Interface graphiques

Java Swing

Les interfaces graphiques avec Java

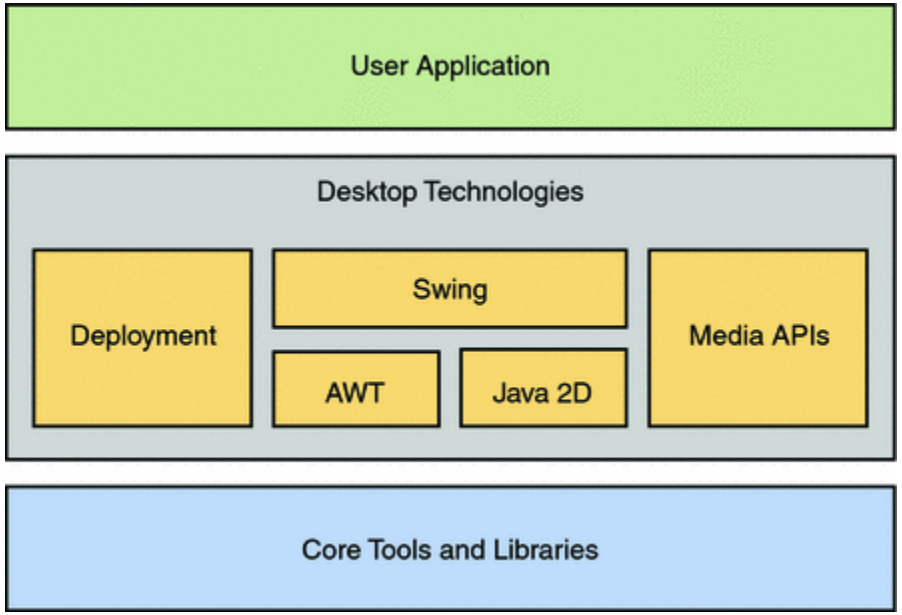
•**Une interface graphique** (IHM) : est un moyen de communication entre l'utilisateur et un système (application). Elle est contient des composants graphiques (fenêtres, boutons, menus, zones de textes , etc).

•**Les IHM en Java** : Java offre des bibliothèques (ensemble de classes) pour construire de telles IHM :

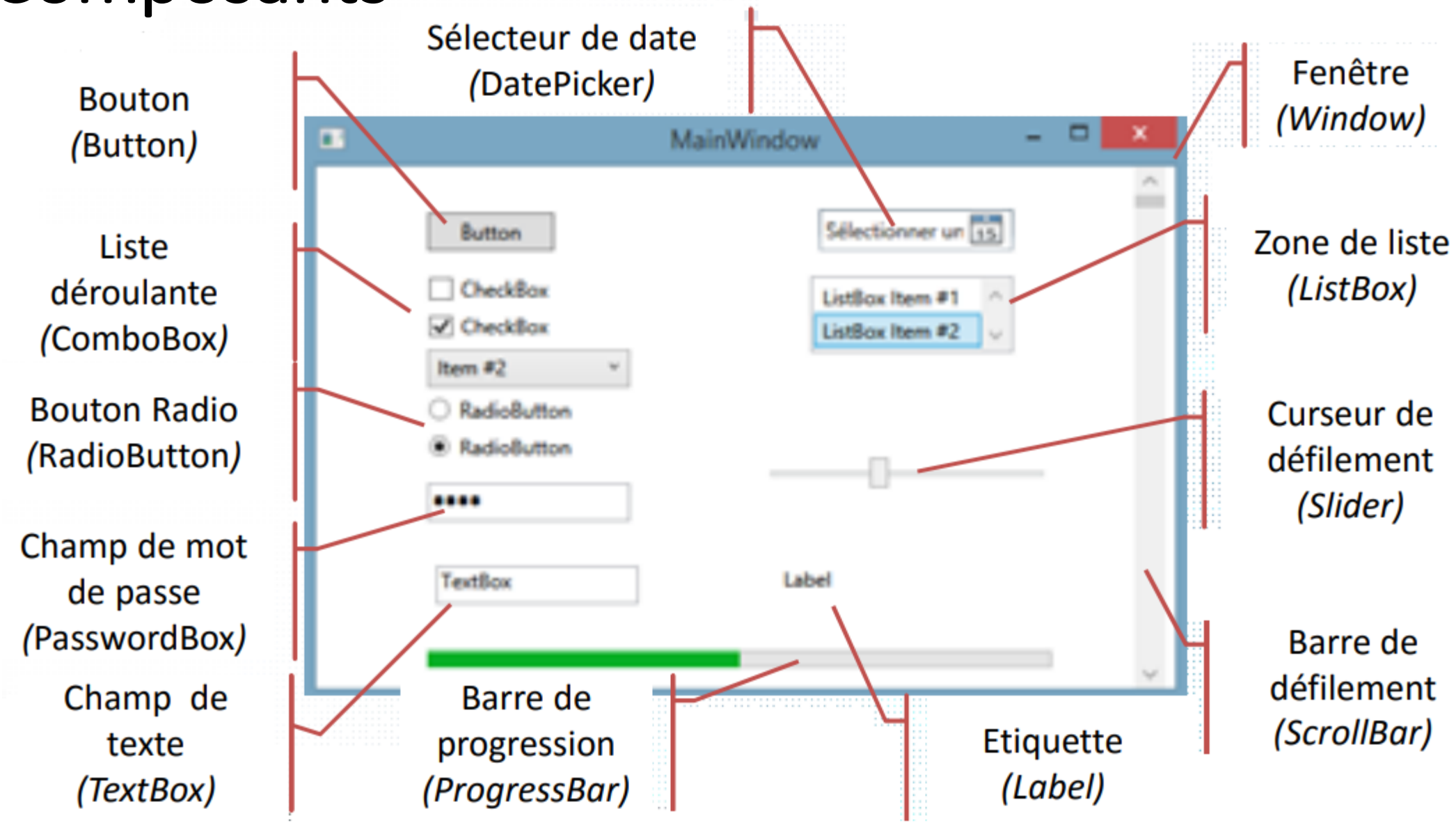
- **AWT** (Abstract Window Toolkit)
- **SWING** (extension de AWT)

•Les environnements de programmation tels que : JBuilder,NetBeans offrent des composants prêt à l'emploi dans la palettedes composants

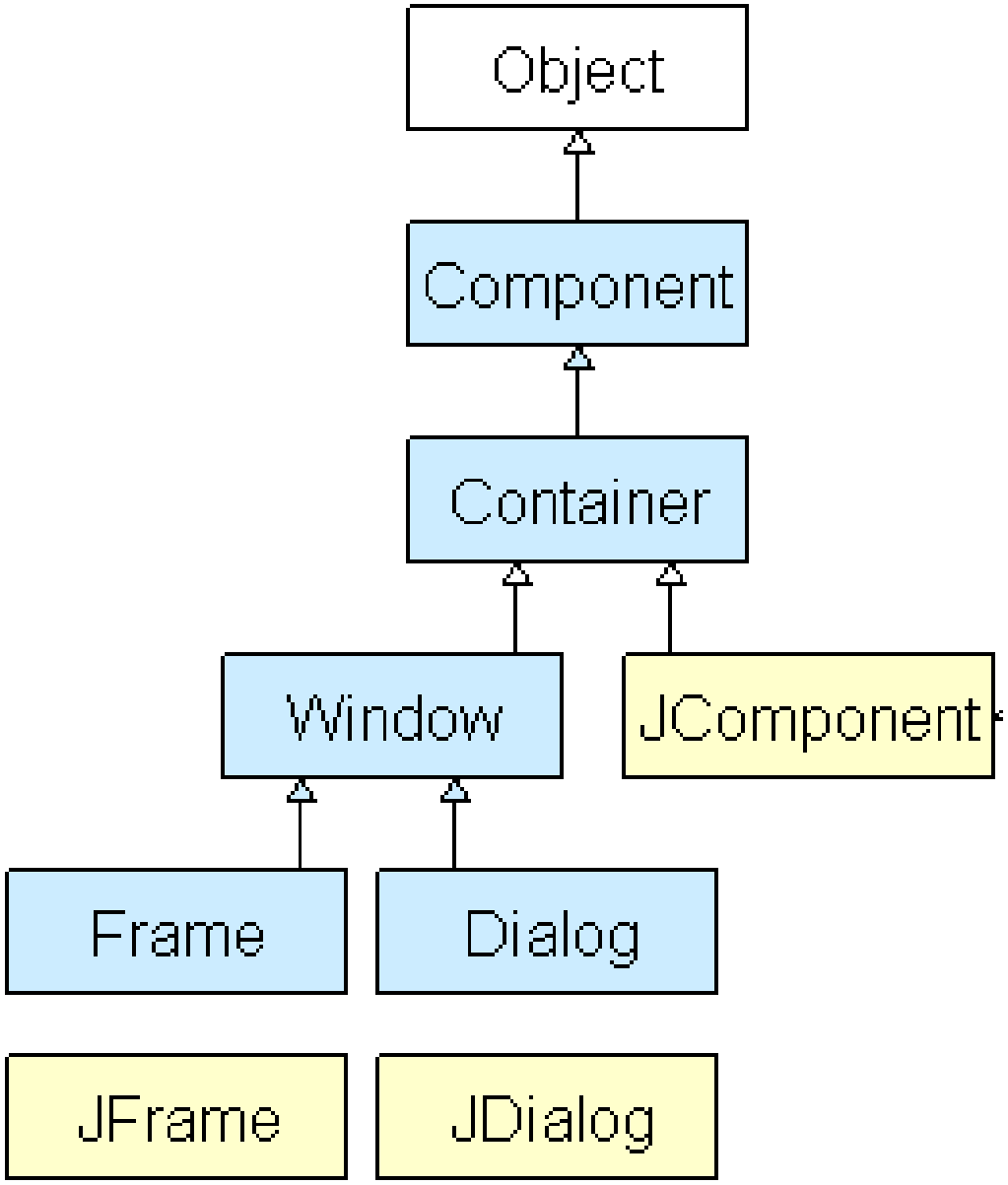
Java SE Desktop Technologies



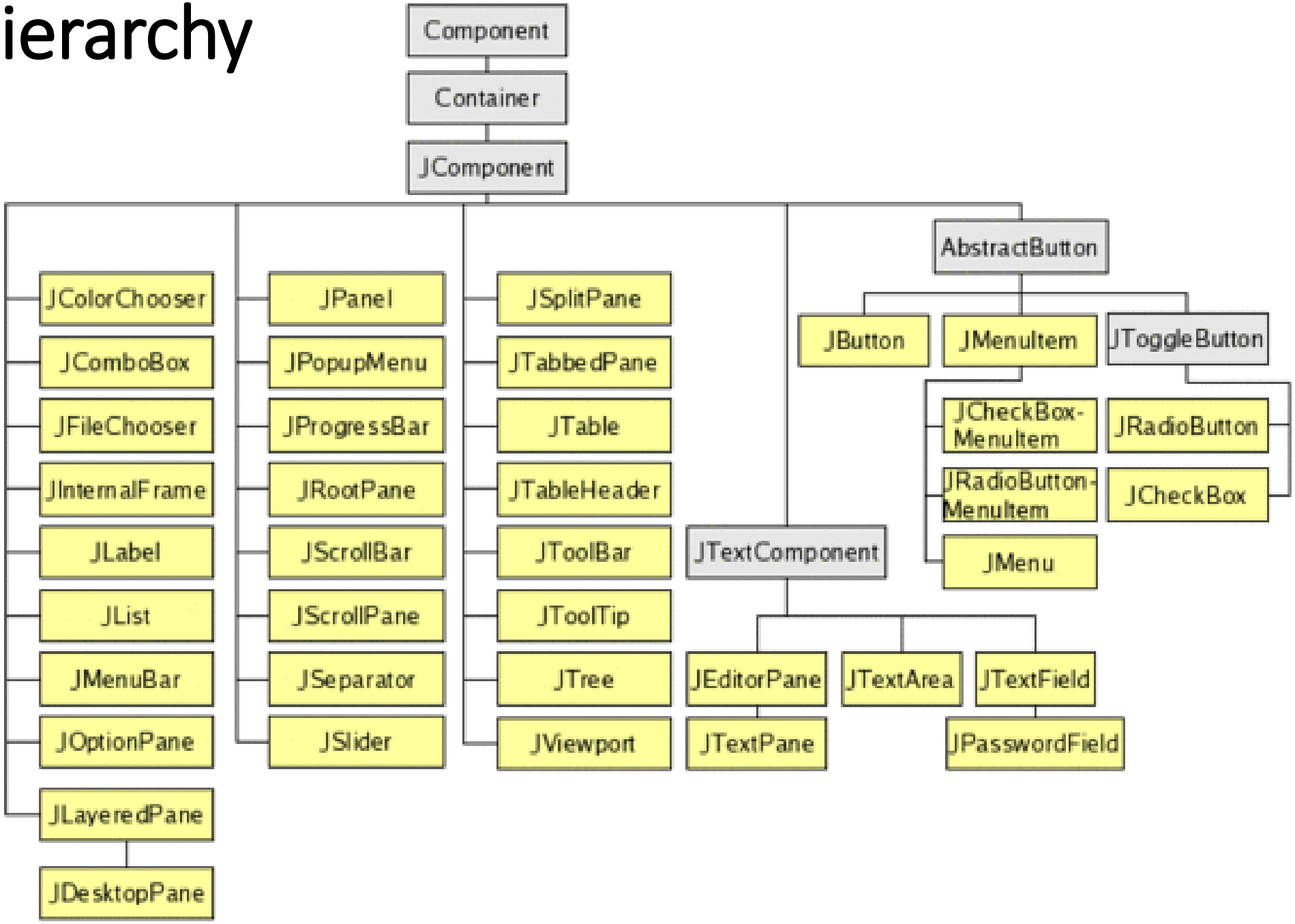
Composants


Sélecteur de date
(DatePicker)
Bouton
(Button)
Fenêtre
(Window)
MainWindow
Button
Sélectionner un
Liste
déroulante
(ComboBox)
CheckBox
CheckBox
ListBox Item #1
ListBox Item #2
Item #2
Zone de liste
(ListBox)
Bouton Radio
(RadioButton)
RadioButton
RadioButton
Curseur de
défilement
(Slider)
Champ de mot
de passe
(PasswordBox)
TextBox
Label
Barre de
défilement
(ScrollBar)
Champ de
texte
(TextBox)
Barre de
progression
(ProgressBar)
Etiquette
(Label)

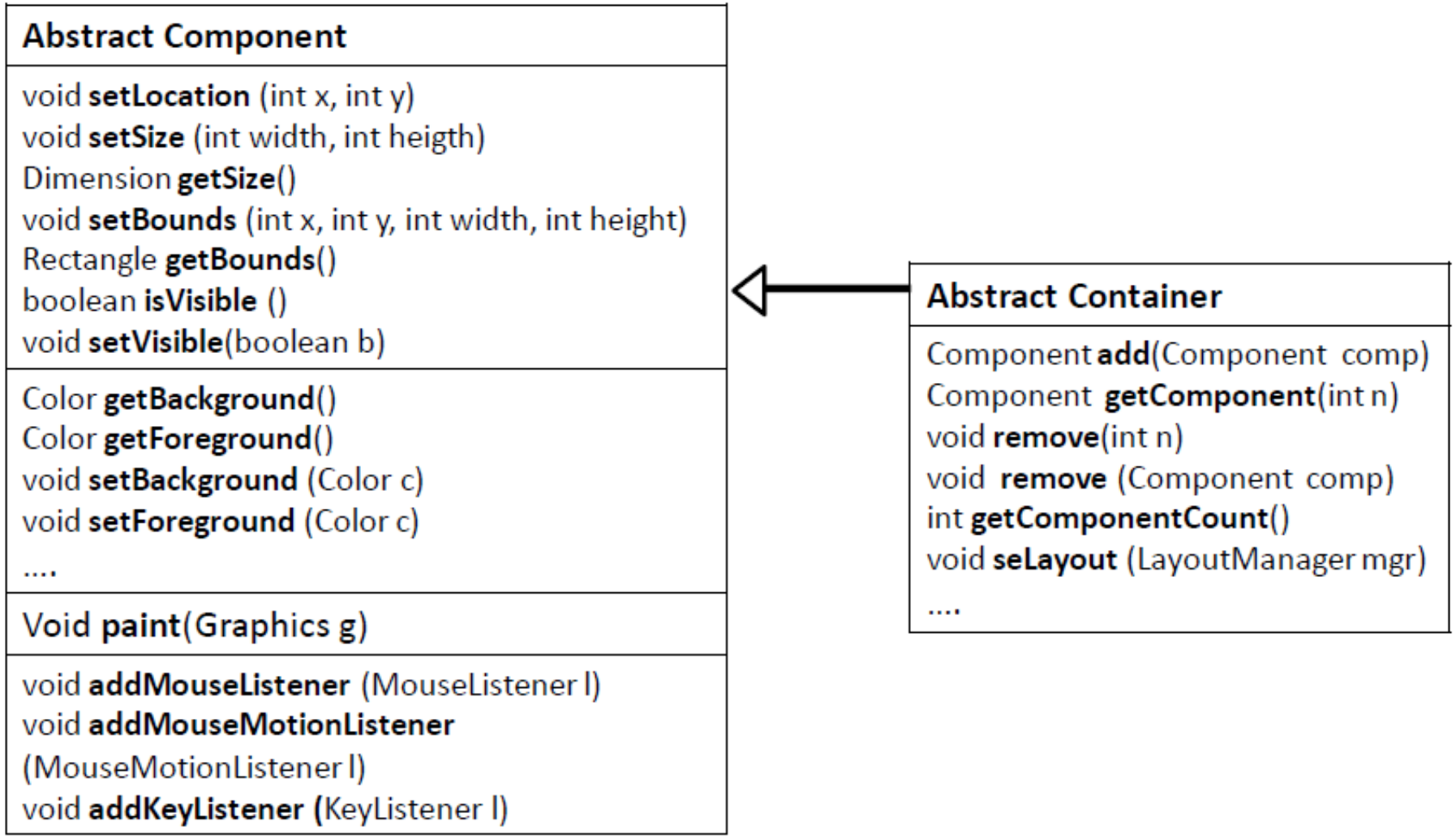
Swing Hierarchy



Swing Hierarchy



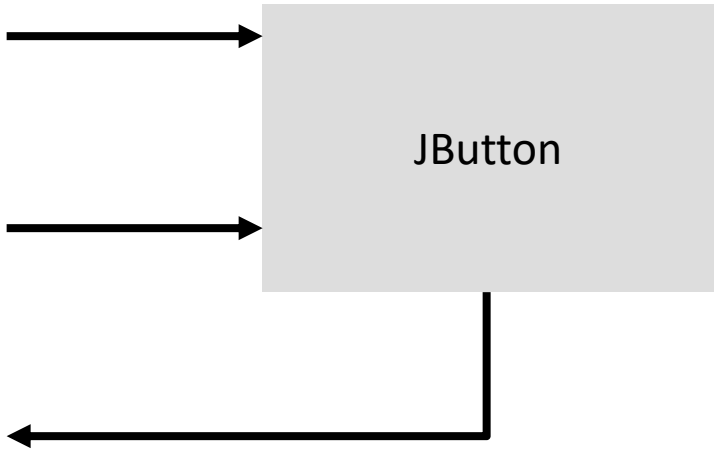
Les classes Component & Container



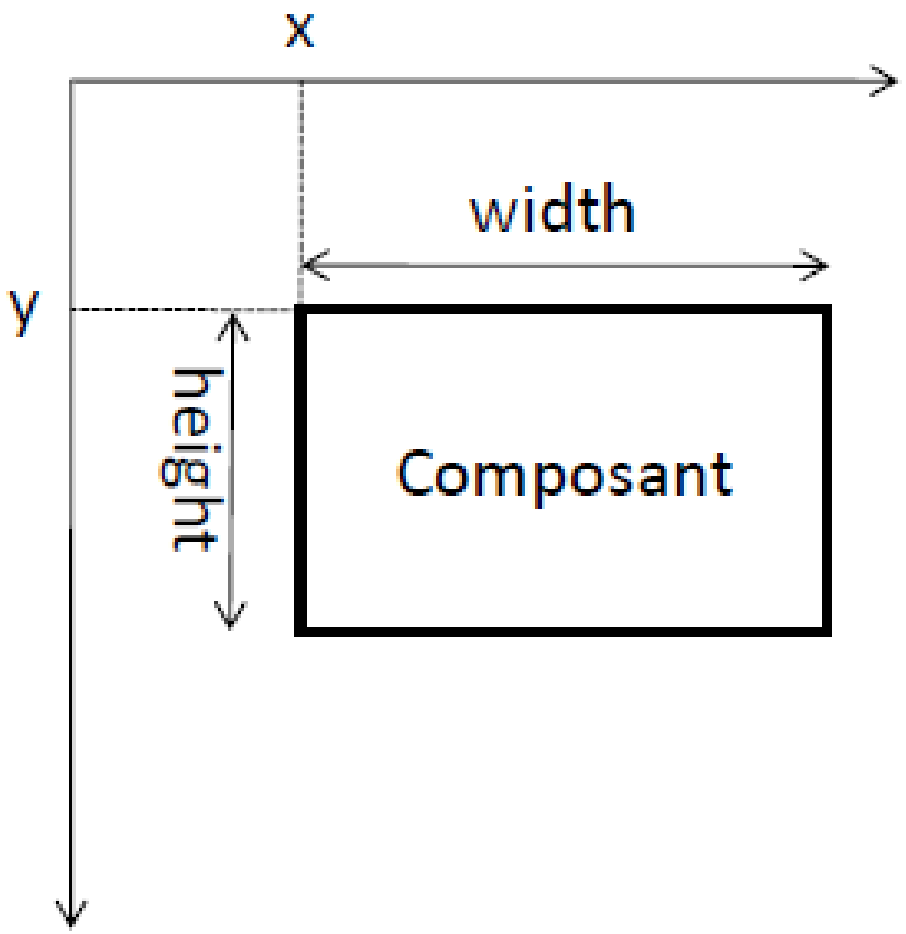
GUI Component API

- Java: GUI component = class

- Properties
    -
- Methods
    -
- Events
    -



Les classes Component & Container

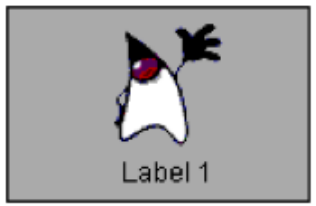


Object

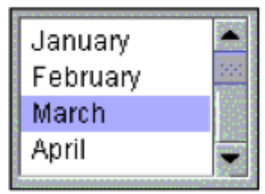
...

JComponent

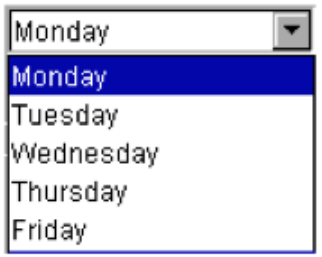
- AbstractButton
- JLabel
- JSeparator
- JList
- JComboBox
- JScrollBar
- JSlider
- JToolTip
- JTextComponent
- *Conteneurs ...*



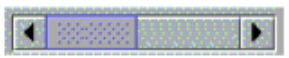
JLabel



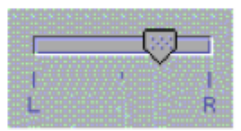
JList



JComboBox

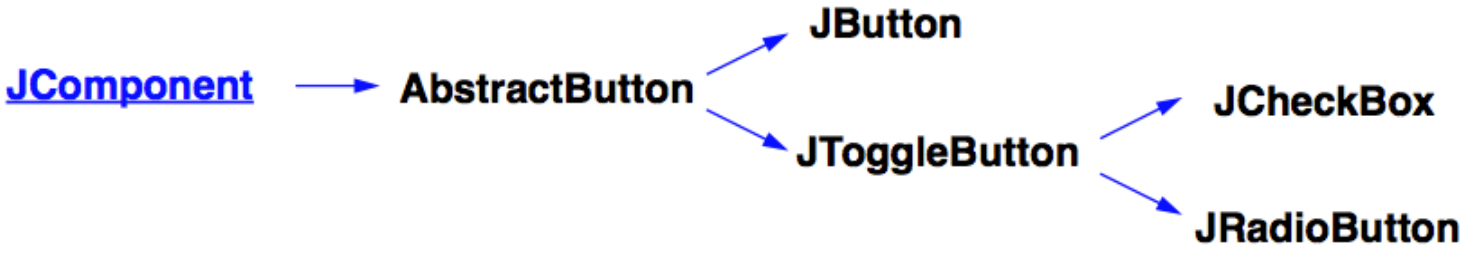


JScrollBar

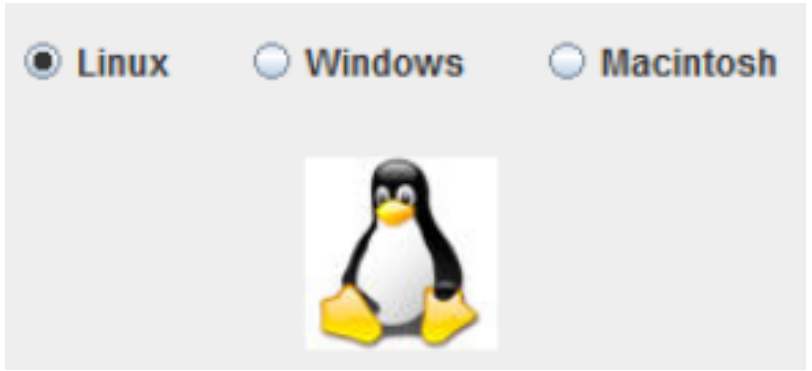


JSlider

Boutons




```
JButton b=new JButton("Click Here");
 b.setBounds(50,100,95,30);
```



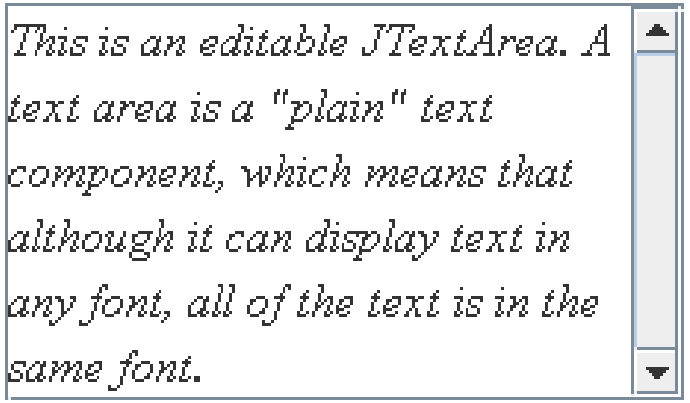
Texte


```
JTextField  t1=new JTextField("Welcome to Javatpoint.");
```

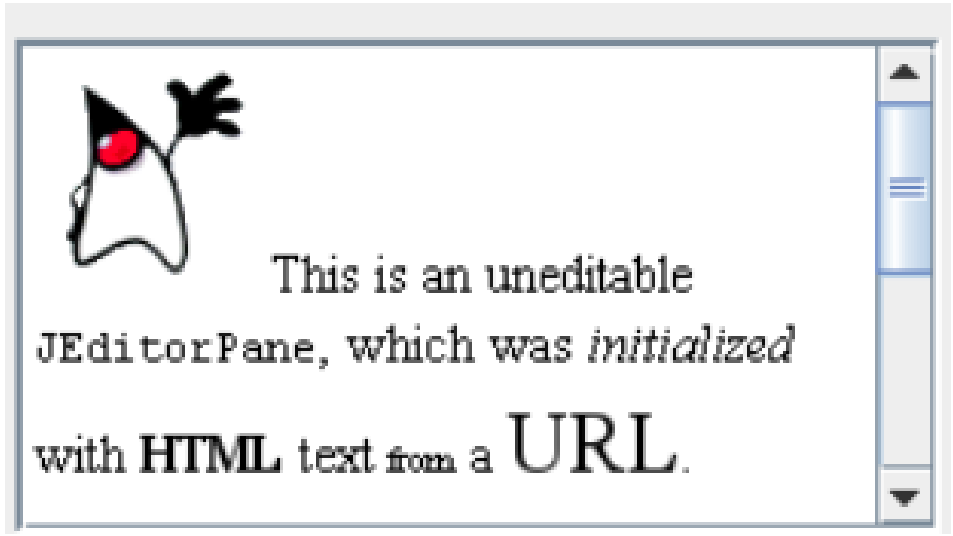
JTextField

JPasswordField

JTextArea



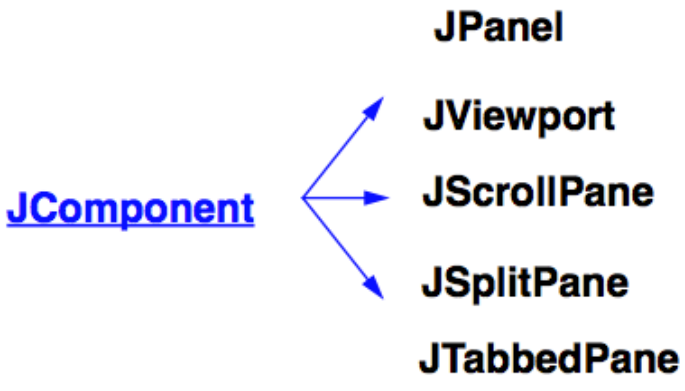
JEditorPane



Label

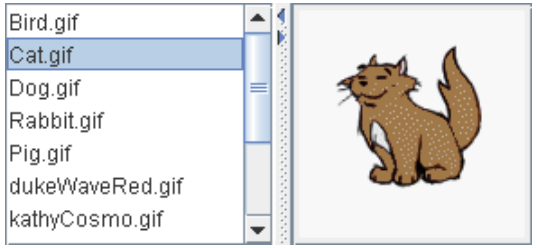
```
JLabel l1 =new JLabel("First Label.");
  l1.setBounds(50,50, 100,30);
```

Conteneurs

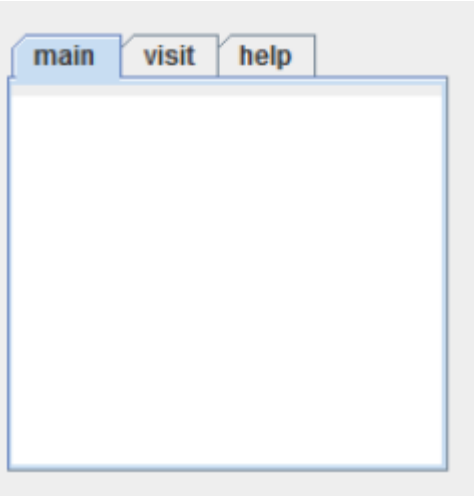


JPanel:

JScrollPane:

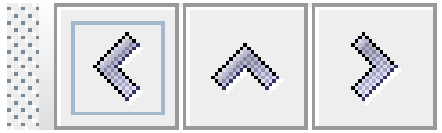


JSplitPane:

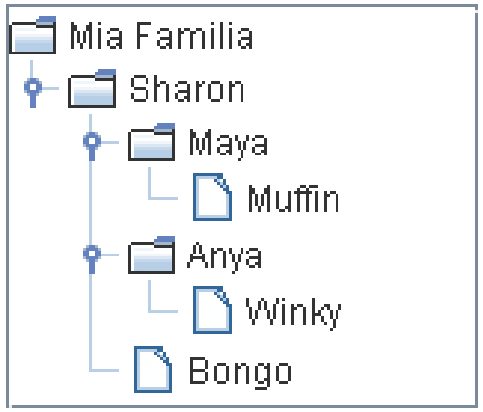


JTabbedPane

Conteneurs



JToolBar:

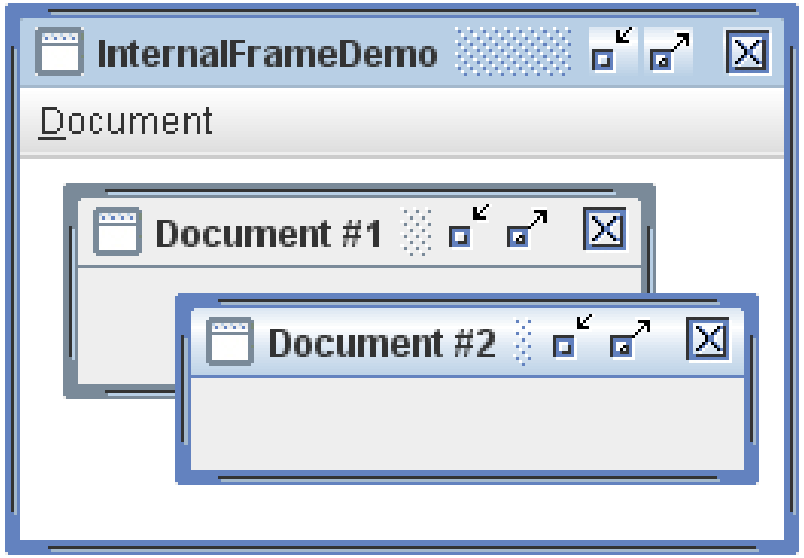


JTree

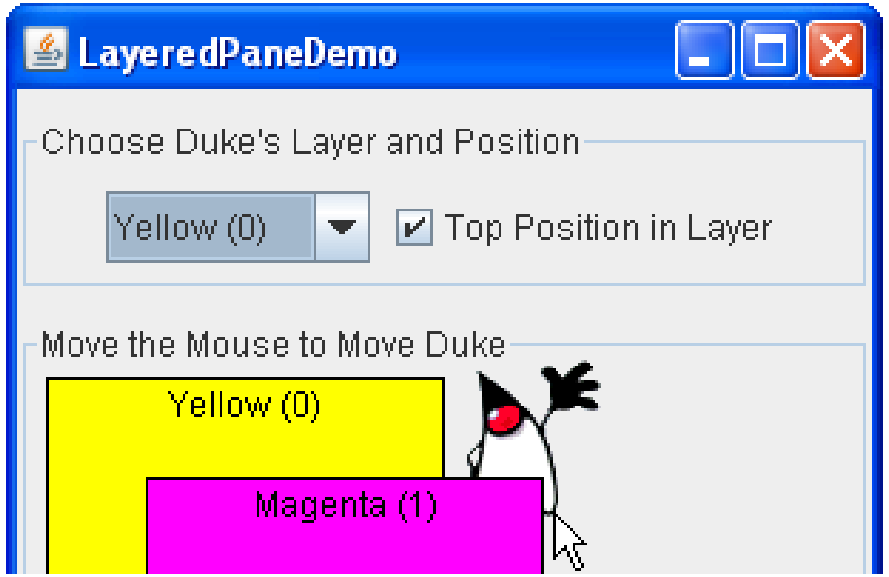
JTable

Host	User	Password	Last Modified
Biocca Games	Freddy	!#asf6Awwzb	Mar 16, 2006
zabble	ichabod	Tazb!34$fZ	Mar 6, 2006
Sun Developer	fraz@hotmail.co...	AasW541!fbZ	Feb 22, 2006
Heirloom Seeds	shams@gmail....	bkz[ADF78!	Jul 29, 2005
Pacific Zoo Shop	seal@hotmail.c...	vbAf124%z	Feb 22, 2006

Conteneurs spécifiques

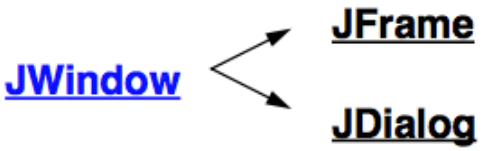


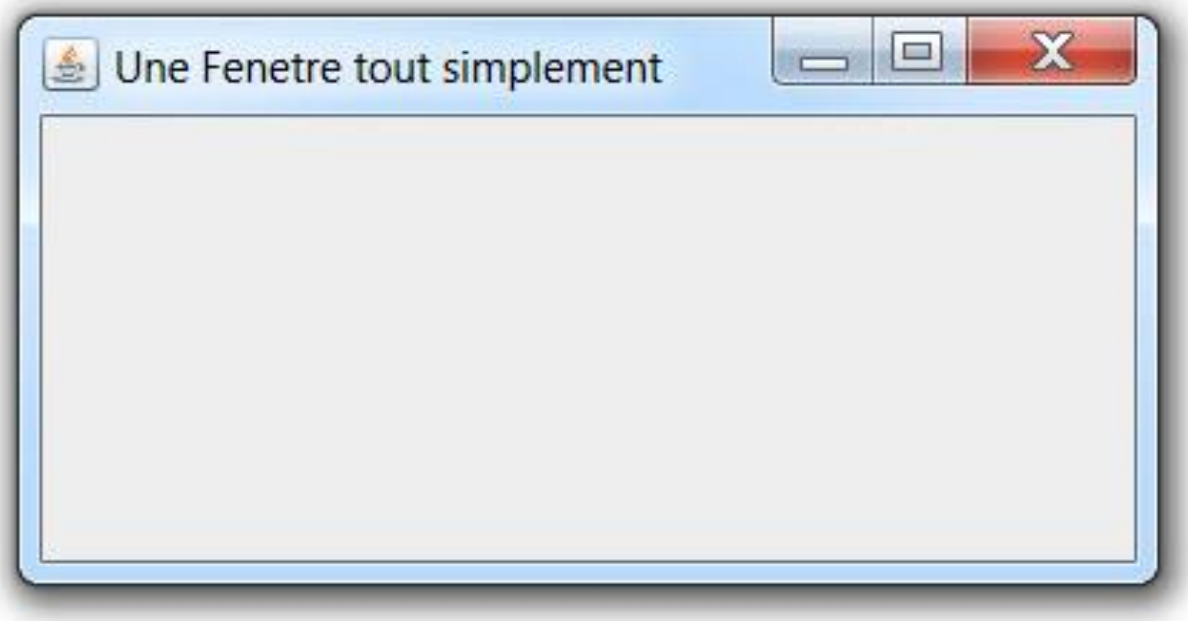
JInternalFrame



JLayeredPane

Fenêtres

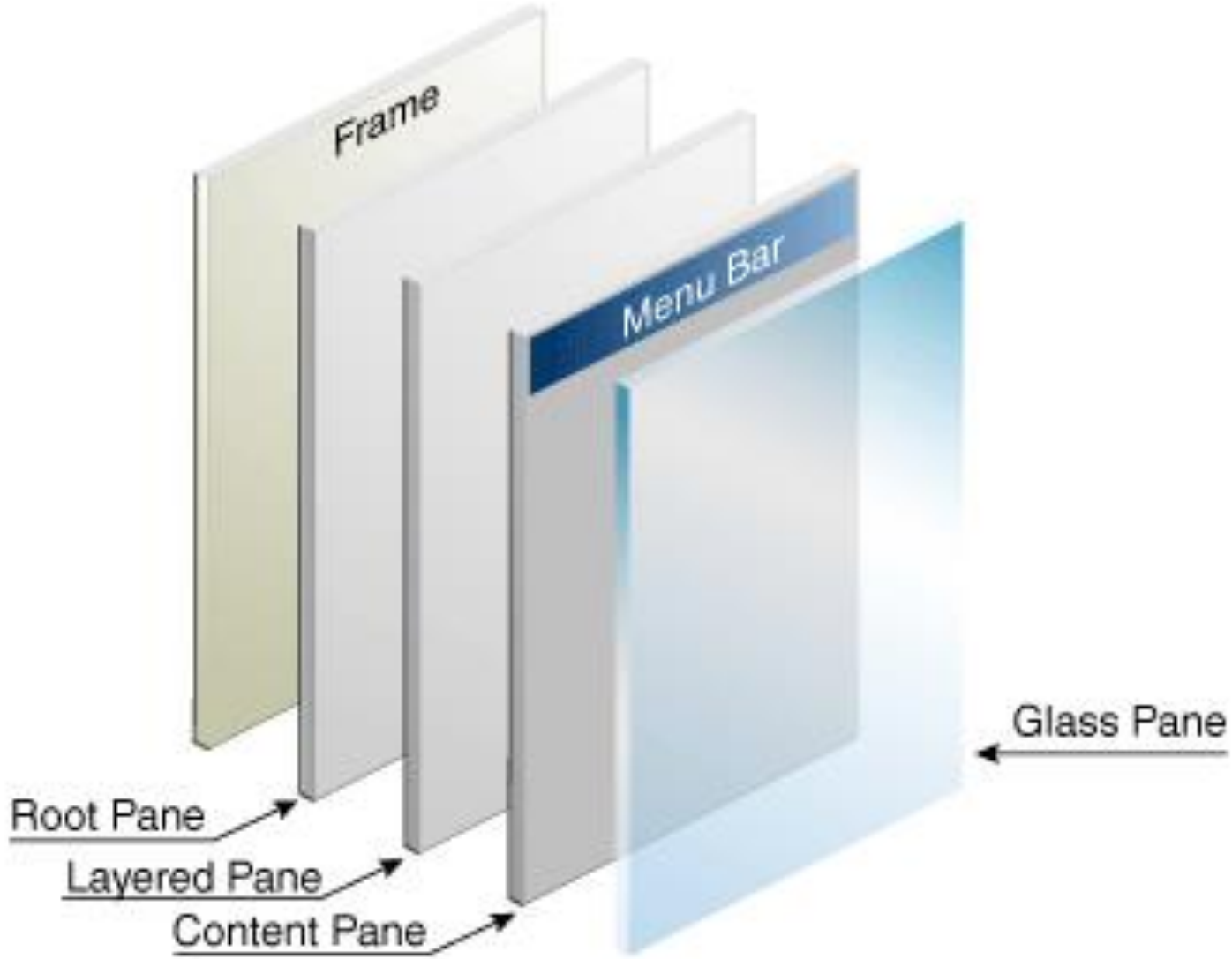




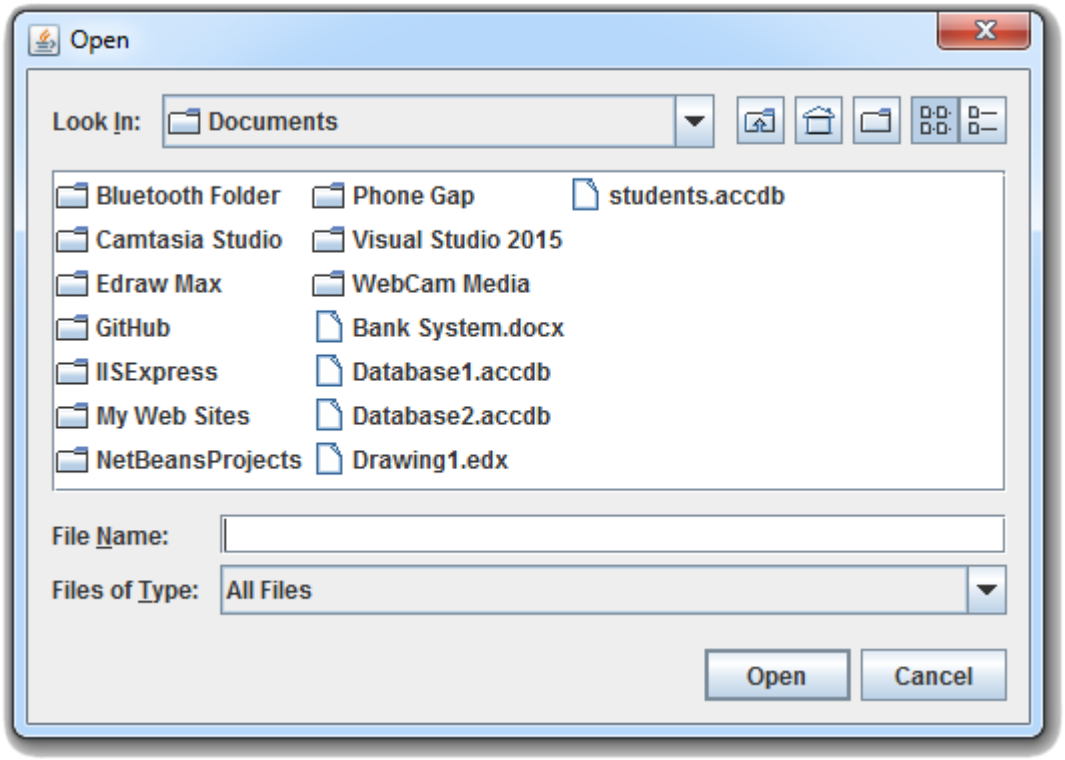
JFrame

JDialog,

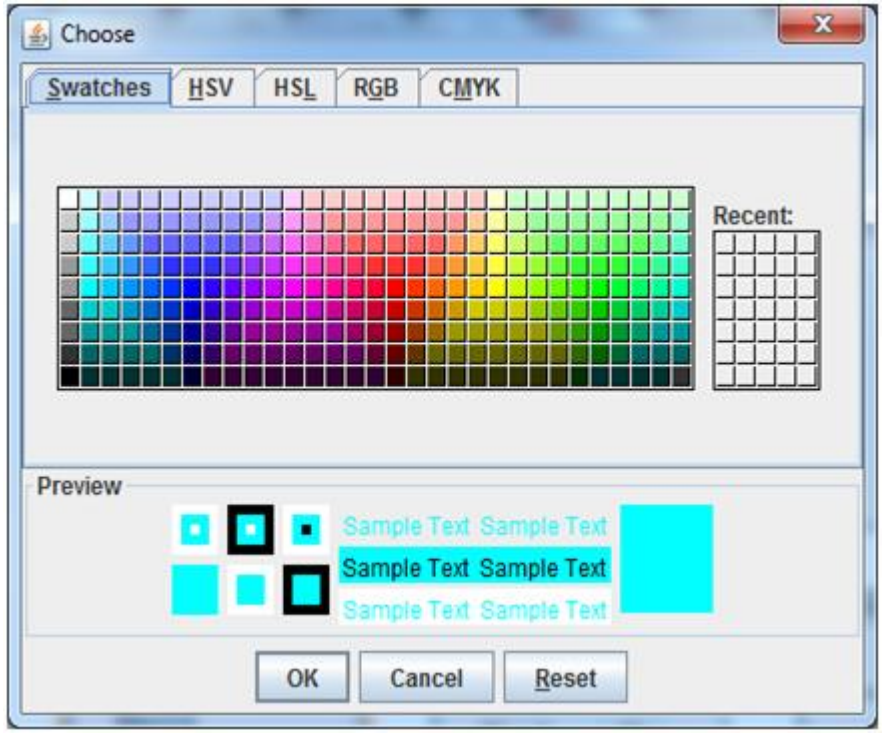
Fenêtres- JRootPane



Boîtes de dialogue prédéfinies



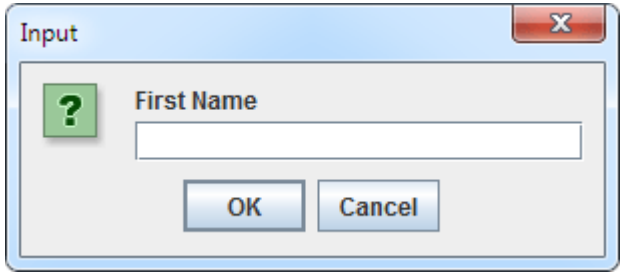
JFileChooser



JColorChooser

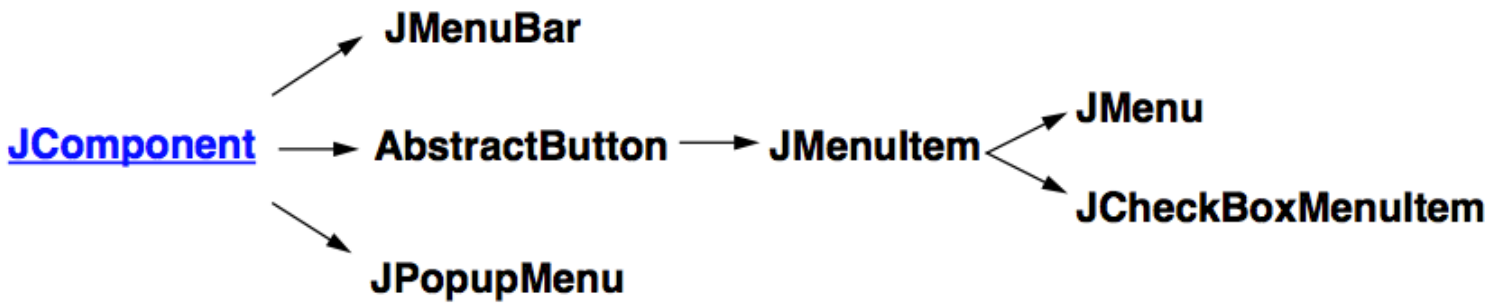
Boîtes de dialogue prédéfinies

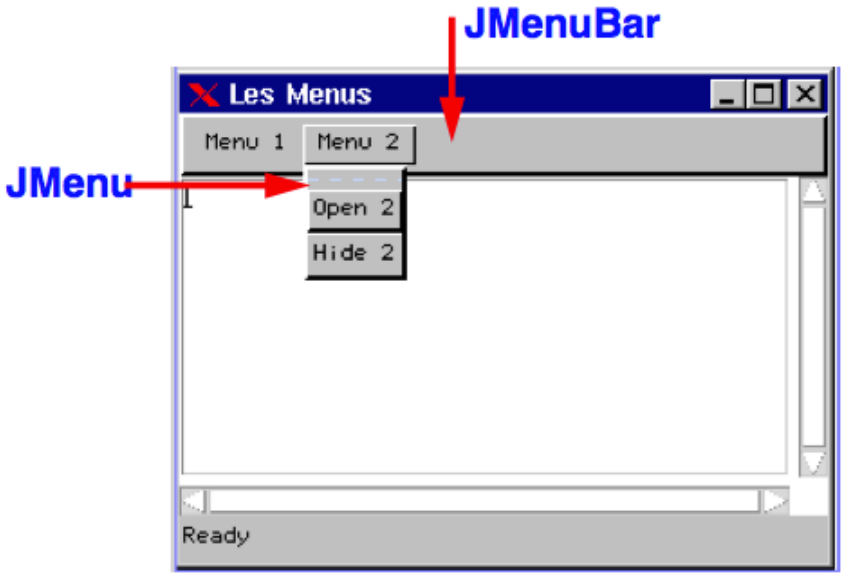
- JOptionPane. showInputDialog(f,"First Name");

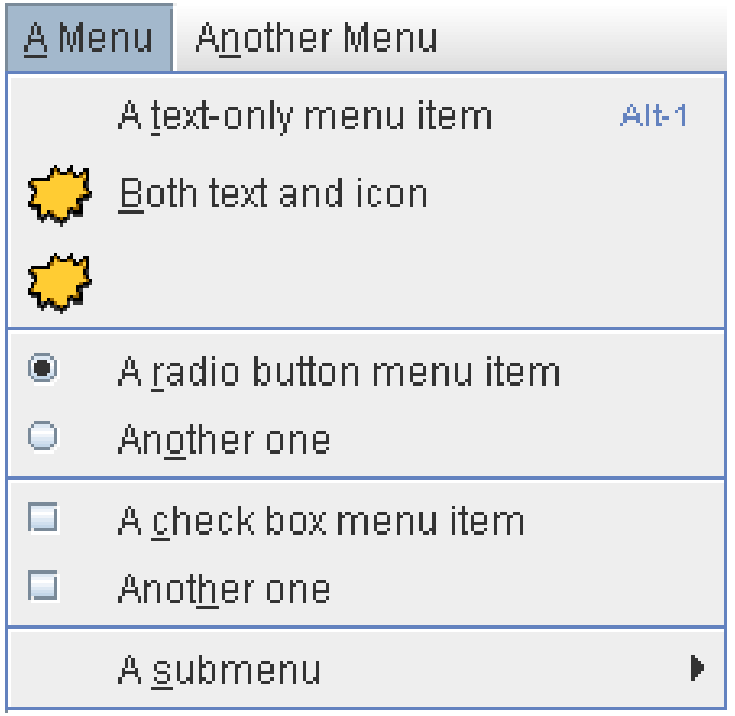


JOptionPane

Menus

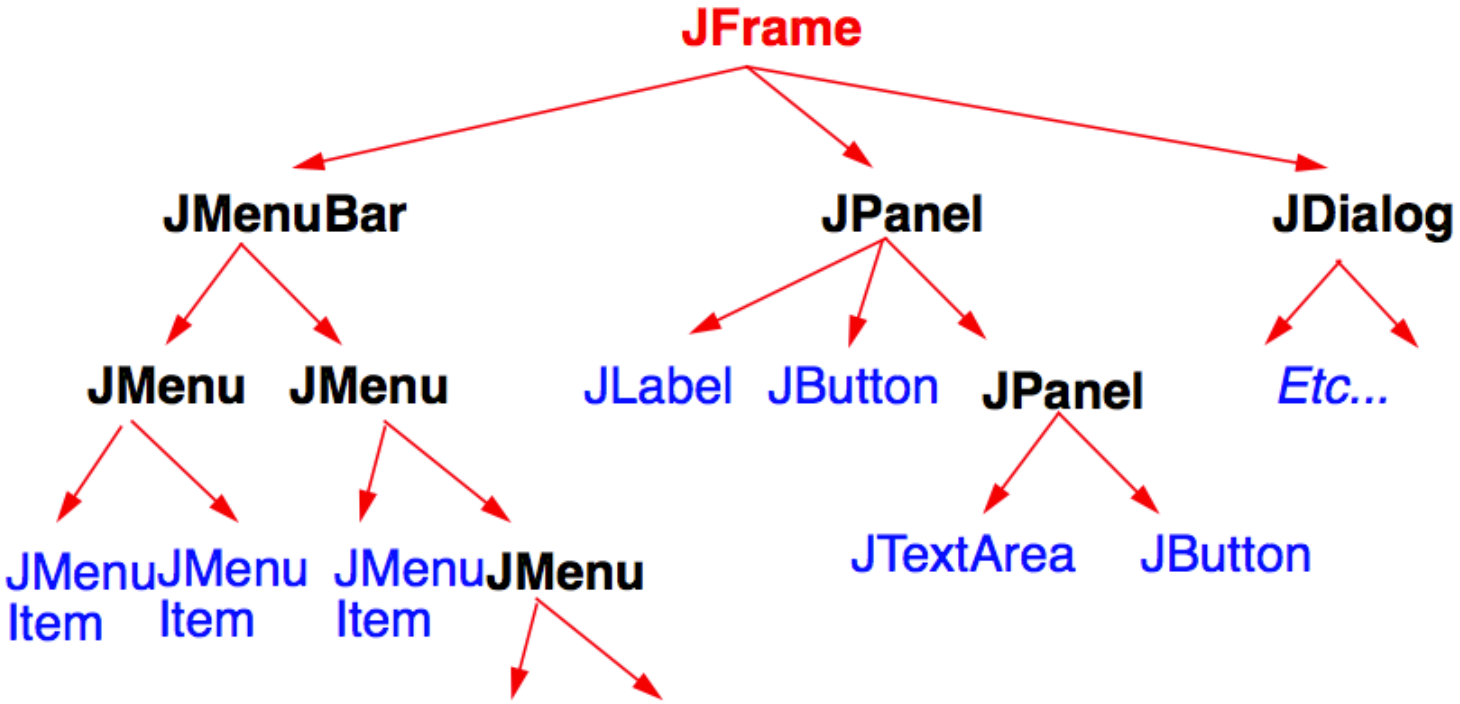

JMenuBar
JComponent
AbstractButton
JMenuItem
JMenu
JCheckBoxMenuItem
JPopupMenu


JMenuBar
JMenu
Les Menus
Menu 1
Menu 2
Open 2
Hide 2
Ready


A Menu
Another Menu
A text-only menu item
Alt-1
Both text and icon
A radio button menu item
Another one
A check box menu item
Another one
A submenu

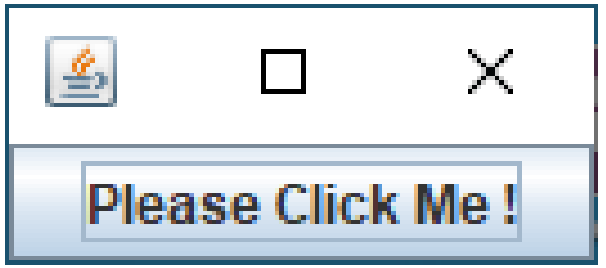
Arbre d'instanciation

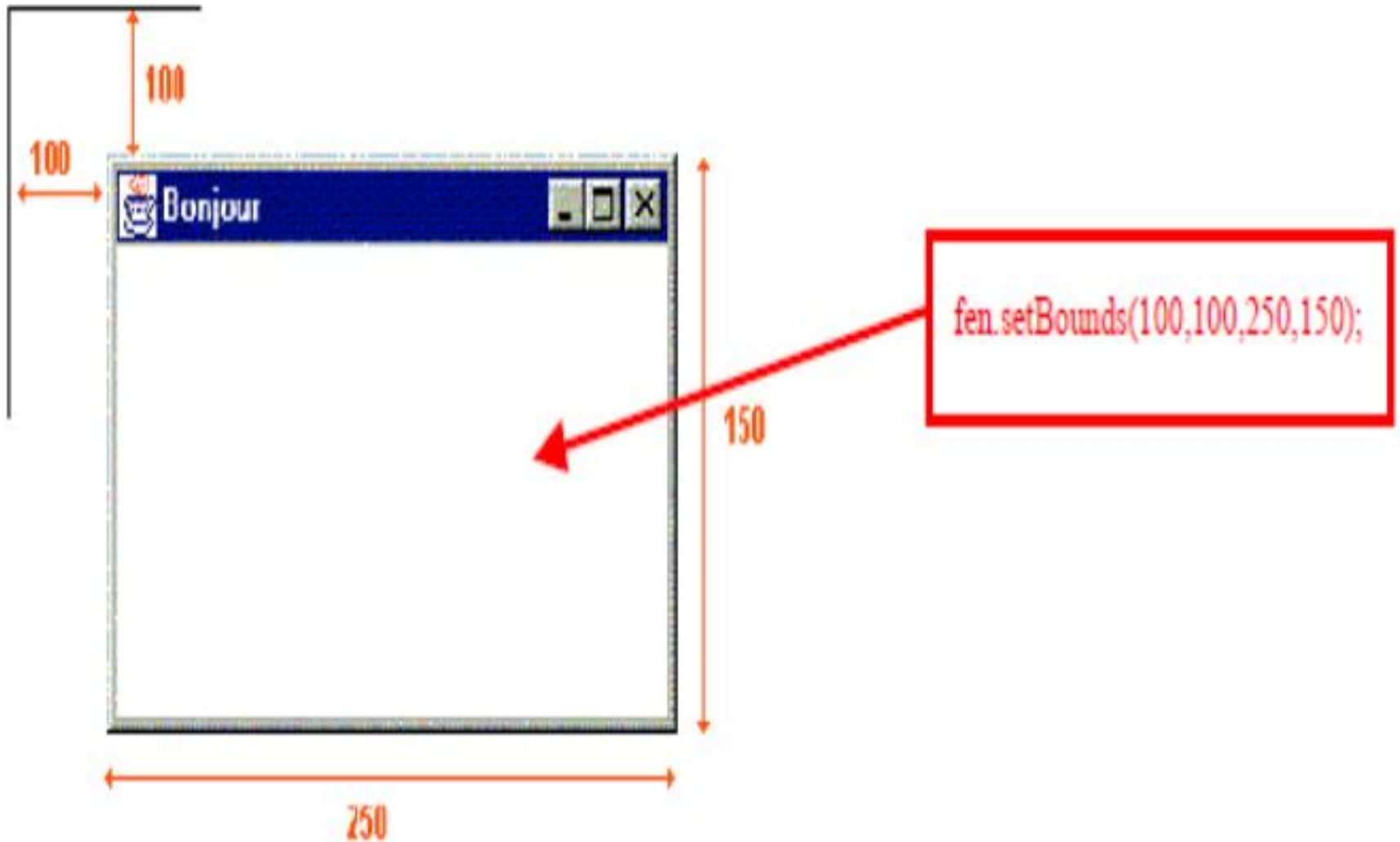
- Chaque objet graphique « contient » ses enfants
  - **superposition** : enfants affichés au dessus des parents
  - **clipping**



Exemple

```
import javax.swing.*;
public class SwingFrame extends JFrame {
        JButton button = null;
        public static void main(String argv[ ]) {
                SwingFrame toplevel = new SwingFrame ();
        }
        public SwingFrame () {
                button = new JButton ("Please Click Me !");
                getContentPane().add(button);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setTitle(« My Frame");
                pack();
                setVisible(true);
        }
}
```




100
100
Bonjour
150
250
fen.setBounds(100,100,250,150);

Disposition spatiale

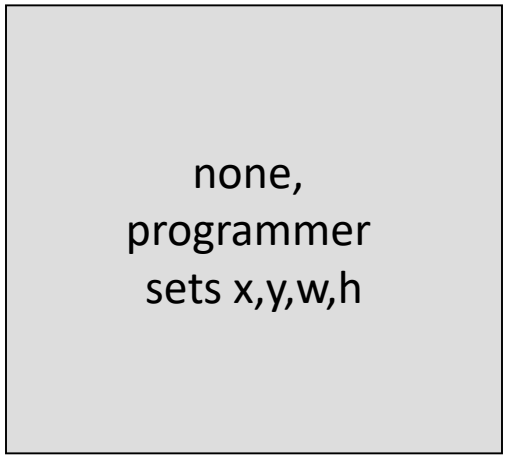
- **Les LayoutManagers**
  - calculent **automatiquement** la **disposition spatiale** des enfants des **Containers**
- **A chaque conteneur est associé un LayoutManager**
  - qui dépend du type de conteneur
  - qui peut être changé par la méthode **setLayout() :** conteneur.**setLayout**(unAutreLayoutManager)
- **Pour faire le calcul "à la main"**
  - à éviter sauf cas particuliers : conteneur.**setLayout**(**null**)

Avantages des LayoutManagers

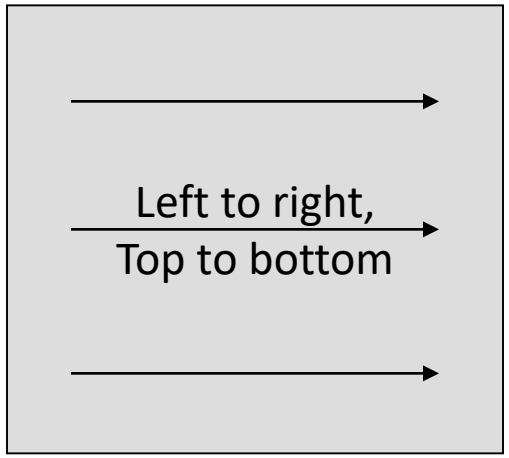
- **C'est plus simple :pas de calculs** compliqués à programmer !
- **Configurabilité**
  - **accessibilité** : indépendance par rapport aux tailles des polices
  - **internationalisation** : indépendance par rapport à la longueur du texte
- **Adaptativité des interfaces :**les composants graphiques se **retaillent automatiquement** quand **l'utilisateur** retaille les fenêtres

Layout Manager

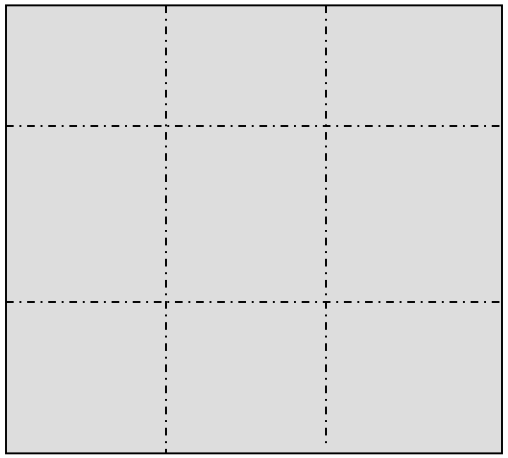
null



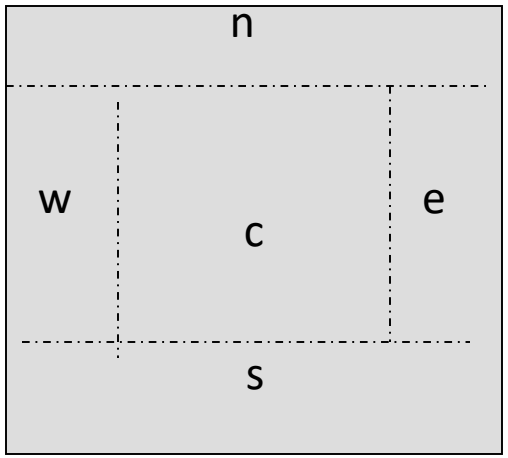
FlowLayout



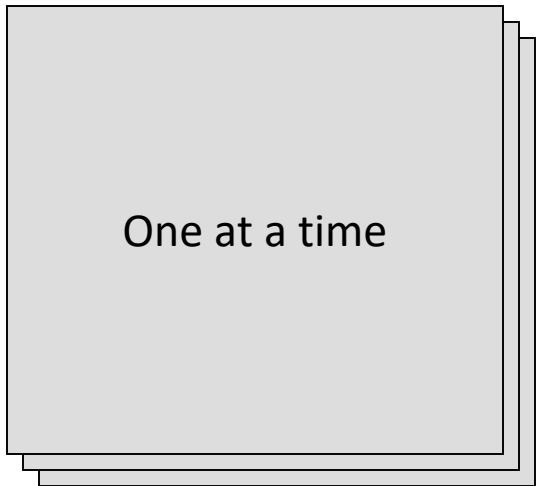
GridLayout



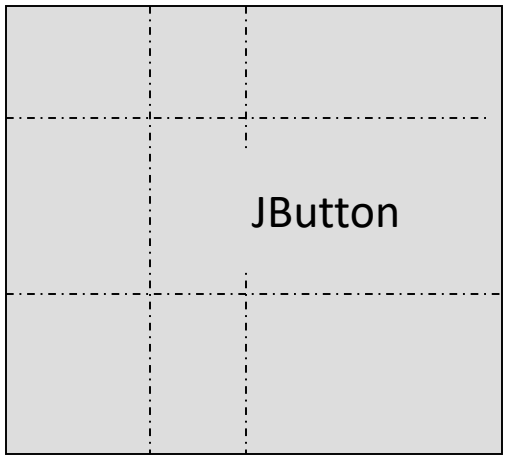
BorderLayout



CardLayout



GridBagLayout

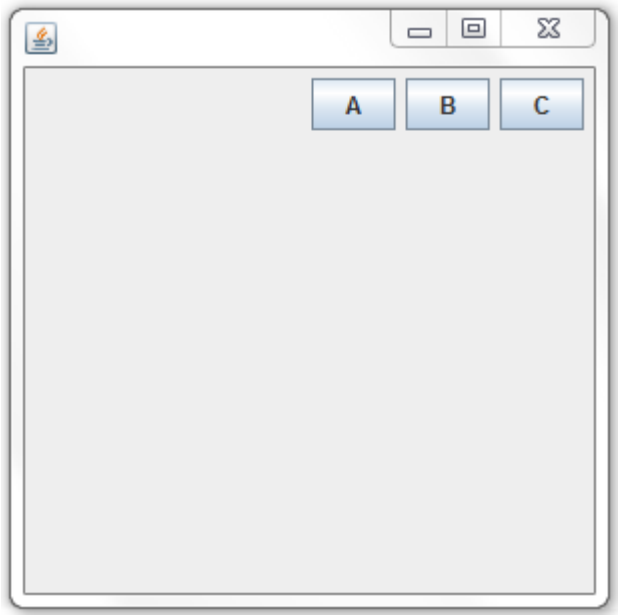


Principaux LayoutManagers

FlowLayout

- défaut des **JPanel**
- met les objets à la suite comme un "flux textuel" dans une page
  - de gauche à droite puis à la ligne

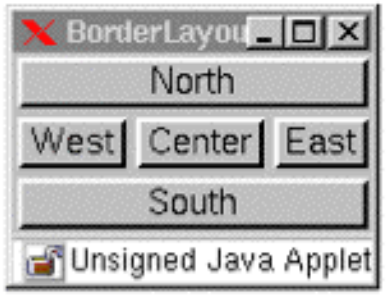
```
import java.awt.*;
import javax.swing.*;
public class DispoExemple{
  DispoExemple()  {
    JFrame frame = new JFrame();
    JButton btn1 = new JButton("A");
    JButton btn2 = new JButton("B");
    JButton btn3 = new JButton("C");
    frame.add(btn1);    frame.add(btn2);    frame.add(btn3);
    //définir la disposition à droite
    frame.setLayout(new FlowLayout(FlowLayout.RIGHT));
    frame.setSize(300,300);
    frame.setVisible(true);
  }
  public static void main(String[] args) {    new DispoExemple();  }
}
```

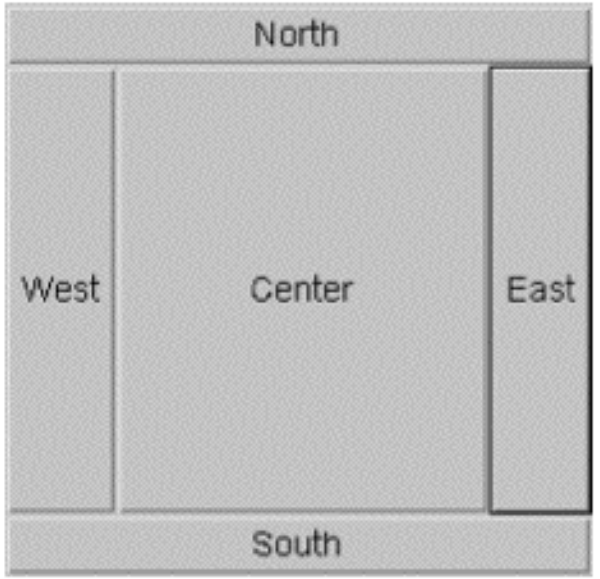


Principaux LayoutManagers

BorderLayout

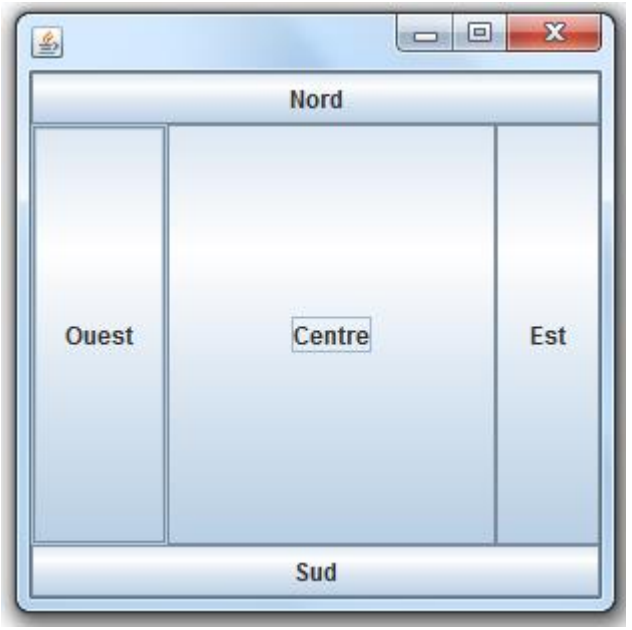
- défaut des **JFrame** et **JDialog**
- retaille **automatiquement** les enfants du conteneur
- disposition de type points cardinaux
  - via constantes: **BorderLayout.CENTER, EAST, NORTH, SOUTH, WEST**





```
import java.awt.*;
import javax.swing.*;
public class DispoExemple{
  DispoExemple()  {
    JFrame frame = new JFrame();
    JButton btn1 = new JButton("Nord");
    JButton btn2 = new JButton("Sud");
    JButton btn3 = new JButton("Est");
    JButton btn4 = new JButton("Ouest");
    JButton btn5 = new JButton("Centre");
    frame.add(btn1, BorderLayout.NORTH);
    frame.add(btn2, BorderLayout.SOUTH);
    frame.add(btn3, BorderLayout.EAST);
    frame.add(btn4, BorderLayout.WEST);
    frame.add(btn5, BorderLayout.CENTER);
    frame.setSize(300,300);
    frame.setVisible(true);
  }
  public static void main(String[] args) { new DispoExemple(); }
}
```

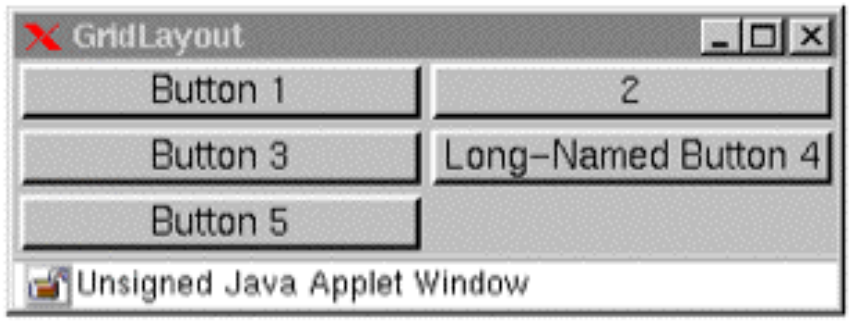
,



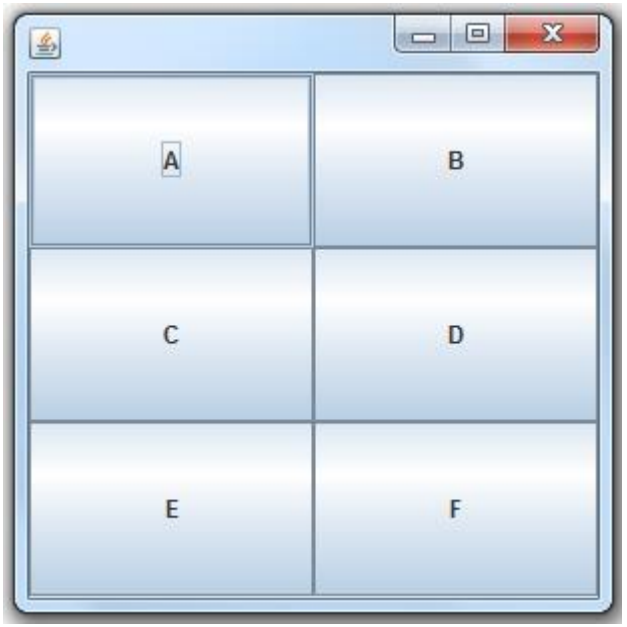
Principaux LayoutManagers

GridLayout

- divise le conteneur en cellules de même taille (grille virtuelle)
  - de gauche à droite et de haut en bas
- retaille automatiquement les enfants



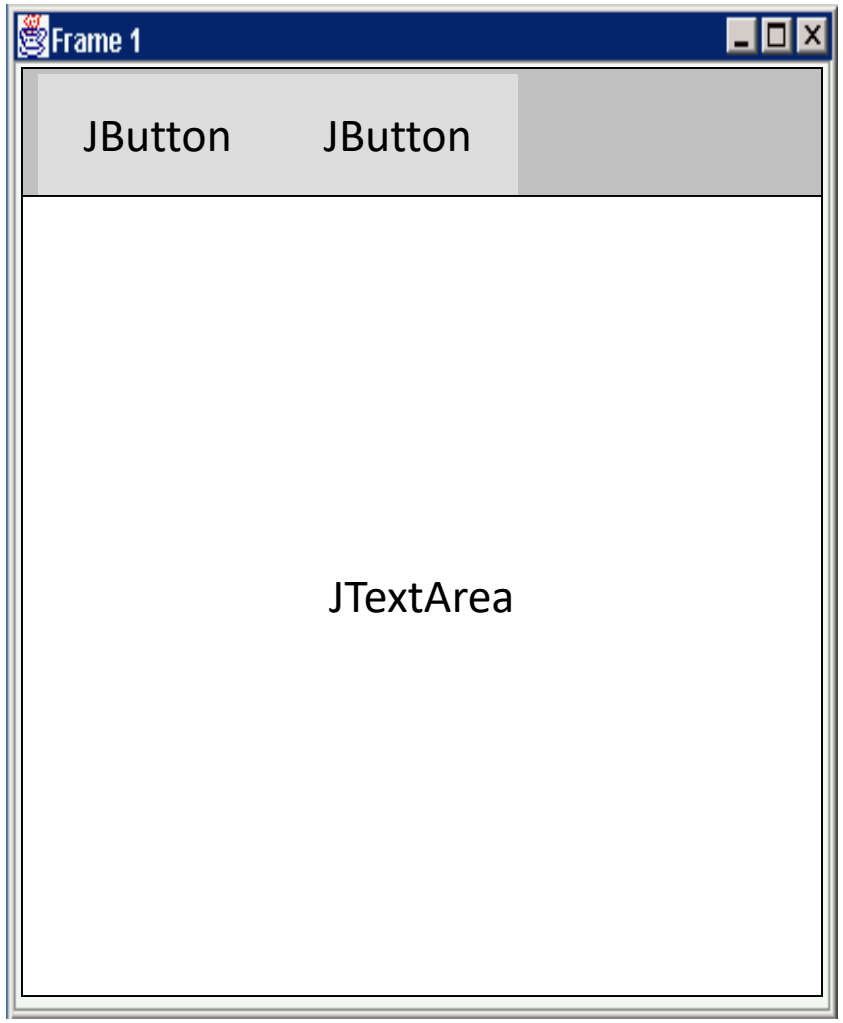
```
import java.awt.*;
import javax.swing.*;
public class DispoExemple{
  DispoExemple()  {
    JFrame frame = new JFrame();
    JButton btn1 = new JButton("A");     JButton btn2 = new JButton("B");
    JButton btn3 = new JButton("C");     JButton btn4 = new JButton("D");
    JButton btn5 = new JButton("E");     JButton btn6 = new JButton("F");
    frame.add(btn1);    frame.add(btn2);
    frame.add(btn3);    frame.add(btn4);
    frame.add(btn5);    frame.add(btn6);
    //définir la disposition en grille de 3 lignes et 2 colonnes
    frame.setLayout(new GridLayout(3,2));
    frame.setSize(300,300);
    frame.setVisible(true);
  }
  public static void main(String[] args) {    new DispoExemple();  }
}
```



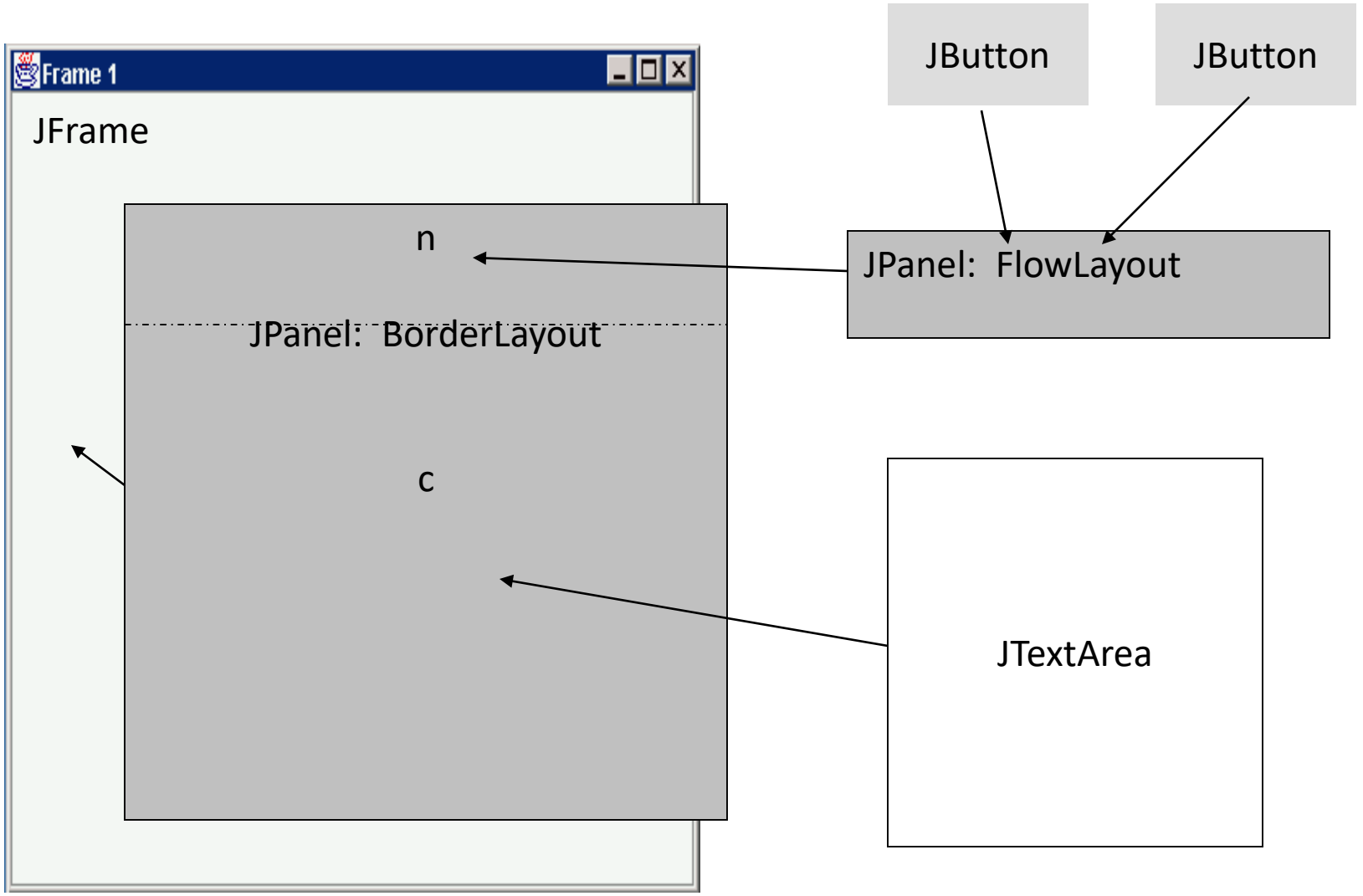
Principaux LayoutManagers

- **GridBagLayout**
- **CardLayout**
- **BoxLayout**
- **SpringLayout**

Combinations



Combinations



Code: null layout

```
JFrame f = new JFrame("title");
JPanel p = new JPanel( );
JButton b = new JButton("press me");

b.setBounds(new Rectangle(10,10, 100,50));
p.setLayout(null);          // x,y layout
p.add(b);
f.setContentPane(p);
```