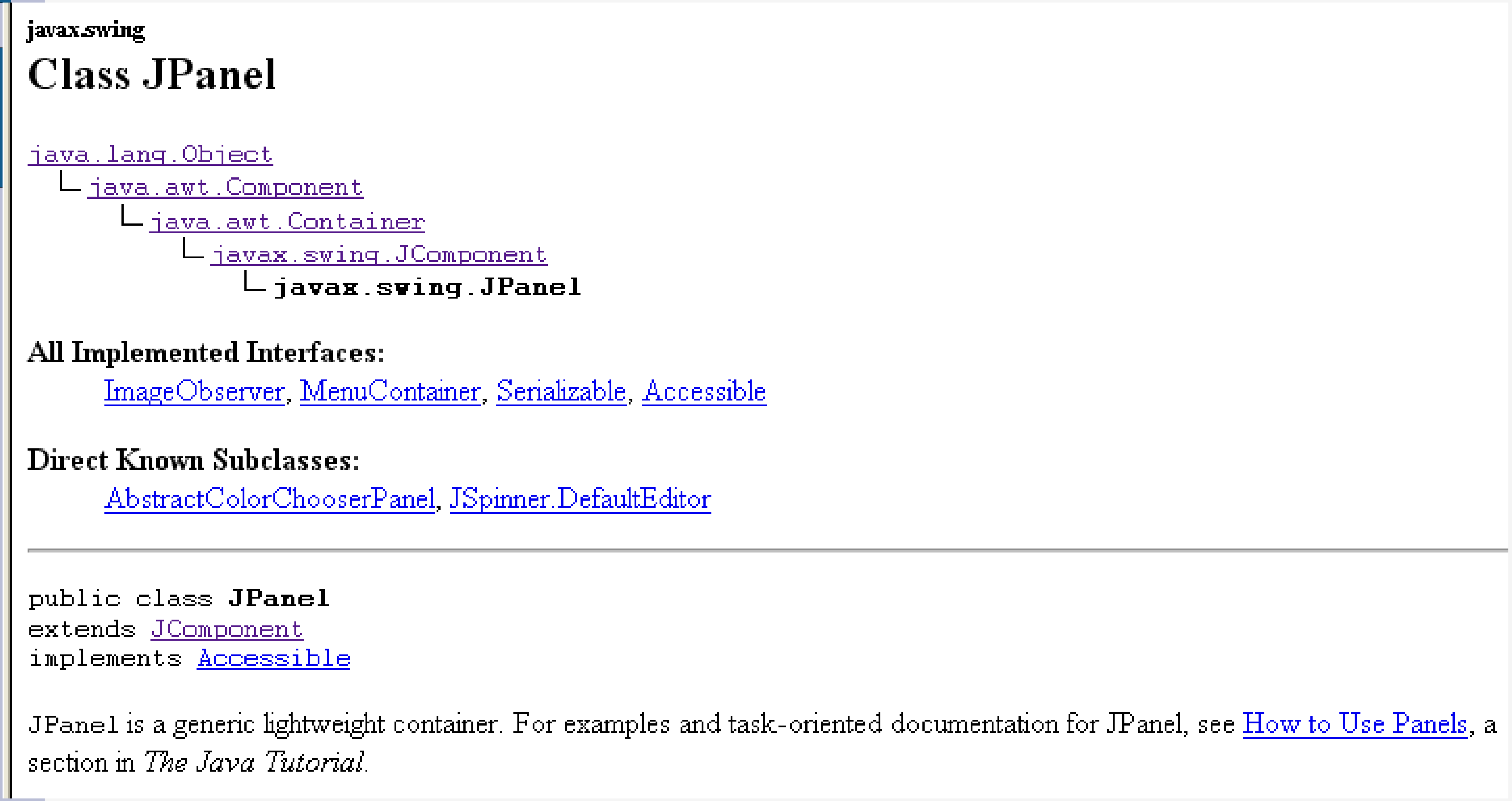

TopLevelDemo


A Top-level Container
JFrame
Content Pane
Menu Bar
JLabel


Frame
Menu Bar
Content Pane with Yellow Label

Le composant de base : JPanel

- JPanel : « un cadre vide »



javax.swing

Class JPanel

```
java.lang.Object
  └ java.awt.Component
      └ java.awt.Container
          └ javax.swing.JComponent
              └ javax.swing.JPanel
```

All Implemented Interfaces:

ImageObserver, MenuContainer, Serializable, Accessible

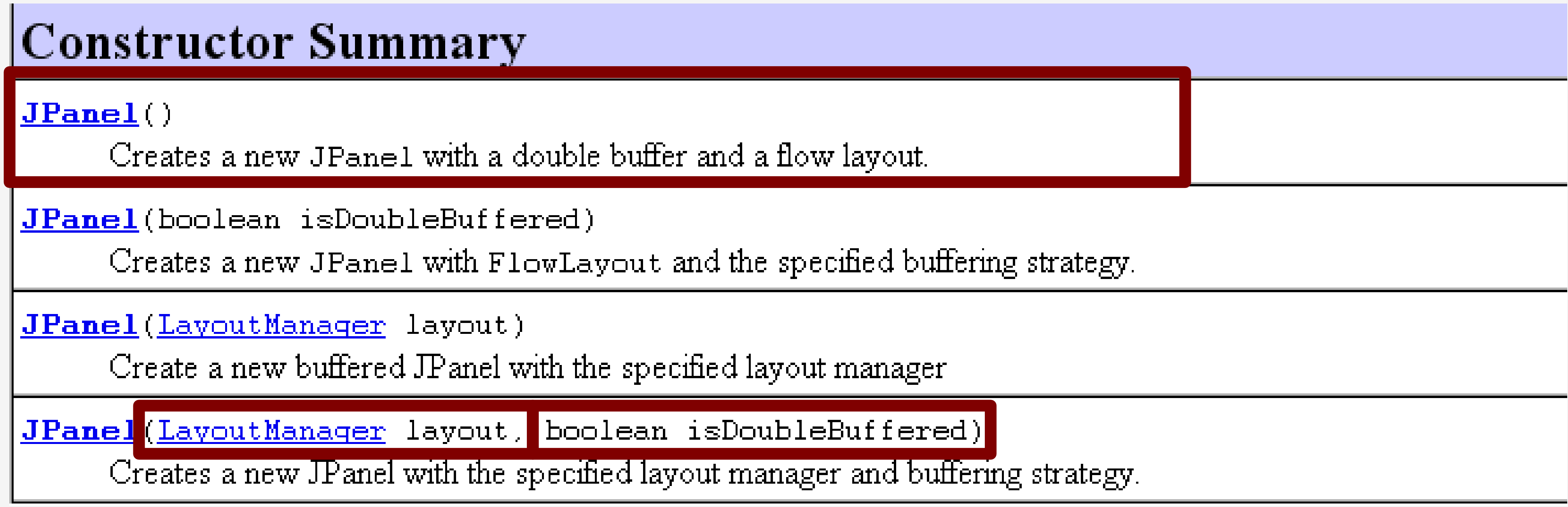
Direct Known Subclasses:

AbstractColorChooserPanel, JSpinner.DefaultEditor

```
public class JPanel
extends JComponent
implements Accessible
```

JPanel is a generic lightweight container. For examples and task-oriented documentation for JPanel, see How to Use Panels, a section in *The Java Tutorial*.

Le composant de base : JPanel

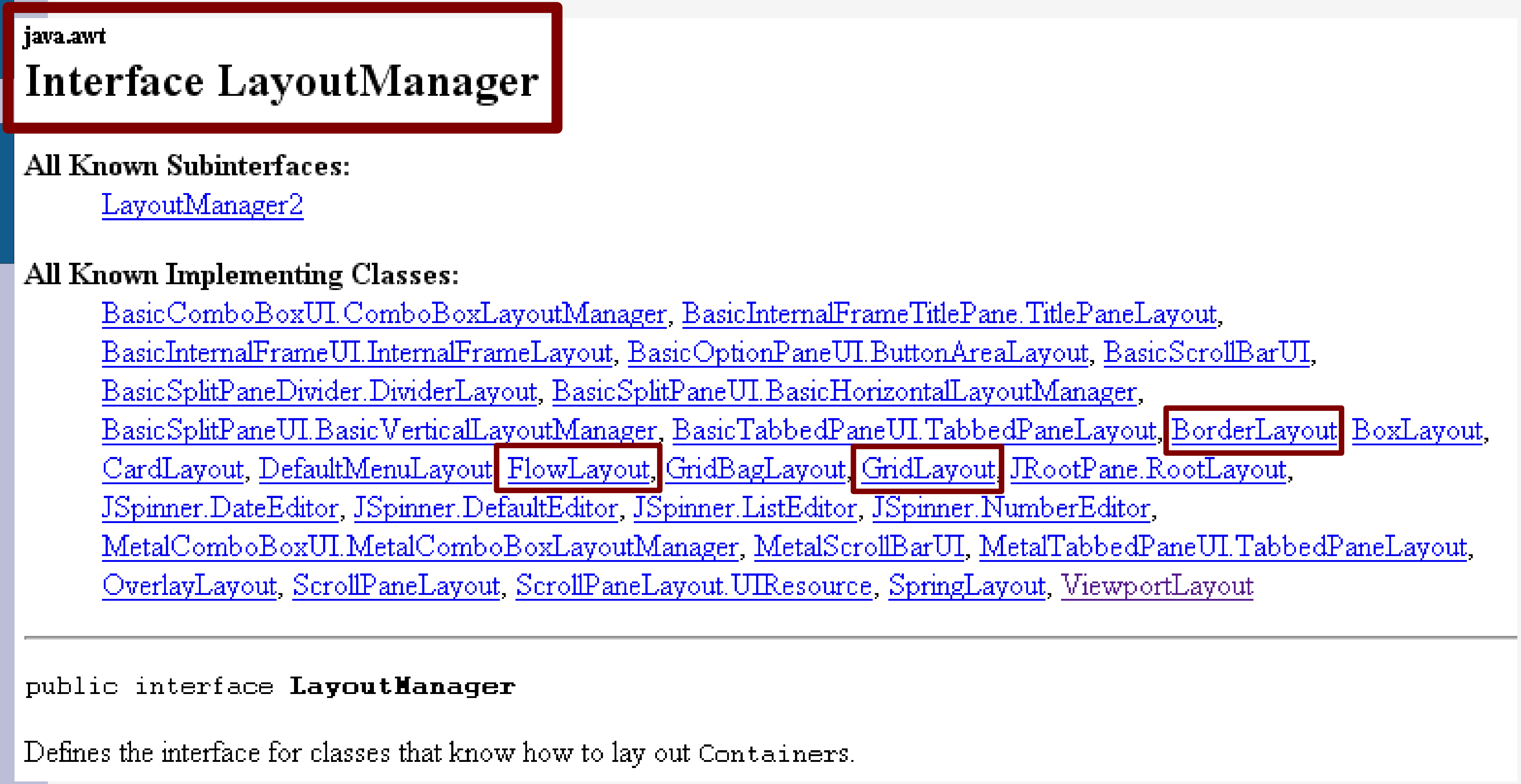


Constructor Summary
`JPanel()` Creates a new `JPanel` with a double buffer and a flow layout.
`JPanel(boolean isDoubleBuffered)` Creates a new `JPanel` with `FlowLayout` and the specified buffering strategy.
`JPanel(LayoutManager layout)` Create a new buffered JPanel with the specified layout manager
`JPanel(LayoutManager layout, boolean isDoubleBuffered)` Creates a new JPanel with the specified layout manager and buffering strategy.

- JPanel possède deux principales propriétés :
  - La technique d'affichage utilisée (double buffer ou pas)
  - Le gestionnaire de mise en page utilisé : le **LayoutManager**

Les gestionnaires de mise en page

Les différents LayoutManager



java.awt

Interface LayoutManager

All Known Subinterfaces:

LayoutManager2

All Known Implementing Classes:

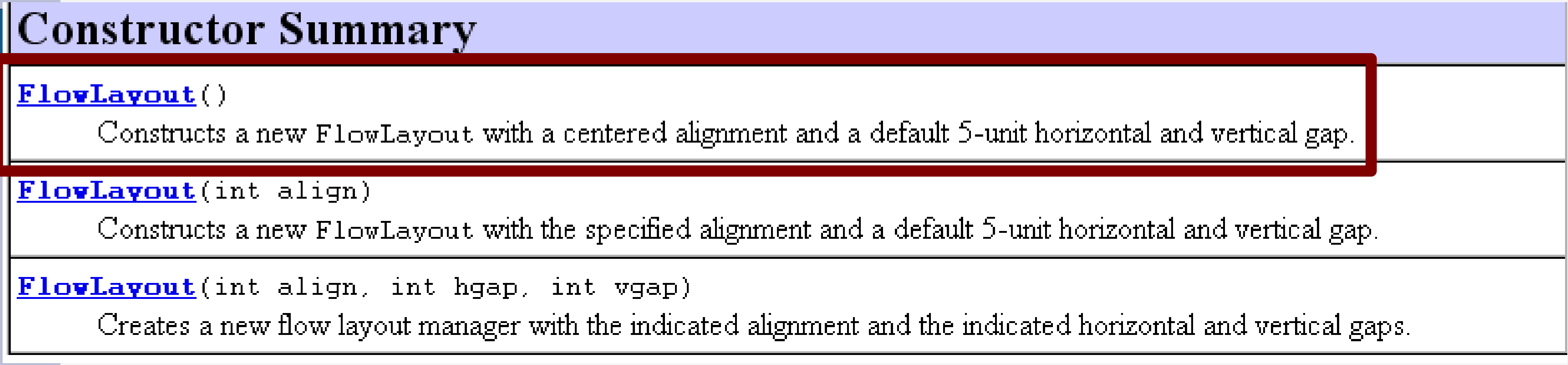
BasicComboBoxUI.ComboBoxLayoutManager, BasicInternalFrameTitlePane.TitlePaneLayout, BasicInternalFrameUI.InternalFrameLayout, BasicOptionPaneUI.ButtonAreaLayout, BasicScrollBarUI, BasicSplitPaneDivider.DividerLayout, BasicSplitPaneUI.BasicHorizontalLayoutManager, BasicSplitPaneUI.BasicVerticalLayoutManager, BasicTabbedPaneUI.TabbedPaneLayout, BorderLayout, BoxLayout, CardLayout, DefaultMenuLayout, FlowLayout, GridBagLayout, GridLayout, JRootPane.RootLayout, JSpinner.DateEditor, JSpinner.DefaultEditor, JSpinner.ListEditor, JSpinner.NumberEditor, MetalComboBoxUI.MetalComboBoxLayoutManager, MetalScrollBarUI, MetalTabbedPaneUI.TabbedPaneLayout, OverlayLayout, ScrollPaneLayout, ScrollPaneLayout.UIResource, SpringLayout, ViewportLayout

```
public interface LayoutManager
```

Defines the interface for classes that know how to lay out `Containers`.

FlowLayout

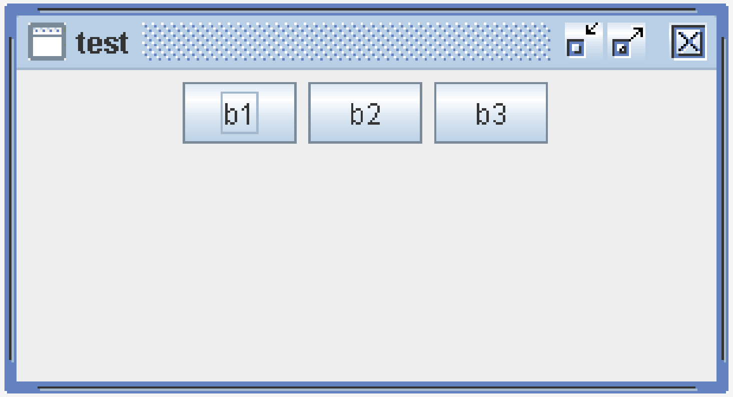
- **FlowLayout** : dispose les composants en ligne les uns après les autres, quand une ligne est pleine les composants suivants sont placés sur la ligne suivante.



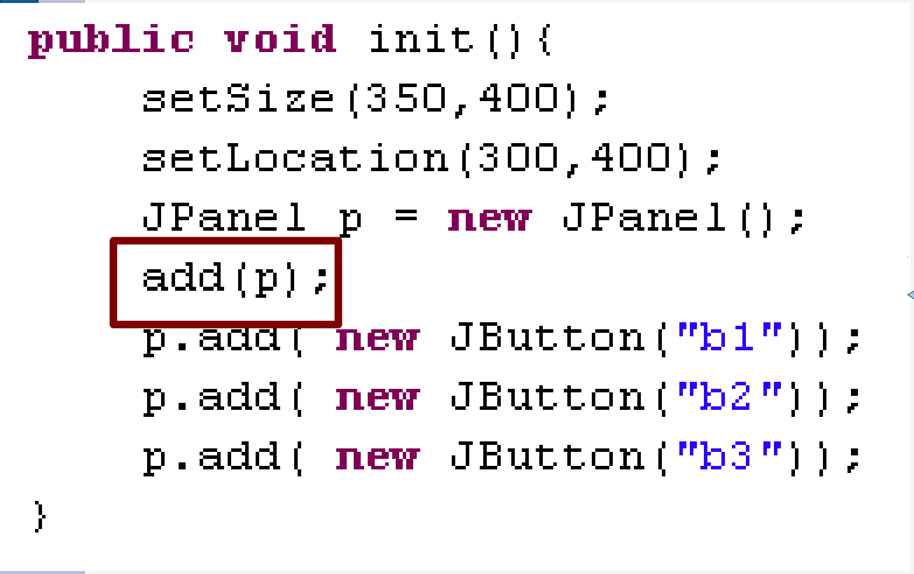
Constructor Summary

FlowLayout()   Constructs a new FlowLayout with a centered alignment and a default 5-unit horizontal and vertical gap.
FlowLayout(int align)   Constructs a new FlowLayout with the specified alignment and a default 5-unit horizontal and vertical gap.
FlowLayout(int align, int hgap, int vgap)   Creates a new flow layout manager with the indicated alignment and the indicated horizontal and vertical gaps.

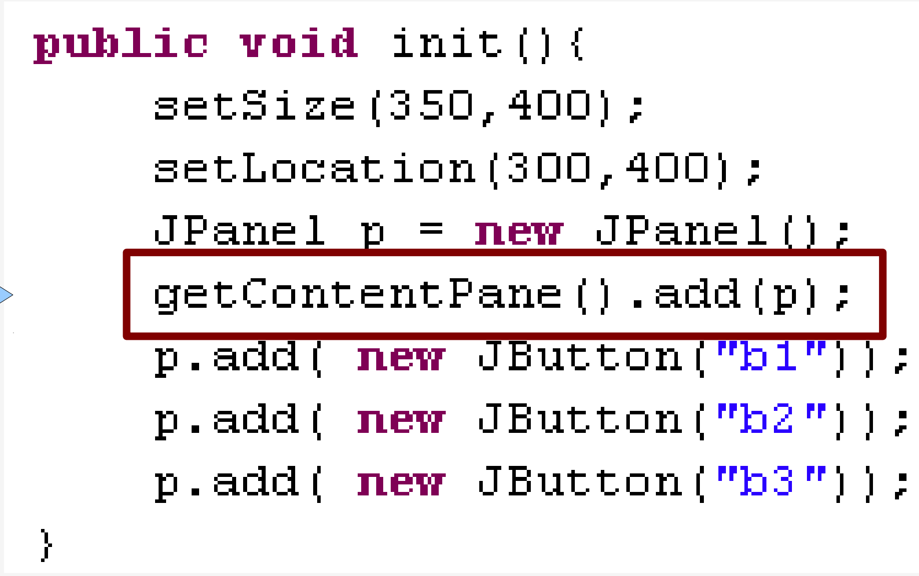
```java
public void init(){
    setSize(350,400);
    setLocation(300,400);
    JPanel p = new JPanel();
    add(p);
    p.add( new JButton("b1"));
    p.add( new JButton("b2"));
    p.add( new JButton("b3"));
}
```



Petite parenthèse



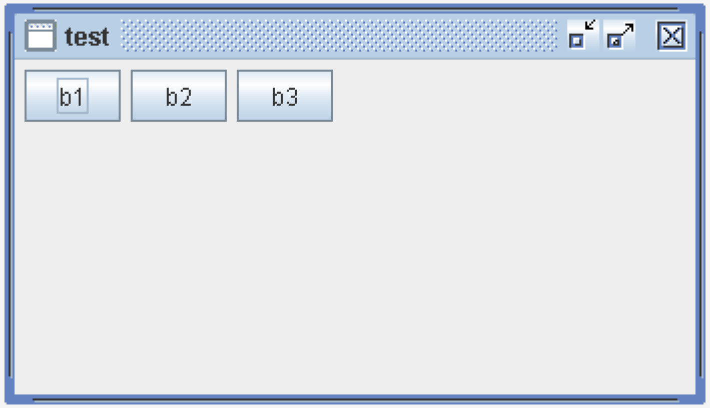
```java
public void init(){
    setSize(350,400);
    setLocation(300,400);
    JPanel p = new JPanel();
    add(p);
    p.add( new JButton("b1"));
    p.add( new JButton("b2"));
    p.add( new JButton("b3"));
}
```



```java
public void init(){
    setSize(350,400);
    setLocation(300,400);
    JPanel p = new JPanel();
    getContentPane().add(p);
    p.add( new JButton("b1"));
    p.add( new JButton("b2"));
    p.add( new JButton("b3"));
}
```

FlowLayout

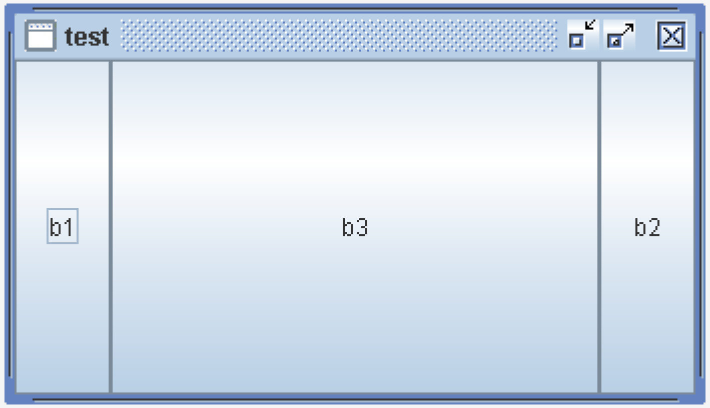
```
public void init(){
    setSize(350,400);
    setLocation(300,400);
    JPanel p = new JPanel(new FlowLayout(FlowLayout.LEFT));
    getContentPane().add(p);
    p.add( new JButton("b1"));
    p.add( new JButton("b2"));
    p.add( new JButton("b3"));
}
```



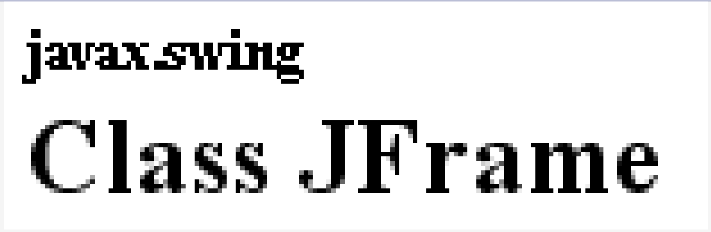
BorderLayout

- Divise le composant en 5 régions : Center, South, North, West et East.

```
public void init(){
      setSize(350,200);
      setLocation(300,400);
      JPanel p = new JPanel(new BorderLayout());
      getContentPane().add(p);
      p.add(BorderLayout.WEST, new JButton("b1"));
      p.add(BorderLayout.EAST, new JButton("b2"));
      p.add( new JButton("b3"));
}
```

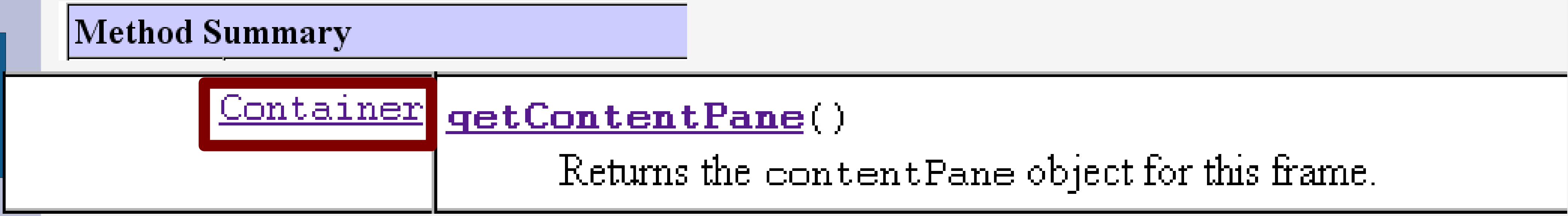


Rappel: Le ContentPane de JFrame



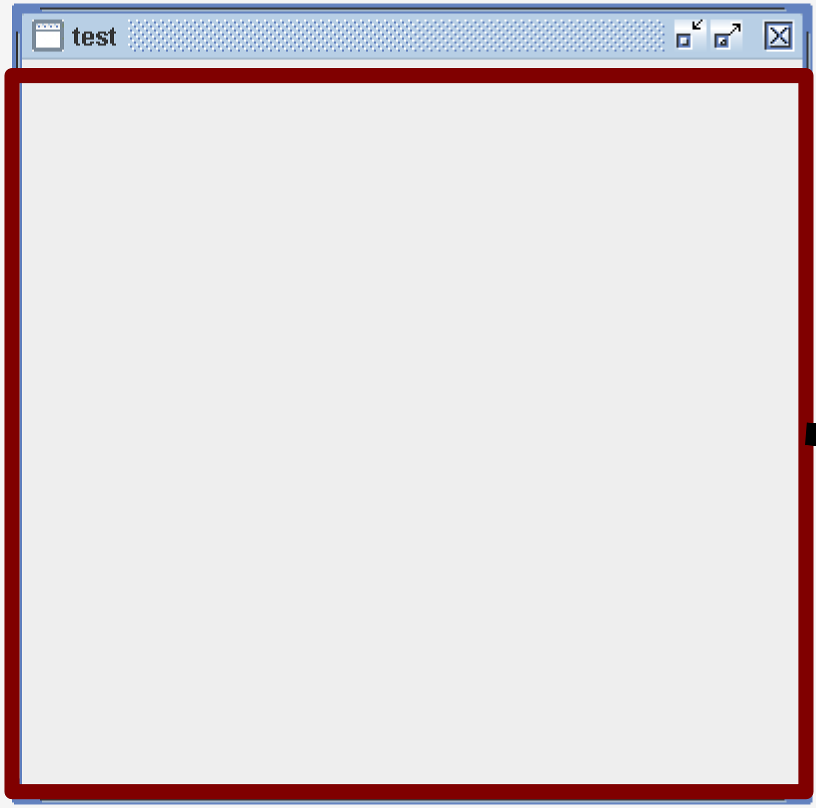
javax.swing

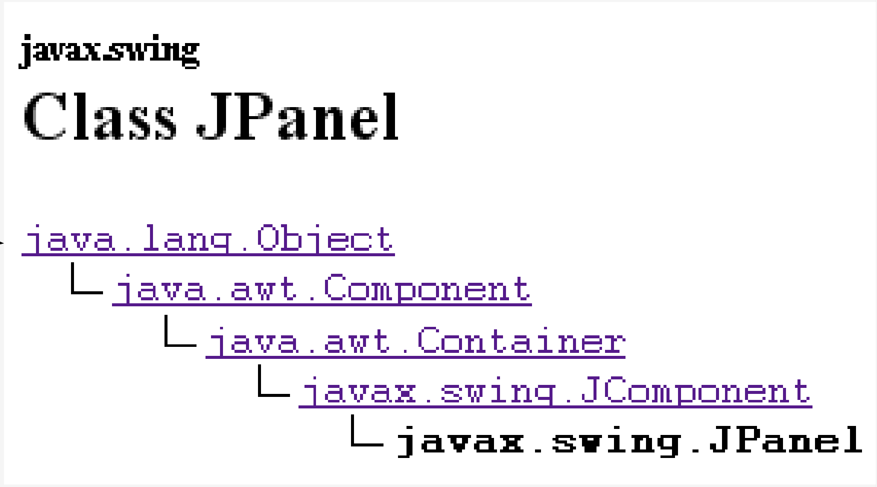
Class JFrame



Method Summary

Container	getContentPane() Returns the contentPane object for this frame.





javax.swing

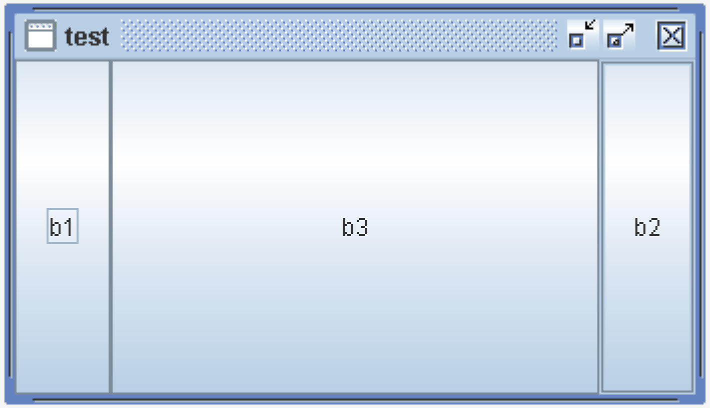
Class JPanel

```
java.lang.Object
  └ java.awt.Component
      └ java.awt.Container
          └ javax.swing.JComponent
              └ javax.swing.JPanel
```

Petite parenthèse

- Le JPanel de base d'une JFrame est, par défaut, géré par un **BorderLayout**

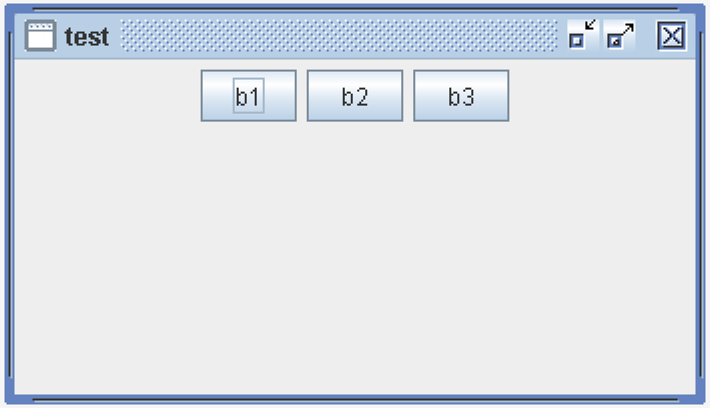
```java
public void init(){
     setSize(350,200);
     setLocation(300,400);
     add(BorderLayout.WEST, new JButton("b1"));
     add(BorderLayout.EAST, new JButton("b2"));
     add( new JButton("b3"));
}
```



Petite parenthèse

- Mais on peut très bien le changer :

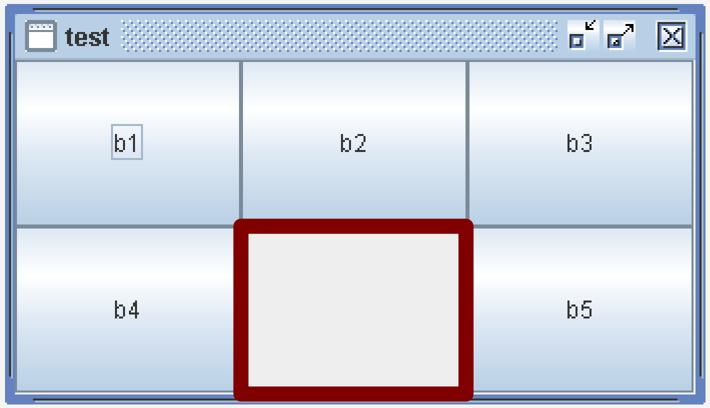
```
public void init(){
     setSize(350,200);
     setLocation(300,400);
     getContentPane().setLayout(new FlowLayout());
     add(new JButton("b1"));
     getContentPane().add(new JButton("b2"));
     add( new JButton("b3"));
}
```



GridLayout

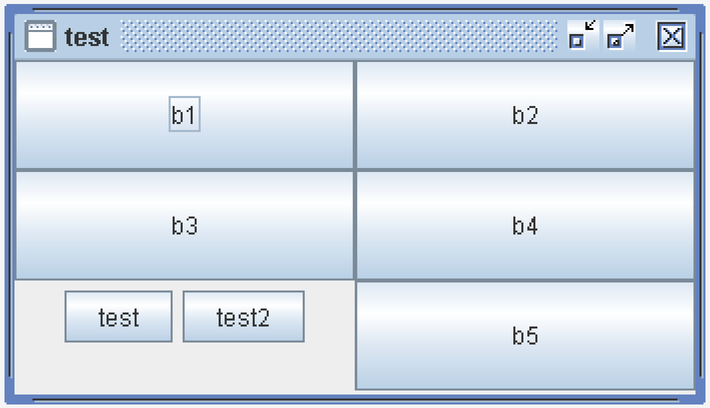
- Définit une **grille** : les composants sont placés en remplissant successivement chacune des cases de la grille dans l'ordre.

```
public void init(){
    setSize(350,200);
    setLocation([illegible]);
    JPanel p = new JPanel(new GridLayout(2,3));
    getContentPane().add(p);
    p.add(new JButton("b1"));
    p.add(new JButton("b2"));
    p.add( new JButton("b3"));
    p.add( new JButton("b4"));
    p.add( new JPanel());
    p.add( new JButton("b5"));
}
```

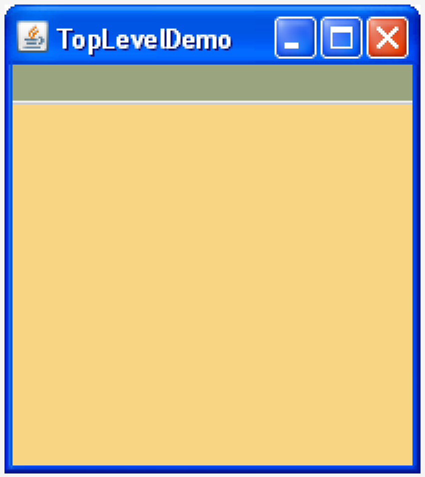


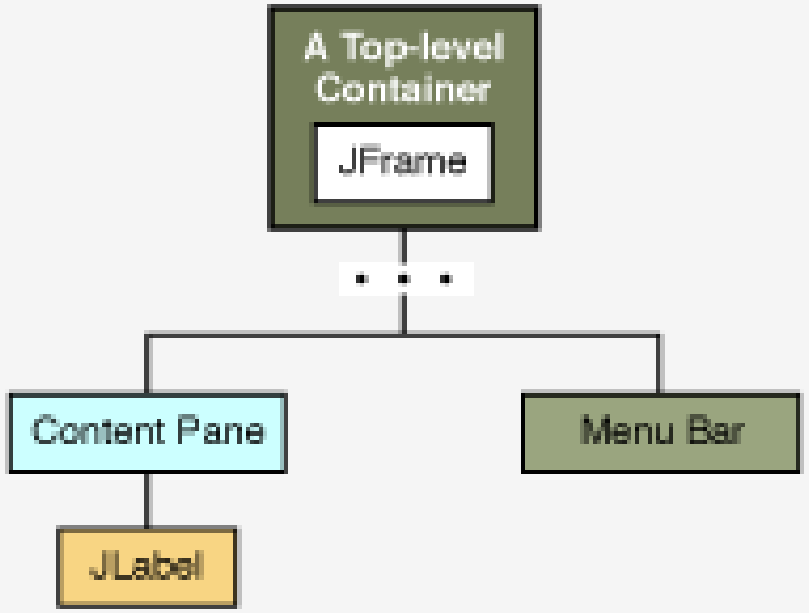
Tout ça est récursif !

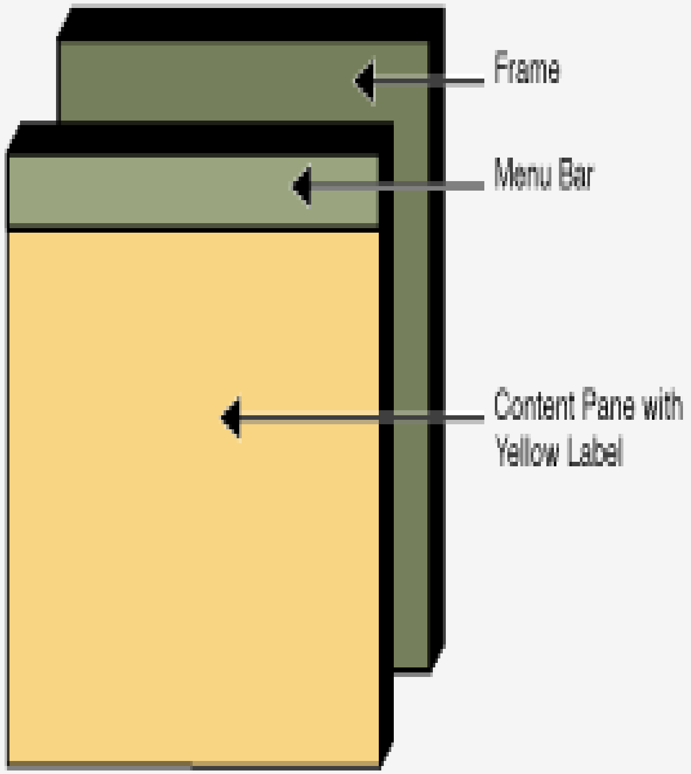
```
public void init(){
    setSize(350,200);
    setLocation(300,400);
    JPanel p = new JPanel(new GridLayout(3,2));
    getContentPane().add(p);
    p.add(new JButton("b1"));
    p.add(new JButton("b2"));
    p.add(new JButton("b3"));
    p.add(new JButton("b4"));
    JPanel p2 = new JPanel();
    p.add(p2);
    p.add(new JButton("b5"));
    p2.add(new JButton("test"));
    p2.add(new JButton("test2"));
}
```



Rappel


TopLevelDemo

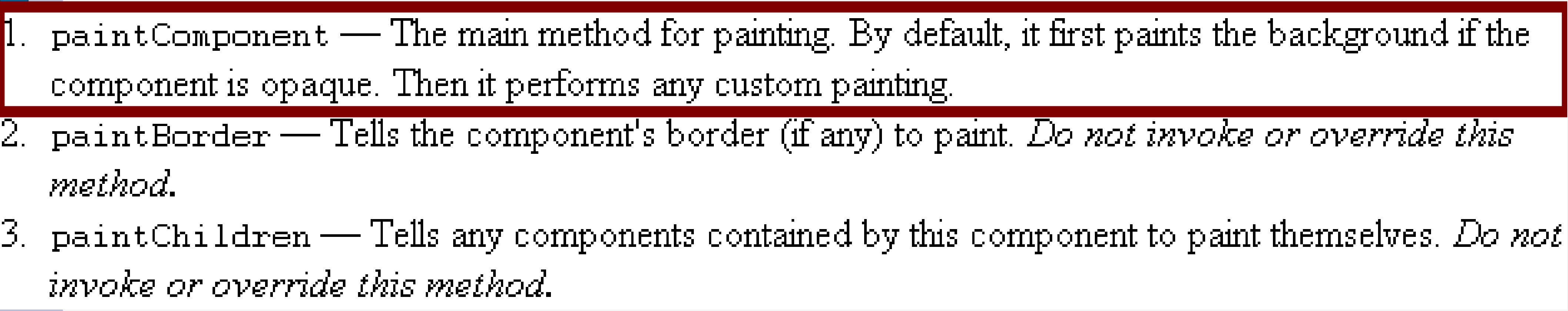

A Top-level Container
JFrame
. . .
Content Pane
Menu Bar
JLabel


Frame
Menu Bar
Content Pane with Yellow Label

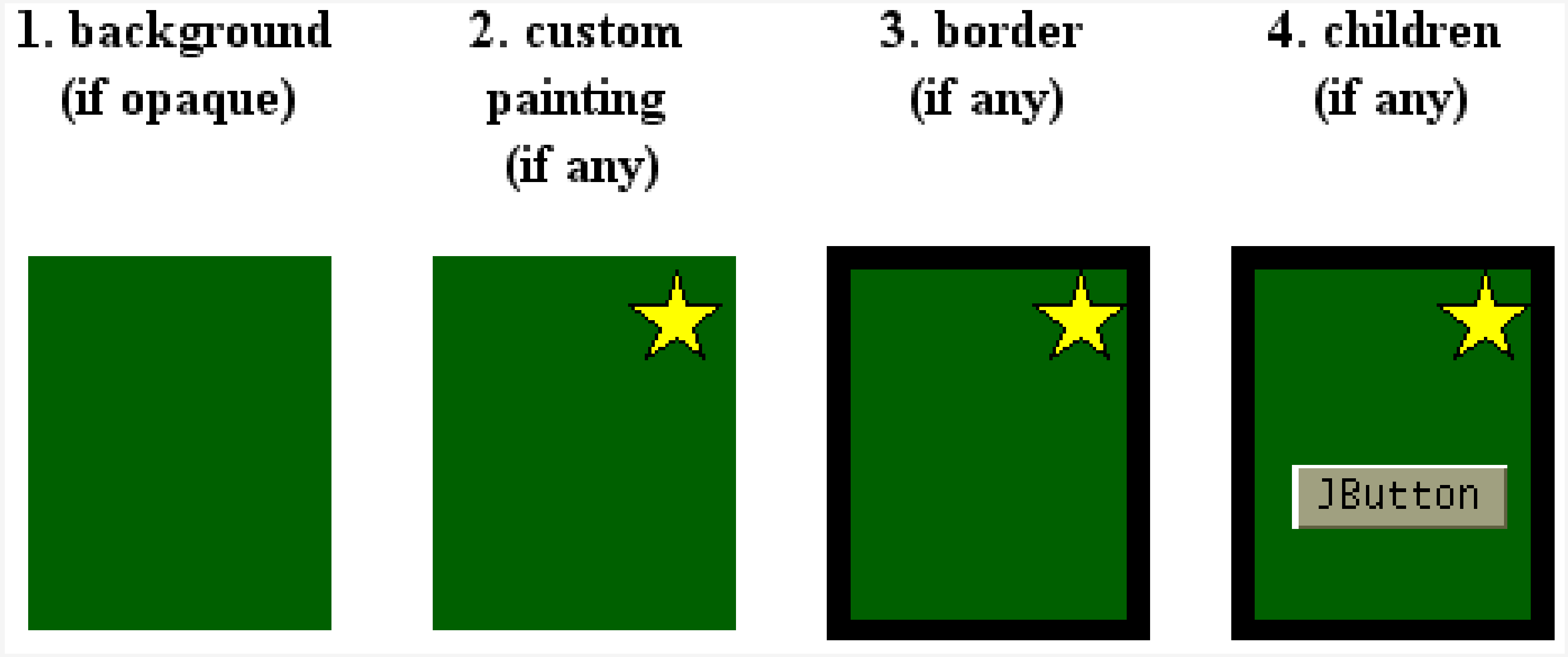
Dessins personnalisés

Comment les composants sont dessinés

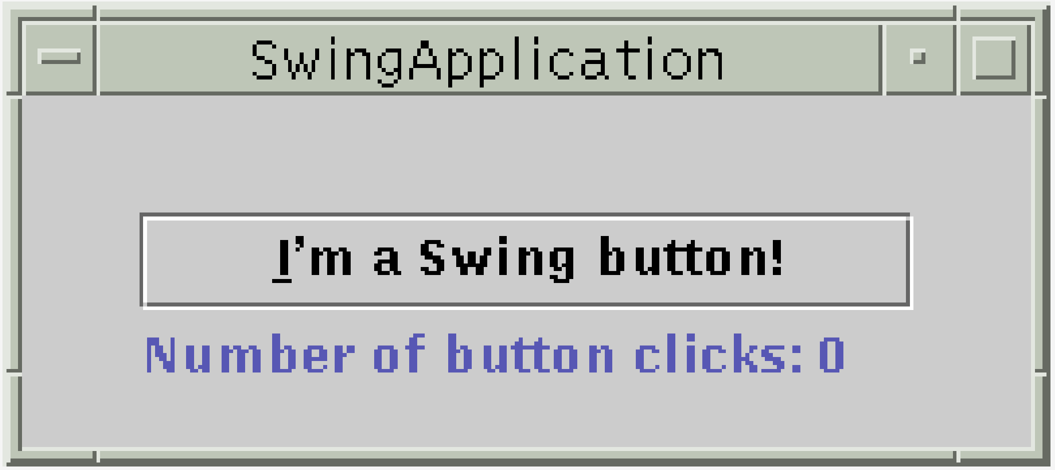
- Dans la classe Jcomponent :

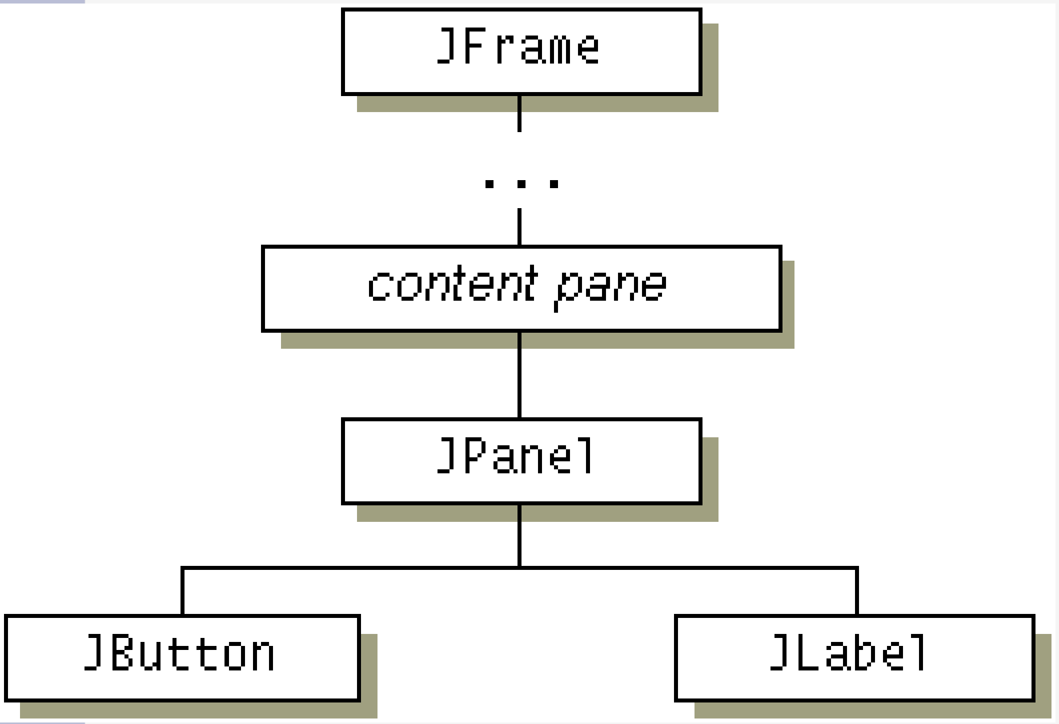


1. `paintComponent` — The main method for painting. By default, it first paints the background if the component is opaque. Then it performs any custom painting.
2. `paintBorder` — Tells the component's border (if any) to paint. *Do not invoke or override this method.*
3. `paintChildren` — Tells any components contained by this component to paint themselves. *Do not invoke or override this method.*



Comment les composants sont dessinés





1. La frame se peint

2. Le contentPane : fond (background : un rectangle gris) et demande ensuite au JPanel de se dessiner

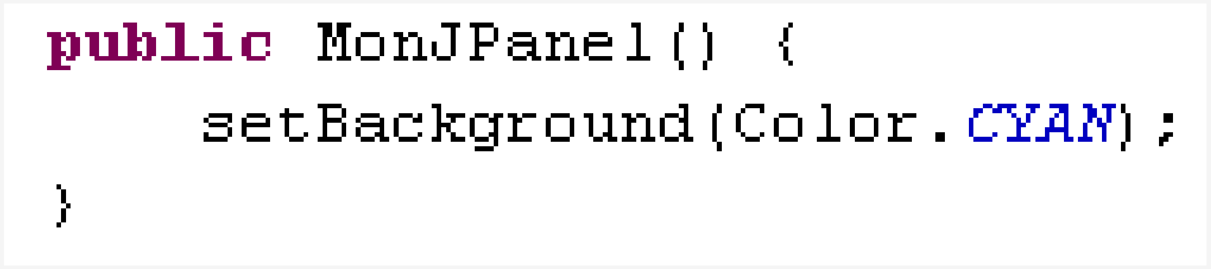
3. JPanel : fond (si opaque), ses bords (vide par défaut) et demande à ses fils de se dessiner

4. JButton : fond, bord, texte

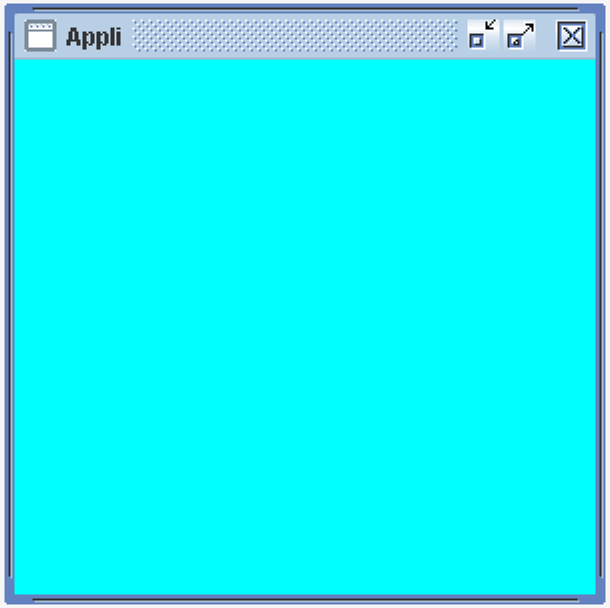
5. Jlabel : affiche le texte

Dessiner ses propres composants

- Un JPanel par exemple :



```
public MonJPanel() {
    setBackground(Color.CYAN);
}
```

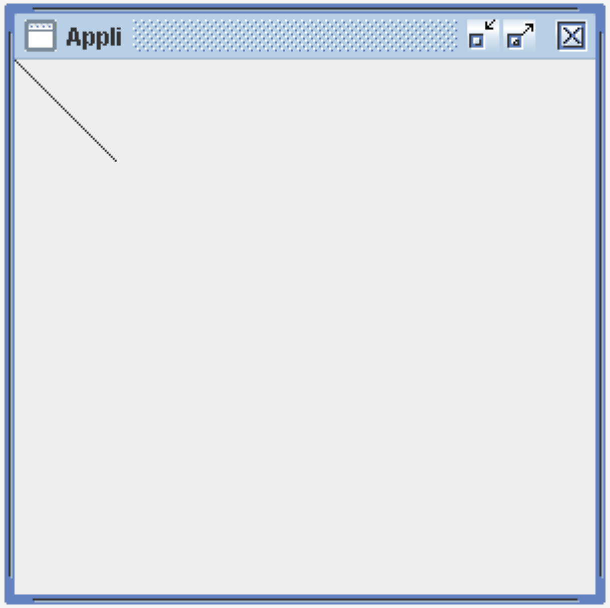


MonJPanel

- Pour définir son propre dessin, il faut surcharger la méthode paintComponent:

```
public MonJPanel() {
     setBackground(Color.CYAN);
}
```

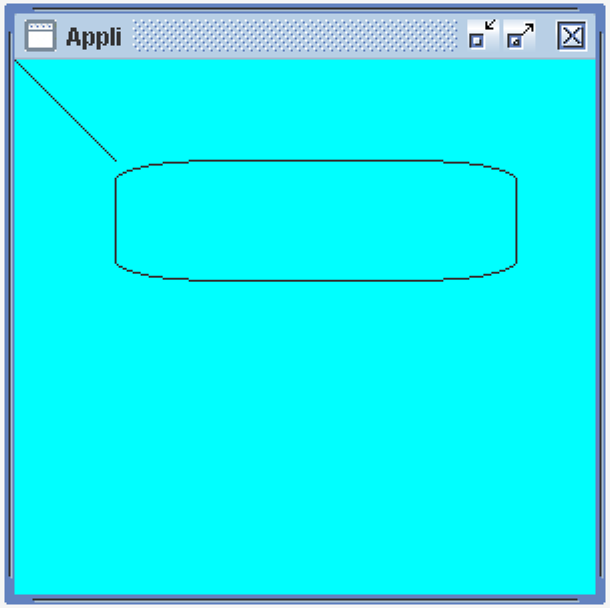
```
protected void paintComponent(Graphics g){
     g.drawLine(0,0,50,50);
}
```



MonJPanel

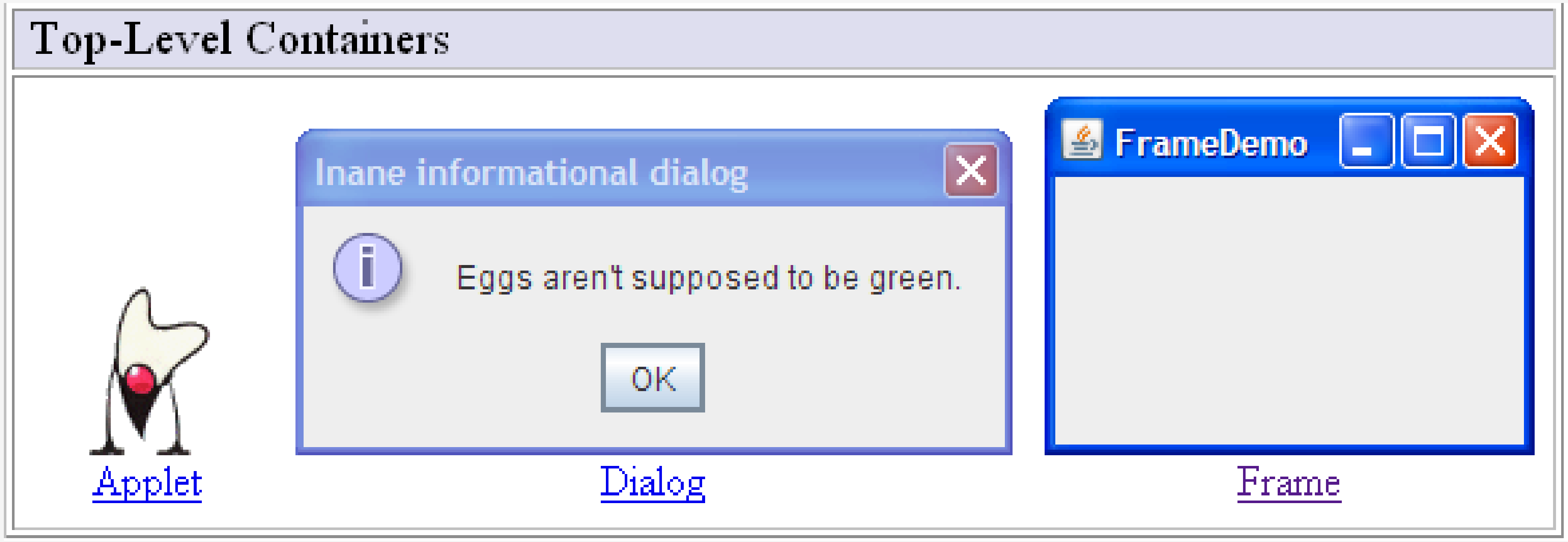
- Ne pas oublier que nous venons de redéfinir une méthode et donc le comportement du composant:

```
protected void paintComponent(Graphics g){
      super.paintComponent(g);
      g.drawLine(0,0,50,50);
      g.drawRoundRect(50, 50, 200, 60, 100, 20);
}
```



Exemples de composants graphiques

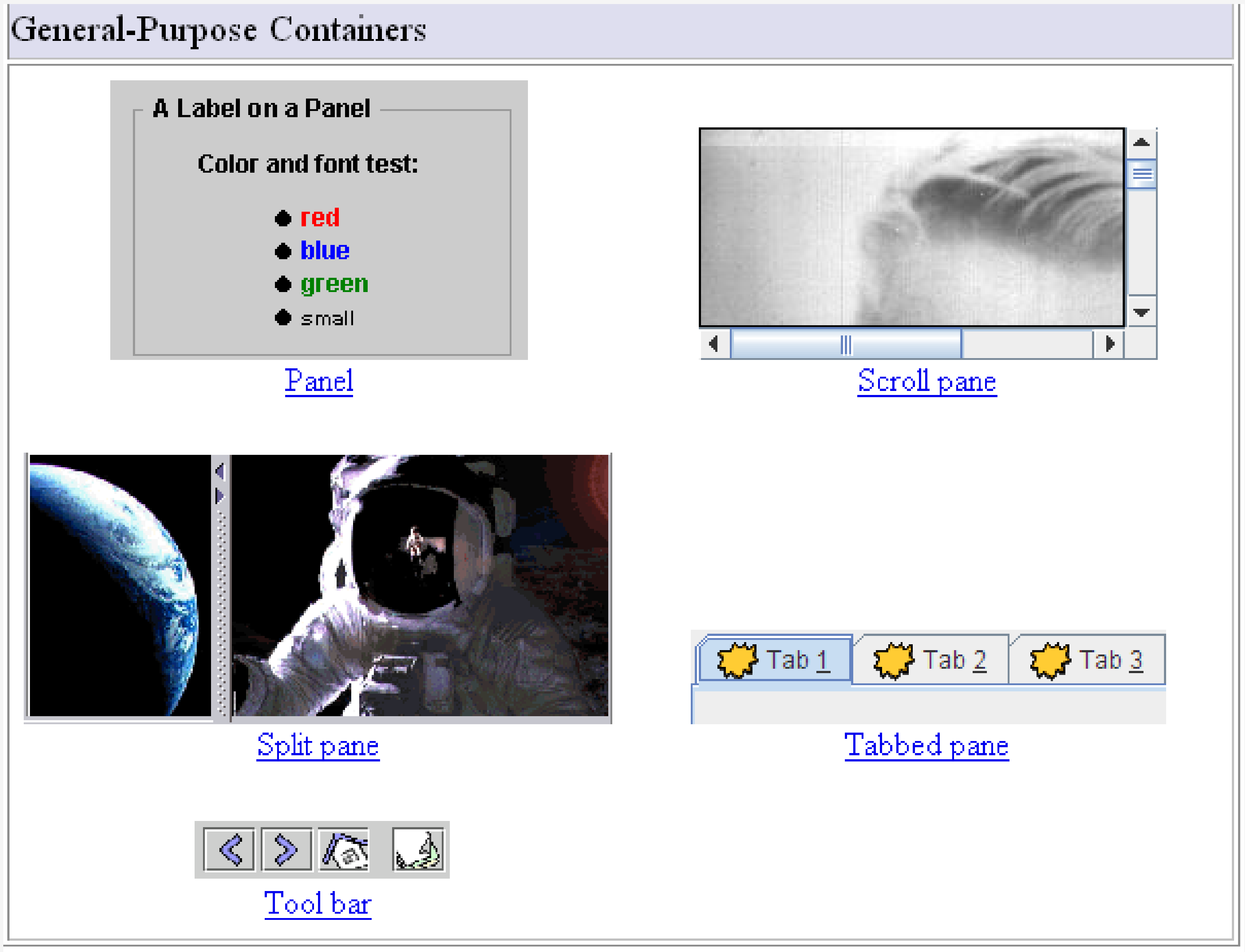
Exemples de container


Top-Level Containers

Inane informational dialog

Eggs aren't supposed to be green.

Exemples de container



General-Purpose Containers

A Label on a Panel

Color and font test:

- red
- blue
- green
- small

Panel

Scroll pane

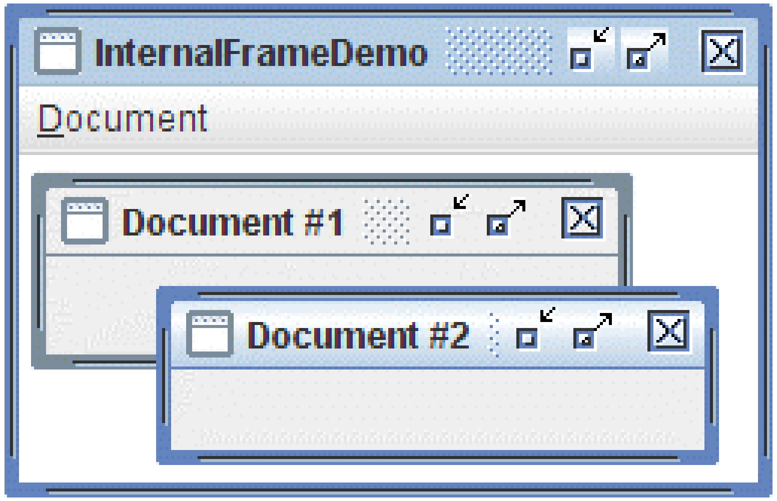
Split pane

Tabbed pane

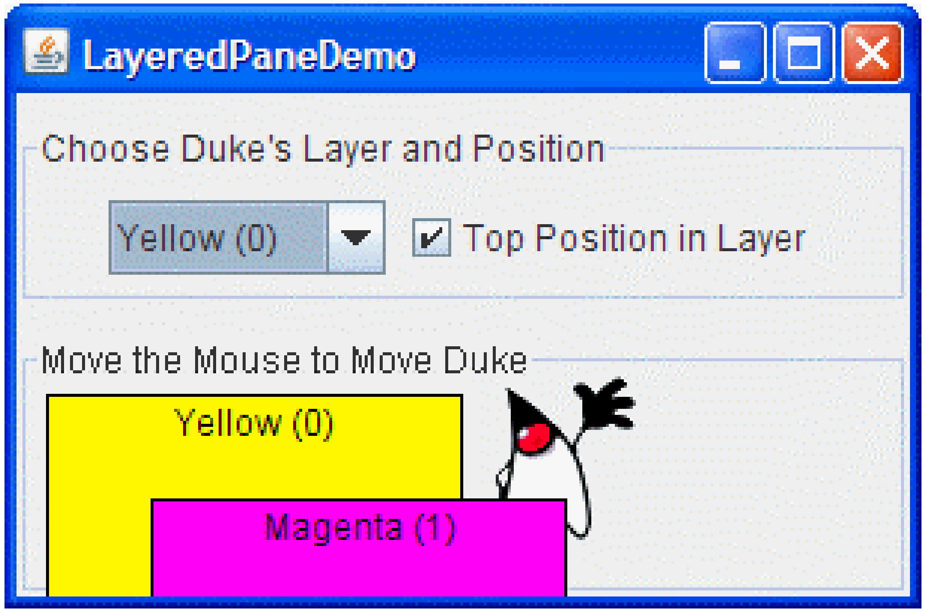
Tool bar

Exemples de container

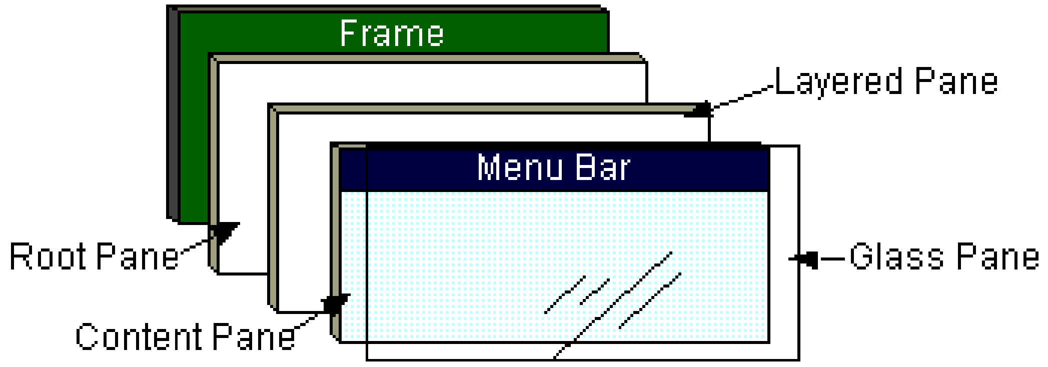
Special-Purpose Containers



Internal frame

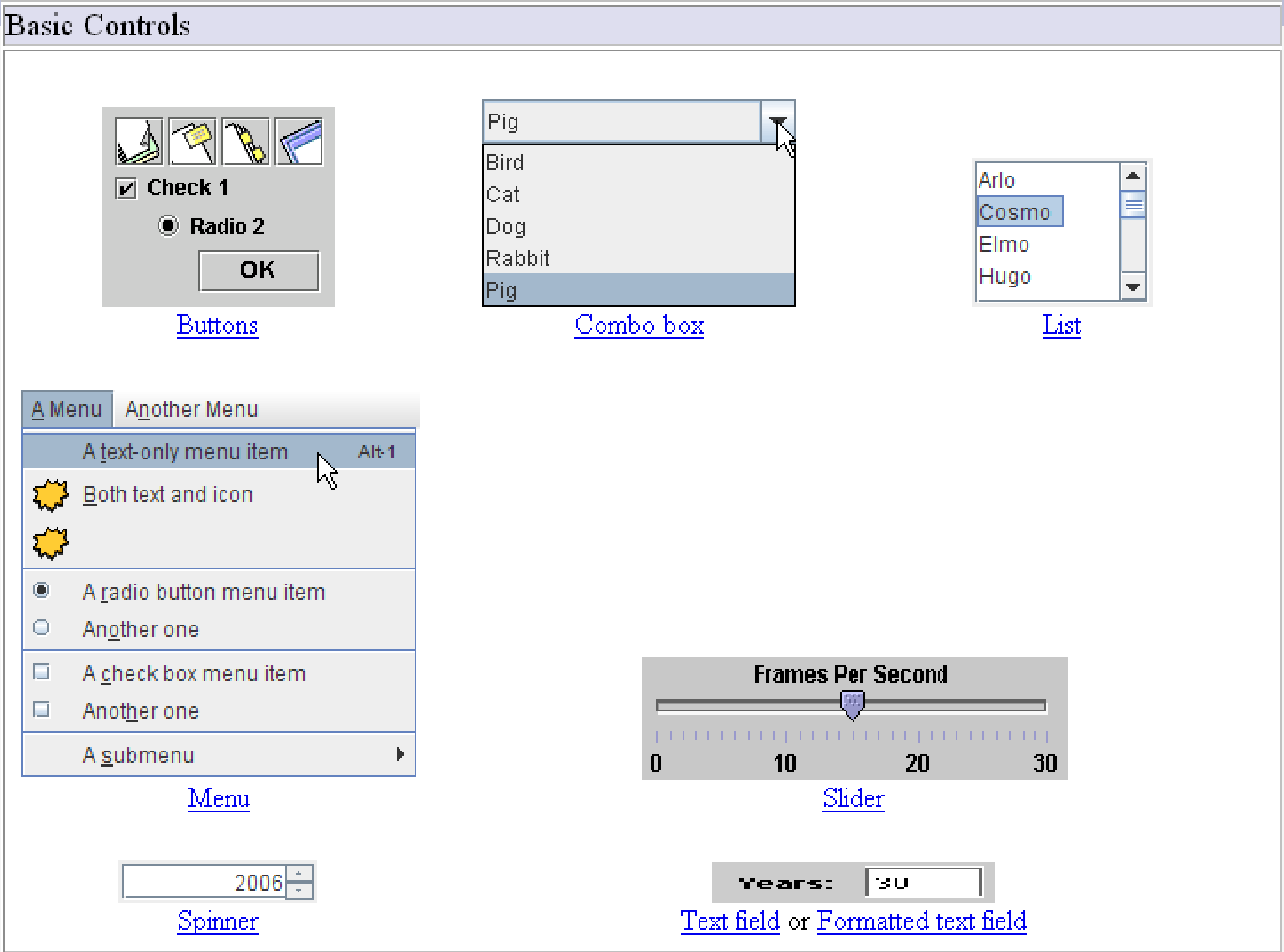


Layered pane

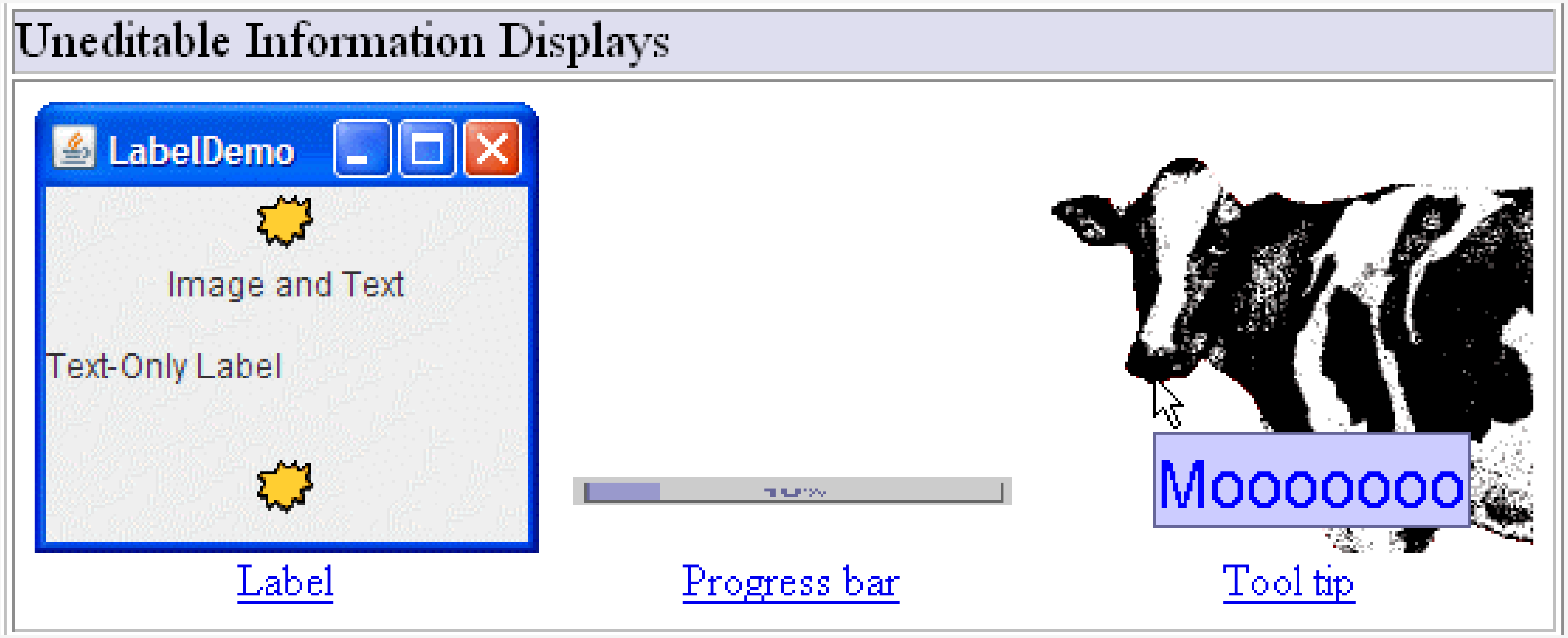


Root pane

Exemples de container



Exemples de container

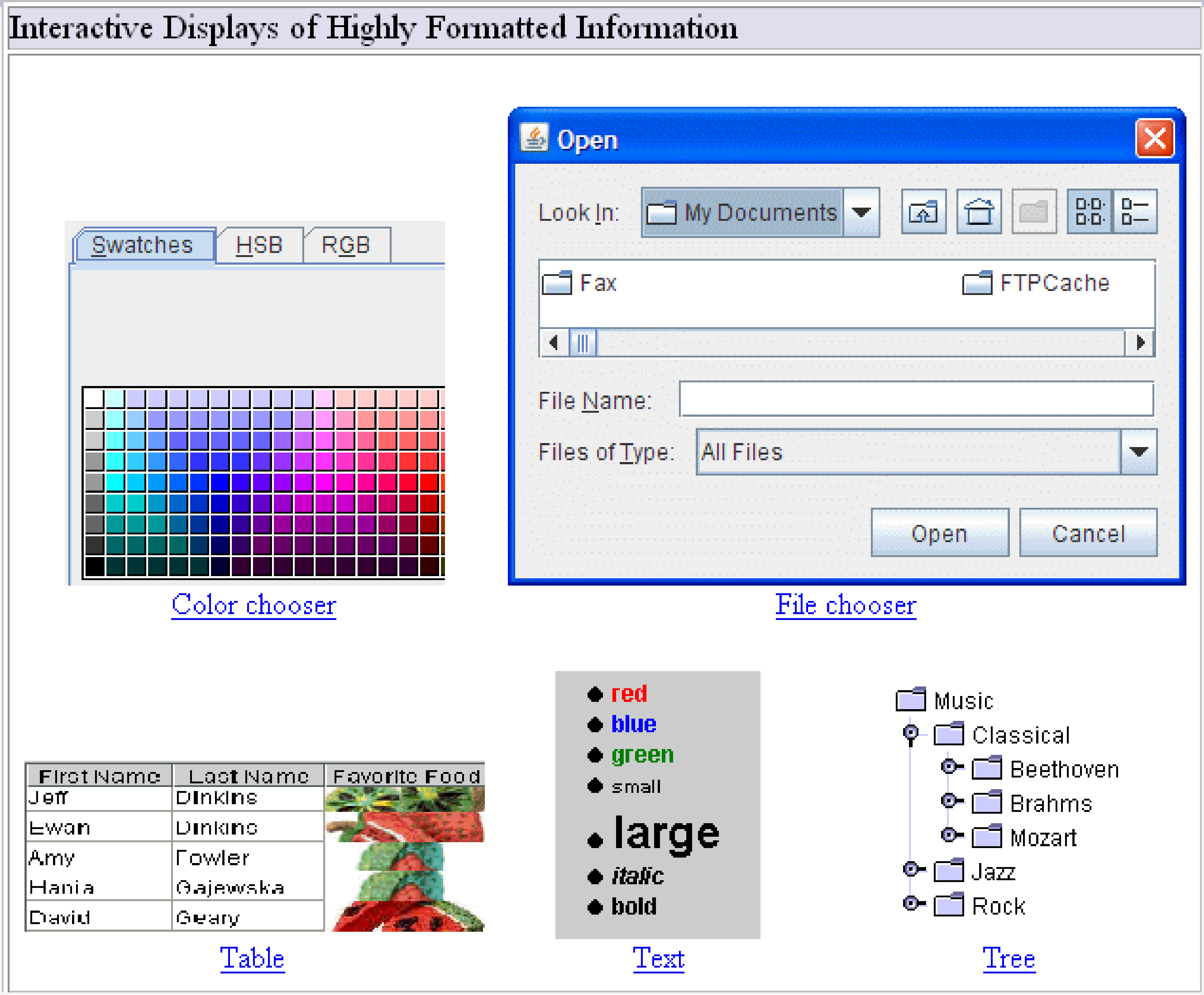

Uneditable Information Displays

Label

Progress bar

Tool tip

Exemples de container



Interactive Displays of Highly Formatted Information

Color chooser

Open

Look In: My Documents

Fax
FTPCache

File Name:

Files of Type: All Files

File chooser

First Name	Last Name	Favorite Food
Jeff	Dinkins	
Ewan	Dinkins	
Amy	Fowler	
Hania	Gajewska	
David	Geary	

Table

- red
- blue
- green
- small
- large
- *italic*
- **bold**

Text

- Music
  - Classical
    - Beethoven
    - Brahms
    - Mozart
  - Jazz
  - Rock

Tree