



Java Avancé - Cours 1 (suite)

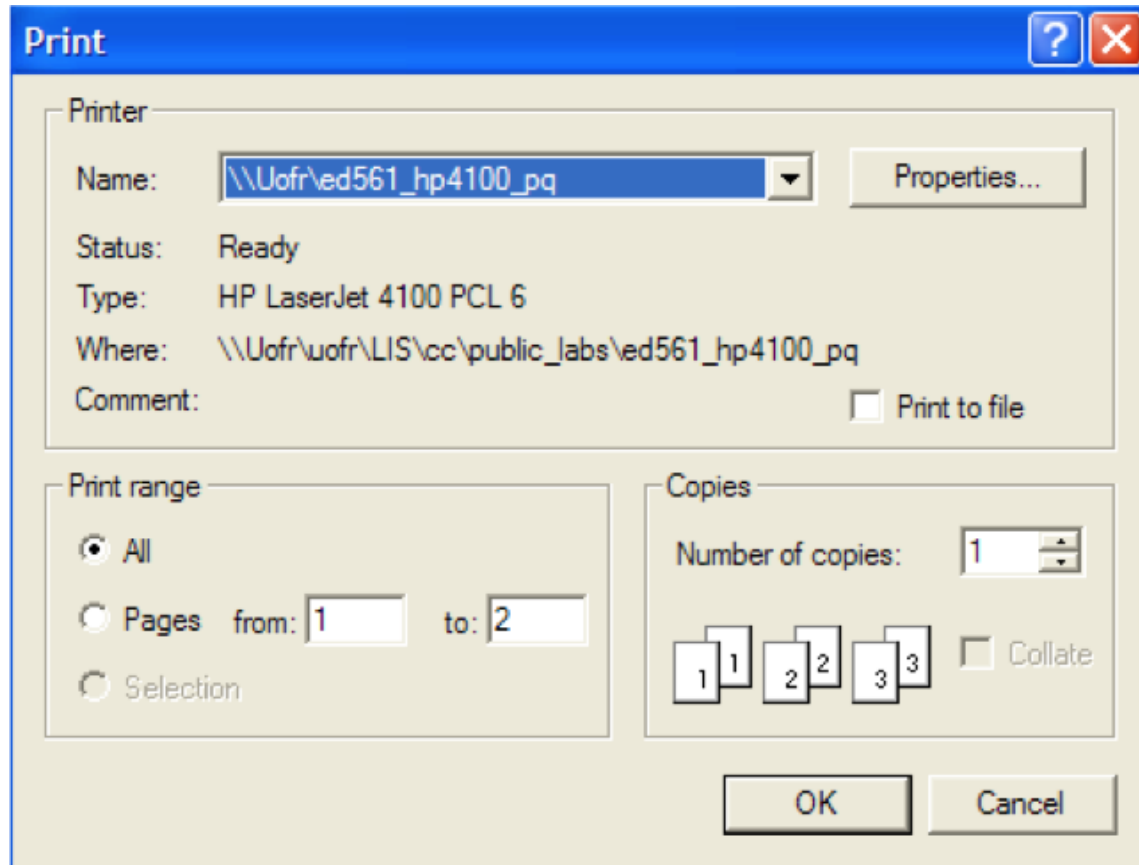
V. DESLANDRES, I. GUIDARA

veronique.deslandres@univ-lyon1.fr

Avril 2018

Sommaire de ce cours

- Placer les composants (layout) 3
 - GridLayout 9
 - BorderLayout 12
 - FlowLayout 15
 - AbsoluteLayout 20



Placer les composants dans la fenêtre (Layout)

Répartition des composants

- Les composants sont **toujours placés dans** un conteneur
 - souvent un **panneau (JPanel)**
- On peut définir n panneaux sur une même fenêtre, par ex. :
 - Panneaux ligne précédent,
 - Panneau colonne des couleurs,
 - Panneau des boutons d'interactions avec l'utilisateur, etc.
- On peut les rendre **visible** ou pas
- Tous les containers de tête dans la hiérarchie (JFrame, JWindow,...) disposent d'un container intermédiaire qui reçoit les éléments fils :
 - `frame.getContentPane()` ;
- On peut **convertir** le conteneur de la fenêtre en panneau par :
 - `JPanel c = (JPanel) getContentPane()` ; *par défaut, retourne un Container*

Fenêtre sans Layout

```
public class Fenetre extends JFrame{
    JLabel label1, label2;
    JTextField text1, text2;
    JButton bouton1, bouton2;
    JPanel panel;
    public Fenetre(){
        //Création des composants
        label1=new JLabel("Label 1");
        label2=new JLabel("Label 2");
        text1=new JTextField("Texte 1");
        text2=new JTextField("Texte 2");
        bouton1=new JButton("Bouton 1");
        bouton2=new JButton("Bouton 2");
        //Par défaut le conteneur est de classe container
        //on le transtype à JPanel
        panel=(JPanel) getContentPane();
        //Ajout des composants
        panel.add(label1);
        panel.add(label2);
        panel.add(text1);
        panel.add(text2);
        panel.add(bouton1);
        panel.add(bouton2);
    }
    public static void main (String[] args){
        JFrame fen=new Fenetre();
        fen.setSize(300,100); //fen.pack();
        fen.setTitle("Une fenêtre avec composants");
        fen.setVisible(true);
    }
}
```

3 étapes

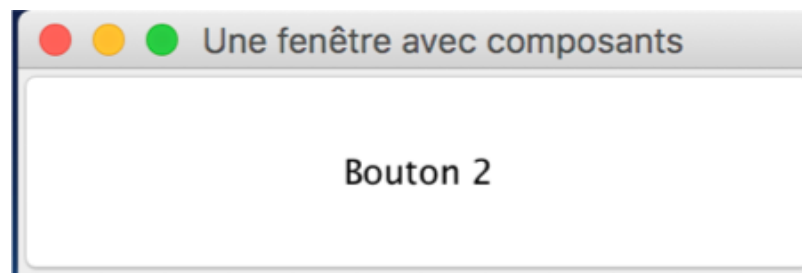
1- Déclaration des **composants**

2- Déclaration du **conteneur**

3- Insertion des **composants** dans le **conteneur**

Répartition des composants

- Pour placer un composant : méthode `add` de `JPanel`
- PROBLÈME : la méthode `add` ajoute toujours le composant **au même endroit** dans le conteneur
 - Ainsi lorsqu'on veut ajouter plusieurs composants dans le panel, seul le dernier composant apparaît
 - Il faut donc **répartir** les composants avec la méthode `setLayout()`

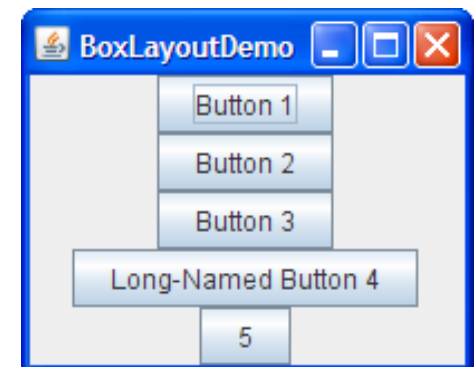
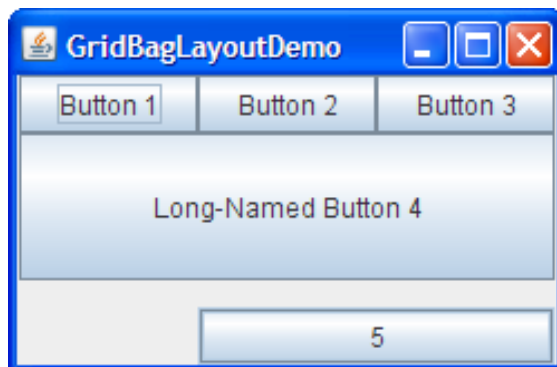
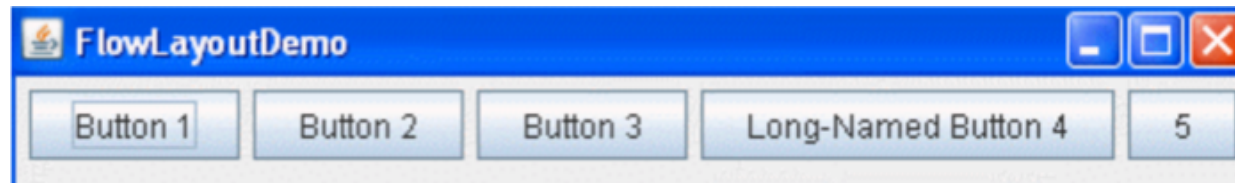
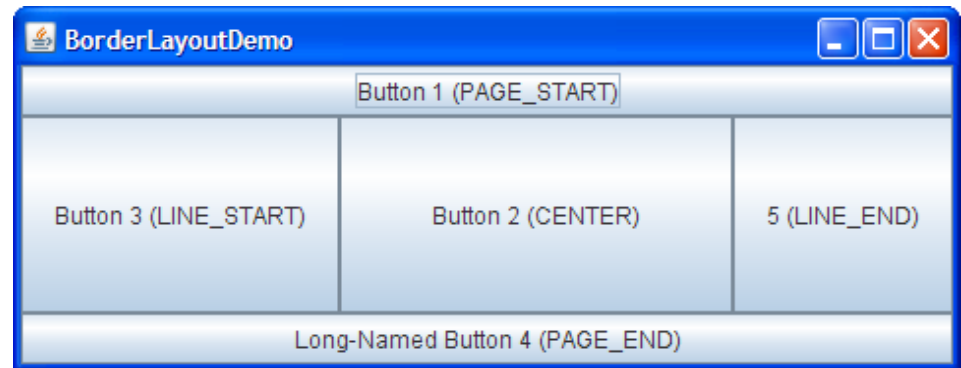


Répartition des composants (Layout)

- Pour gérer la disposition des composants
 - Il existe des modes de répartitions *prédéfinis*
 - On choisit le **gestionnaire de répartition** avec `setLayout(monLayout)`
- Les gestionnaires de répartition appartiennent au package `java.awt`
 - `GridLayout`
 - `BorderLayout` (disposition par défaut dans une `JFrame`)
 - `FlowLayout` (disposition par défaut dans un `JPanel`)
 - `xxxLayout`

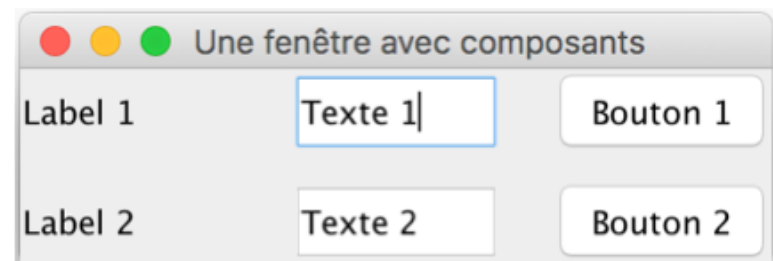
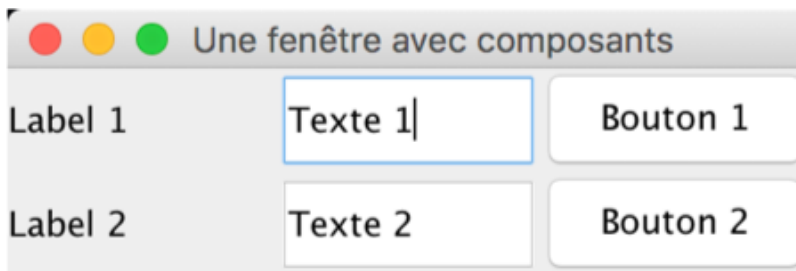
Gestionnaire de placement

- Chaque conteneur utilise un gestionnaire de placement (Layout) pour disposer ses composants



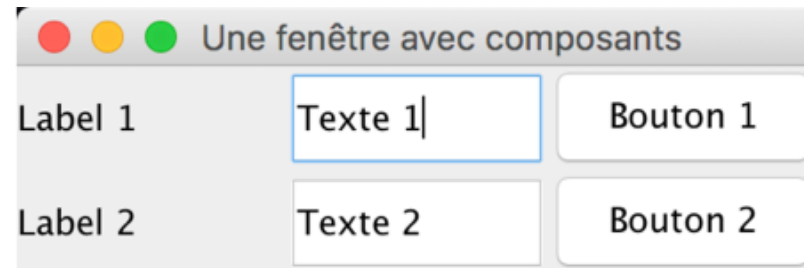
GridLayout(1/3)

- Place les composants **dans un tableau**
- Possède 3 constructeurs :
 - `GridLayout()`
 - `GridLayout(int rows, int col)`
 - *rows* = nombre de lignes et *col* = nombre de colonnes
 - `GridLayout(int rows, int col, int hgap, int vgap)`
 - *hgap* est l'espacement **horizontal** (d'une colonne à une autre)
 - *vgap* est l'espacement **vertical** (d'une ligne à une autre)
- On transmet à la méthode `add` le **nom** du composant
 - `panel.setLayout(new GridLayout(2, 3));`
 - `panel.add(bouton1);`



GridLayout(2/3)

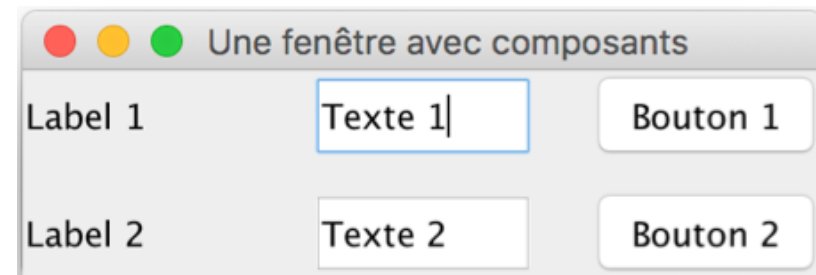
```
public class FenetreGridLayout extends JFrame{
    JLabel label1, label2;
    JTextField text1, text2;
    JButton bouton1, bouton2;
    JPanel panel;
    public FenetreGridLayout(){
        //Création des composants
        label1=new JLabel("Label 1");
        label2=new JLabel("Label 2");
        text1=new JTextField("Texte 1");
        text2=new JTextField("Texte 2");
        bouton1=new JButton("Bouton 1");
        bouton2=new JButton("Bouton 2");
        //Par défaut le conteneur est de classe container
        //on le transtype à JPanel
        panel=(JPanel) getContentPane();
        panel.setLayout(new GridLayout(2,3));
        //Ajout des composants
        panel.add(label1);
        panel.add(text1);
        panel.add(bouton1);
        panel.add(label2);
        panel.add(text2);
        panel.add(bouton2);
    }
    public static void main (String[] args){
        JFrame fen=new FenetreGridLayout();
        fen.setSize(300,100);
        fen.setTitle("Une fenêtre avec composants");
        fen.setVisible(true);
    }
}
```



GridLayout(3/3)

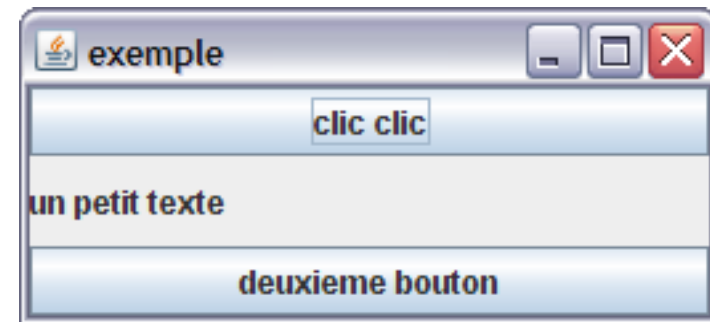
2 lignes et 3 colonnes espacées de 20 pixels entre les colonnes et de 10 pixels entre les lignes

```
public class FenetreGridLayout extends JFrame{
    JLabel label1, label2;
    JTextField text1, text2;
    JButton bouton1, bouton2;
    JPanel panel;
    public FenetreGridLayout(){
        //Création des composants
        label1=new JLabel("Label 1");
        label2=new JLabel("Label 2");
        text1=new JTextField("Texte 1");
        text2=new JTextField("Texte 2");
        bouton1=new JButton("Bouton 1");
        bouton2=new JButton("Bouton 2");
        //Par défaut le conteneur est de classe contain
        //on le transtype à JPanel
        panel=(JPanel) getContentPane();
        panel.setLayout(new GridLayout(2,3, 20, 10));
        //Ajout des composants
        panel.add(label1);
        panel.add(text1);
        panel.add(bouton1);
        panel.add(label2);
        panel.add(text2);
        panel.add(bouton2);
    }
    public static void main (String[] args){
        JFrame fen=new FenetreGridLayout();
        fen.setSize(300,100);
        fen.setTitle("Une fenêtre avec composants");
        fen.setVisible(true);
    }
}
```



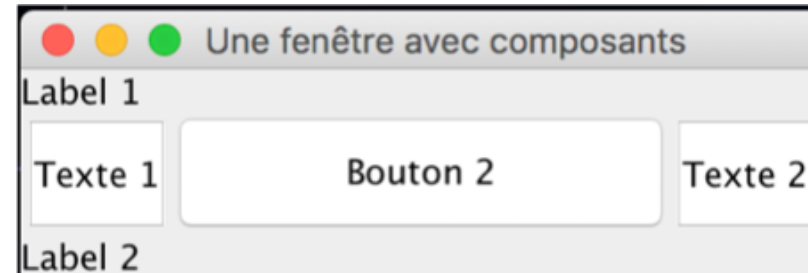
BorderLayout (1/3)

- Positionnement par défaut d'une `JFrame`
- Découpe l'écran en 5 régions
 - **south north east west center**
- On peut n'en utiliser que certaines
- Il n'y a **qu'un seul composant** par région
- Possède 2 constructeurs :
 - `BorderLayout()`
 - `BorderLayout(int hgap, int vgap)`
- On transmet à la méthode `add` **l'emplacement** et le **nom** du composant
 - `c.setLayout(new BorderLayout(10, 5));`
 - `c.add(bouton1, BorderLayout.NORTH);`



BorderLayout (2/3)

```
public class FenetreBorderLayout extends JFrame{
    JLabel label1, label2;
    JTextField text1, text2;
    JButton bouton1, bouton2;
    JPanel panel;
    public FenetreBorderLayout(){
        //Création des composants
        label1=new JLabel("Label 1");
        label2=new JLabel("Label 2");
        text1=new JTextField("Texte 1");
        text2=new JTextField("Texte 2");
        bouton1=new JButton("Bouton 1");
        bouton2=new JButton("Bouton 2");
        //Par défaut le conteneur est de classe container
        //on le transtype à JPanel
        panel=(JPanel) getContentPane();
        panel.setLayout(new BorderLayout());
        //Ajout des composants
        panel.add(label1, BorderLayout.NORTH);
        panel.add(text1, BorderLayout.WEST);
        panel.add(bouton1, BorderLayout.CENTER);
        panel.add(label2, BorderLayout.SOUTH);
        panel.add(text2, BorderLayout.EAST);
    }
    public static void main (String[] args){
        JFrame fen=new FenetreBorderLayout();
        fen.setSize(300,100);
        fen.setTitle("Une fenêtre avec composants");
        fen.setVisible(true);
    }
}
```



BorderLayout (3/3)

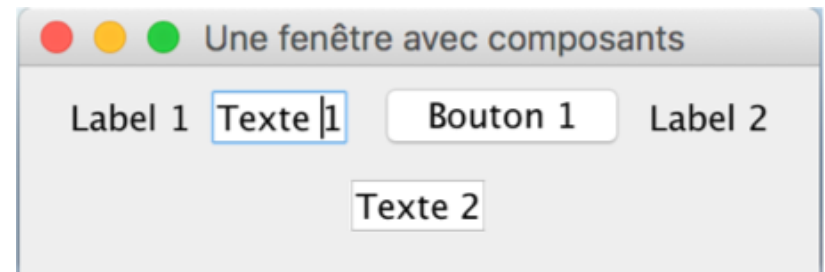
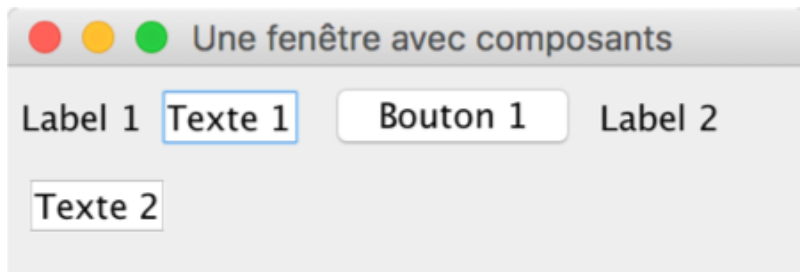


Si on veut mettre plus d'un composant dans une région ?

- Est-ce **possible** ?
- Comment procéder ?

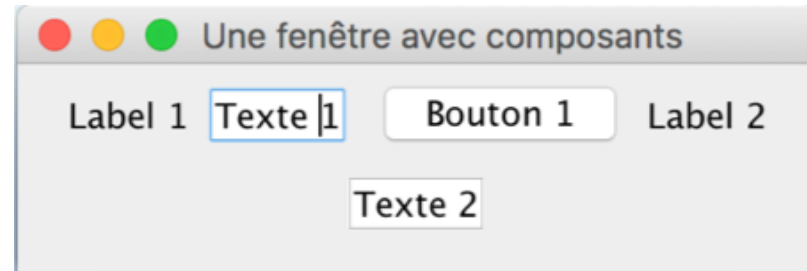
FlowLayout (1/5)

- Place les composants *les uns à la suite des autres*, de façon centrée
 - ligne par ligne, en passant à la ligne suivante si nécessaire
- C'est une classe, avec 3 constructeurs :
 - `FlowLayout()` *(LEFT par défaut)*
 - `FlowLayout(int align)`
 - Alignement (`FlowLayout.LEFT` ou `FlowLayout.RIGHT` ou `FlowLayout.CENTER`)
 - `FlowLayout(int align, int hgap, int vgap)`
- On transmet à la méthode `add` le **nom** du composant
 - `panel.setLayout(new FlowLayout());`
 - `panel.add(bouton1);`



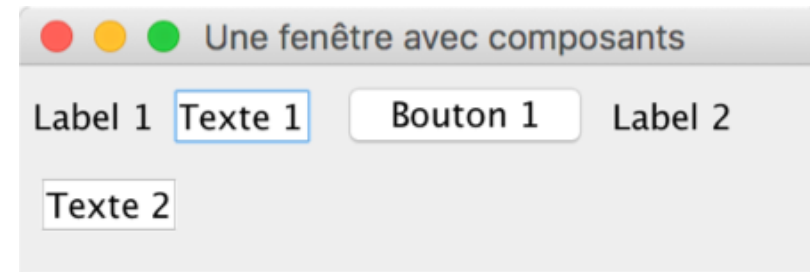
FlowLayout (2/5)

```
public class FenetreFlowLayout extends JFrame{
    JLabel label1, label2;
    JTextField text1, text2;
    JButton bouton1, bouton2;
    JPanel panel;
    public FenetreFlowLayout(){
        //Création des composants
        label1=new JLabel("Label 1");
        label2=new JLabel("Label 2");
        text1=new JTextField("Texte 1");
        text2=new JTextField("Texte 2");
        bouton1=new JButton("Bouton 1");
        bouton2=new JButton("Bouton 2");
        //Par défaut le conteneur est de classe container
        //on le transtype à JPanel
        panel=(JPanel) getContentPane();
        panel.setLayout(new FlowLayout());
        //Ajout des composants
        panel.add(label1);
        panel.add(text1);
        panel.add(bouton1);
        panel.add(label2);
        panel.add(text2);
    }
    public static void main (String[] args){
        JFrame fen=new FenetreFlowLayout();
        fen.setSize(300,100);
        fen.setTitle("Une fenêtre avec composants");
        fen.setVisible(true);
    }
}
```



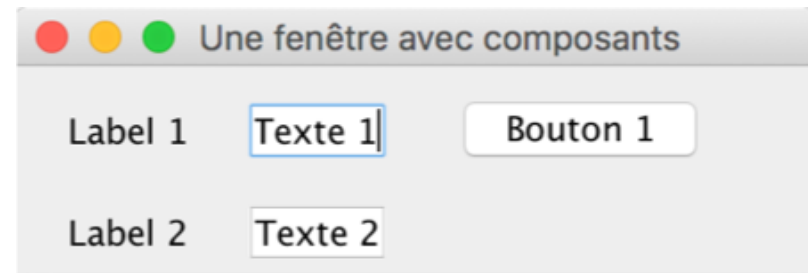
FlowLayout (3/5)

```
public class FenetreFlowLayout extends JFrame{
    JLabel label1, label2;
    JTextField text1, text2;
    JButton bouton1, bouton2;
    JPanel panel;
    public FenetreFlowLayout(){
        //Création des composants
        label1=new JLabel("Label 1");
        label2=new JLabel("Label 2");
        text1=new JTextField("Texte 1");
        text2=new JTextField("Texte 2");
        bouton1=new JButton("Bouton 1");
        bouton2=new JButton("Bouton 2");
        //Par défaut le conteneur est de classe container
        //on le transtype à JPanel
        panel=(JPanel) getContentPane();
        panel.setLayout(new FlowLayout(FlowLayout.LEFT));
        //Ajout des composants
        panel.add(label1);
        panel.add(text1);
        panel.add(bouton1);
        panel.add(label2);
        panel.add(text2);
    }
    public static void main (String[] args){
        JFrame fen=new FenetreFlowLayout();
        fen.setSize(300,100);
        fen.setTitle("Une fenêtre avec composants");
        fen.setVisible(true);
    }
}
```



FlowLayout (4/5)

```
public class FenetreFlowLayout extends JFrame{
    JLabel label1, label2;
    JTextField text1, text2;
    JButton bouton1, bouton2;
    JPanel panel;
    public FenetreFlowLayout(){
        //Création des composants
        label1=new JLabel("Label 1");
        label2=new JLabel("Label 2");
        text1=new JTextField("Texte 1");
        text2=new JTextField("Texte 2");
        bouton1=new JButton("Bouton 1");
        bouton2=new JButton("Bouton 2");
        //Par défaut le conteneur est de classe container
        //on le transtype à JPanel
        panel=(JPanel) getContentPane();
        panel.setLayout(new FlowLayout(3, 20, 10));
        //Ajout des composants
        panel.add(label1);
        panel.add(text1);
        panel.add(bouton1);
        panel.add(label2);
        panel.add(text2);
    }
    public static void main (String[] args){
        JFrame fen=new FenetreFlowLayout();
        fen.setSize(300,100);
        fen.setTitle("Une fenêtre avec composants");
        fen.setVisible(true);
    }
}
```



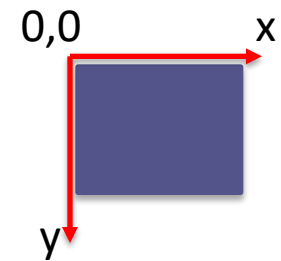
FlowLayout (5/5)

```
public class FenetreFlowLayout extends JFrame{
    JLabel label1, label2;
    JTextField text1, text2;
    JButton bouton1, bouton2;
    JPanel panel;
    public FenetreFlowLayout(){
        //Création des composants
        label1=new JLabel("Label 1");
        label2=new JLabel("Label 2");
        text1=new JTextField("Texte 1");
        text2=new JTextField("Texte 2");
        bouton1=new JButton("Bouton 1");
        bouton2=new JButton("Bouton 2");
        //Par défaut le conteneur est de classe container
        //on le transtype à JPanel
        panel=(JPanel) getContentPane();
        panel.setLayout(new FlowLayout(FlowLayout.CENTER, 20, 10));
        //Ajout des composants
        panel.add(label1);
        panel.add(text1);
        panel.add(bouton1);
        panel.add(label2);
        panel.add(text2);
    }
    public static void main (String[] args){
        JFrame fen=new FenetreFlowLayout();
        fen.setSize(300,100);
        fen.setTitle("Une fenêtre avec composants");
        fen.setVisible(true);
    }
}
```



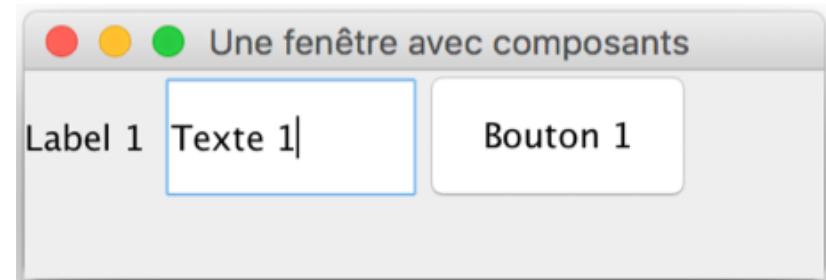
Absolute layout (1/2)

- Pour placer les composants à un endroit précis, il faut indiquer **qu'on n'utilise pas de *LayoutManager*** :
 - `ObjetConteneur.setLayout(null);`
- On indique alors les coordonnées et la taille de chaque composant :
 - `Composant.setBounds(x, y, larg, haut);`
 - `x` est le déplacement par rapport au bord gauche du conteneur
 - `y` est le déplacement par rapport au bord haut du conteneur
 - `larg` est la largeur du composant
 - `haut` est la hauteur du composant
- LIMITE : ici les composants **ne sont pas redimensionnés** en cas de modification de la taille de la fenêtre



Absolute Layout (2/2)

```
public class AbsoluteLayout extends JFrame{
    JLabel label1, label2;
    JTextField text1, text2;
    JButton bouton1, bouton2;
    JPanel panel;
    public AbsoluteLayout(){
        //Création des composants
        label1=new JLabel("Label 1");
        label2=new JLabel("Label 2");
        text1=new JTextField("Texte 1");
        text2=new JTextField("Texte 2");
        bouton1=new JButton("Bouton 1");
        bouton2=new JButton("Bouton 2");
        //Par défaut le conteneur est de classe container
        //on le transtype à JPanel
        panel=(JPanel) getContentPane();
        panel.setLayout(null); // positionnement absolu
        //Ajout des composants
        label1.setBounds(0,0,100,50);
        panel.add(label1);
        text1.setBounds(50,0,100,50);
        panel.add(text1);
        bouton1.setBounds(150,0,100,50);
        panel.add(bouton1);
    }
    public static void main (String[] args){
        JFrame fen=new AbsoluteLayout();
        fen.setSize(300,100);
        fen.setTitle("Une fenêtre avec composants");
        fen.setVisible(true);
    }
}
```



Panneaux Imbriqués

Panneaux imbriqués

- Si on veut mettre plus d'un composant dans une zone, on utilise un **panneau** placé dans la zone
 - dans lequel on positionne les composants
- Un container peut être inséré dans un autre container, souvent des **JPanel**
- Pour ajouter un panneau à un autre panneau il faut utiliser la méthode `add`

BorderLayout

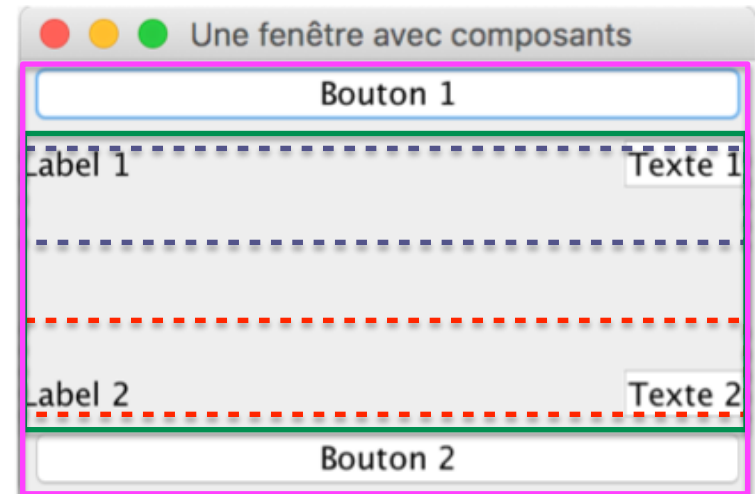
```
public class PanneauxImbriquésBorder extends JFrame{
    JLabel label1, label2;
    JTextField text1, text2;
    JButton bouton1, bouton2;
    JPanel panel1, panel2, panel3, panel4;
    public PanneauxImbriquésBorder(){
        //Création des composants
        label1=new JLabel("Label 1");
        label2=new JLabel("Label 2");
        text1=new JTextField("Texte 1");
        text2=new JTextField("Texte 2");
        bouton1=new JButton("Bouton 1");
        bouton2=new JButton("Bouton 2");
        panel1=(JPanel) getContentPane();
        panel1.setLayout(new BorderLayout());
        panel2= new JPanel(new BorderLayout());
        panel3= new JPanel(new BorderLayout());
        panel4= new JPanel(new BorderLayout());
        //Ajout des composants dans panel2
        panel2.add(label1, BorderLayout.WEST);
        panel2.add(text1, BorderLayout.EAST);
        //Ajout des composants dans panel3
        panel3.add(label2, BorderLayout.WEST);
        panel3.add(text2, BorderLayout.EAST);
        //Ajout des composants dans panel4
        panel4.add(panel2, BorderLayout.NORTH);
        panel4.add(panel3, BorderLayout.SOUTH);
        //Ajout des composants dans panel1
        panel1.add(bouton1, BorderLayout.NORTH);
        panel1.add(panel4, BorderLayout.CENTER);
        panel1.add(bouton2, BorderLayout.SOUTH);
    }
    public static void main (String[] args){
        JFrame fen=new PanneauxImbriquésBorder();
        fen.setSize(300,200);
        fen.setTitle("Une fenêtre avec composants");
        fen.setVisible(true);
    }
}
```

Panel 2

Panel 3

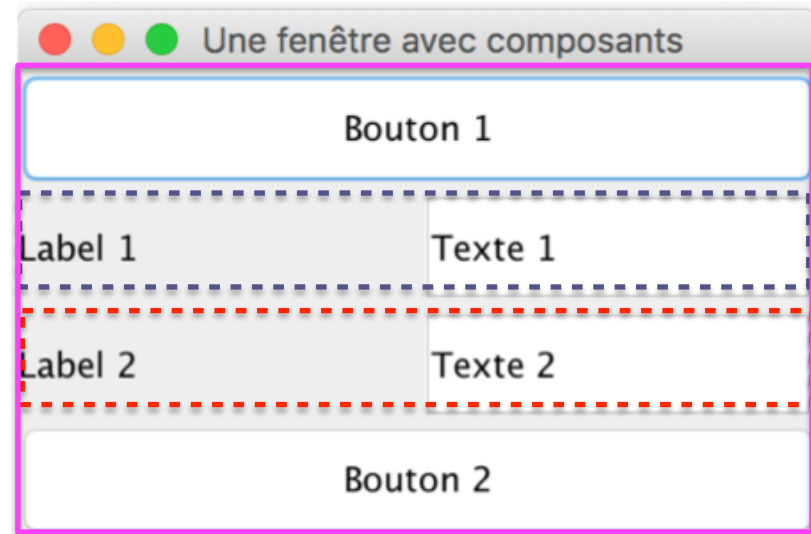
Panel 4

Panel 1



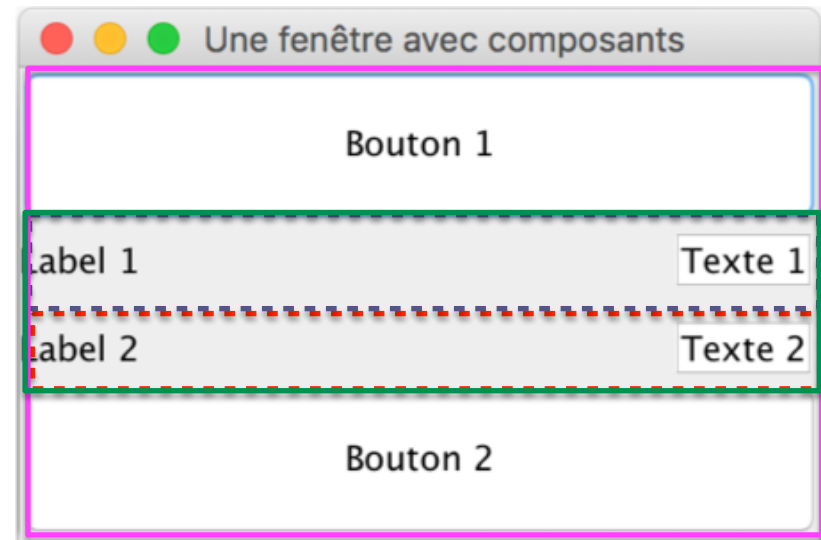
GridLayout

```
public class PanneauxImbriquésGrid extends JFrame{
    JLabel label1, label2;
    JTextField text1, text2;
    JButton bouton1, bouton2;
    JPanel panel1, panel2, panel3;
    public PanneauxImbriquésGrid(){
        //Création des composants
        label1=new JLabel("Label 1");
        label2=new JLabel("Label 2");
        text1=new JTextField("Texte 1");
        text2=new JTextField("Texte 2");
        bouton1=new JButton("Bouton 1");
        bouton2=new JButton("Bouton 2");
        panel1=(JPanel) getContentPane();
        panel1.setLayout(new GridLayout(4,1));
        panel2= new JPanel(new GridLayout(1,2));
        panel3= new JPanel(new GridLayout(1,2));
        //Ajout des composants dans panel2
        panel2.add(label1);
        panel2.add(text1);
        //Ajout des composants dans panel3
        panel3.add(label2);
        panel3.add(text2);
        //Ajout des composants dans panel1
        panel1.add(bouton1);
        panel1.add(panel2);
        panel1.add(panel3);
        panel1.add(bouton2);
    }
    public static void main (String[] args){
        JFrame fen=new PanneauxImbriquésGrid();
        fen.setSize(300,200);
        fen.setTitle("Une fenêtre avec composants");
        fen.setVisible(true);
    }
}
```



Grid & BorderLayout

```
public class PanneauxImbriquésBorderGrid extends JFrame{
    JLabel label1, label2;
    JTextField text1, text2;
    JButton bouton1, bouton2;
    JPanel panel1, panel2, panel3, panel4;
    public PanneauxImbriquésBorderGrid(){
        //Création des composants
        label1=new JLabel("Label 1");
        label2=new JLabel("Label 2");
        text1=new JTextField("Texte 1");
        text2=new JTextField("Texte 2");
        bouton1=new JButton("Bouton 1");
        bouton2=new JButton("Bouton 2");
        panel1=(JPanel) getContentPane();
        panel1.setLayout(new GridLayout(3,1));
        panel2= new JPanel(new BorderLayout());
        panel3= new JPanel(new BorderLayout());
        panel4= new JPanel(new BorderLayout());
        //Ajout des composants dans panel2
        panel2.add(label1, BorderLayout.WEST);
        panel2.add(text1, BorderLayout.EAST);
        //Ajout des composants dans panel3
        panel3.add(label2, BorderLayout.WEST);
        panel3.add(text2, BorderLayout.EAST);
        //Ajout des composants dans panel4
        panel4.add(panel2, BorderLayout.NORTH);
        panel4.add(panel3, BorderLayout.SOUTH);
        //Ajout des composants dans panel1
        panel1.add(bouton1);
        panel1.add(panel4);
        panel1.add(bouton2);
    }
    public static void main (String[] args){
        JFrame fen=new PanneauxImbriquésBorderGrid();
        fen.setSize(300,200);
        fen.setTitle("Une fenêtre avec composants");
        fen.setVisible(true);
    }
}
```



Affichage d'une image

- Pour afficher une image sur un panneau, on utilise un label
- 2 méthodes:
 - En utilisant l'objet `ImageIcon` directement dans la déclaration du label:

```
JLabel lbl_image=new JLabel(new ImageIcon("src/gif/IUT.png"));
```

- La méthode `setIcon()` qui attend un objet `ImageIcon` :
 - `monLabel.setIcon(new ImageIcon(fichier_image));`

```
JLabel lbl_image=new JLabel();
```

```
lbl_image.setIcon(new ImageIcon("src/gif/IUT.png"));
```

- Attention sous Windows : comme en Java l'antislash est un marqueur de caractère spécial : `'\n'`, `'\t'`, etc.
 - utiliser le double antislash comme caractère de séparation
C:\\Users\\Anna\\workspace\\progIHM\\TP1...
- Pas de problème sous Unix ou MacOS :
 - `monLabel.setIcon(new ImageIcon("src/gif/image1.gif"));`

Ajout d'une image en icône

- Pour affecter une image comme icône de l'application **SOUS WINDOWS**, 3 méthodes sont utilisées:



- En utilisant la classe `Toolkit` qui contient la méthode `getDefaultToolkit()`. L'objet `Toolkit` contient la méthode `getImage()` qui prend l'URL de l'image et rend un objet de type `Image`.

```
//Affecter une image comme icône de l'application SOUS WINDOWS
Toolkit tk=Toolkit.getDefaultToolkit();
Image im = tk.getImage("src/gif/image.gif");
setIconImage(im);
```

- En utilisant `ImageIcon`. La méthode `getImage()` retourne une instance image de `ImageIcon`

```
setIconImage(new ImageIcon("Java_logo.png").getImage());
```

- En utilisant `ImageIO`. La méthode `read()` de la classe `ImageIO` prend un objet `InputStream` qui pointe vers le fichier de l'image.

```
try{
    setIconImage(ImageIO.read(new FileInputStream("Java_logo.png")));
}
catch(Exception e){}
```

Taille d'une fenêtre, d'un composant ou d'un panel

- Mettre une fenêtre au centre

```
JFrame frame=new JFrame();  
frame.setLocationRelativeTo(null);
```

- Mettre une fenêtre sur tout l'écran

```
GraphicsEnvironment gEnv = GraphicsEnvironment.getLocalGraphicsEnvironment();  
Rectangle bounds = gEnv.getMaximumWindowBounds();  
frame.setBounds(bounds);  
frame.setVisible(true);
```

- Pour redimensionner un panel ou un composant, on utilise la méthode `setPreferredSize()`

```
lbl_image.setPreferredSize(new Dimension(500,100));
```

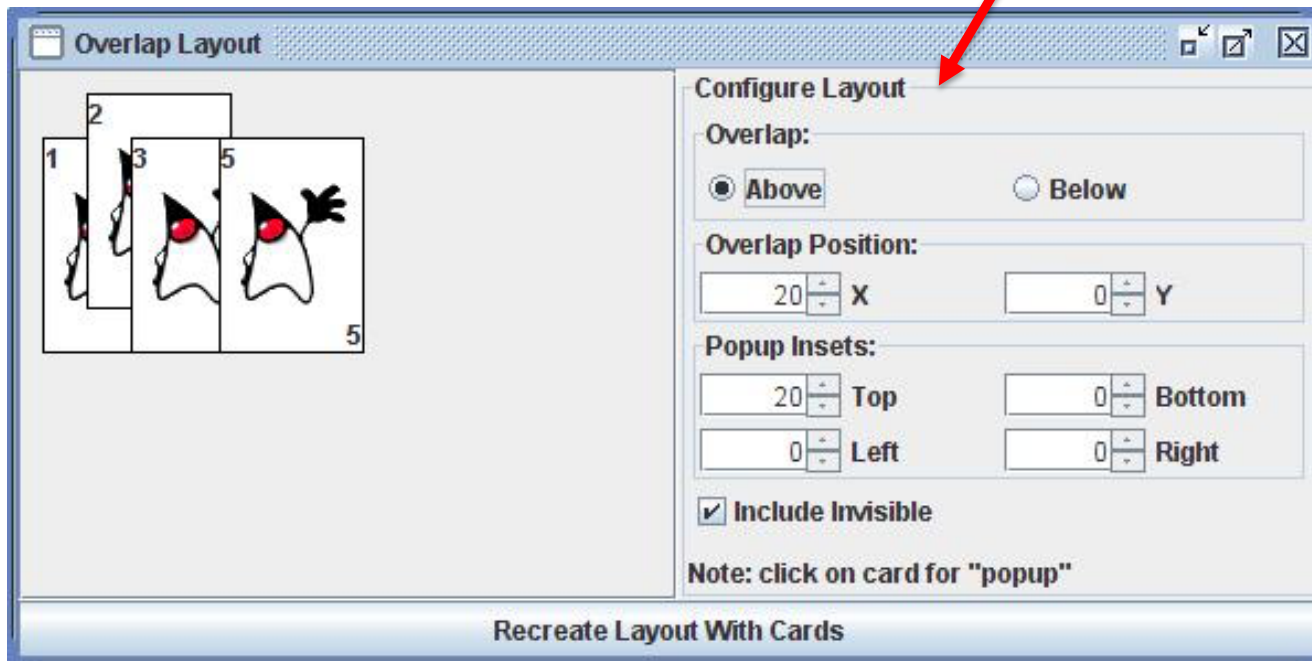
```
JPanel pan=(JPanel) getContentPane();  
pan.setPreferredSize(new Dimension(200,300));
```

Bordure et titre d'un JPanel

- Pour ajouter une bordure à un panneau, il faut utiliser la méthode `setBorder()`

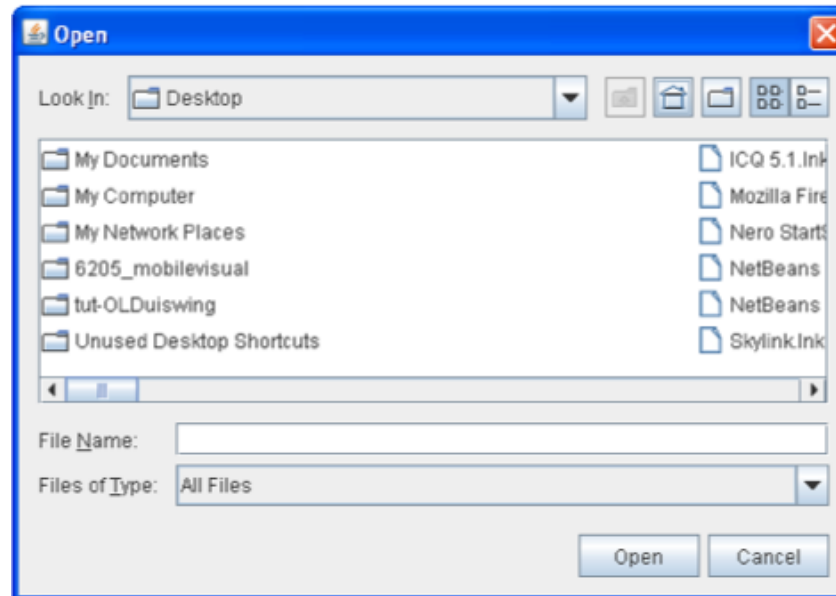
```
JPanel pan=(JPanel) getContentPane();  
pan.setBorder(BorderFactory.createTitledBorder("Configure Layout"));
```

- Un exemple avec cette fenêtre :



Fenêtre de sélection de fichiers

- La **FileDialog** d'AWT : fenêtre de base permettant d'ouvrir ou d'enregistrer un fichier
- Le **JFileChooser** de SWING : fenêtre plus élaborée avec notamment la possibilité de filtrer les fichiers



Fenêtre de sélection de fichiers: FileDialog

Choix du fichier pour ouvrir

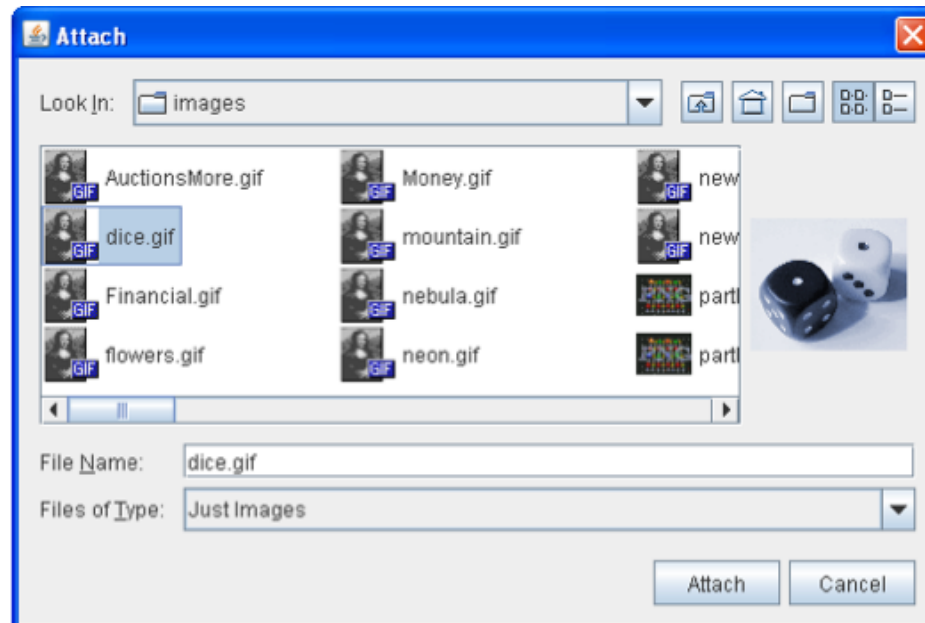
```
String nomFic = new String("");
try {
    // ouvrir un fichier
    FileDialog fd = new FileDialog(this, "Sélectionnez votre fichier...", FileDialog.LOAD);
    fd.setVisible(true);
    nomFic = ((fd.getDirectory()).concat(fd.getFile()));
}
catch (NullPointerException e) {
    System.out.println("Erreur ouverture dossier !");
}
```

Choix du fichier pour enregistrer

```
String nomFic = new String("");
try {
    FileDialog fd = new FileDialog(this, "Sélectionnez votre fichier...", FileDialog.SAVE);
    fd.setVisible(true);
    nomFic = ((fd.getDirectory()).concat(fd.getFile()));
}
catch(NullPointerException e) {
    System.out.println("Erreur ouverture dossier !");
}
```

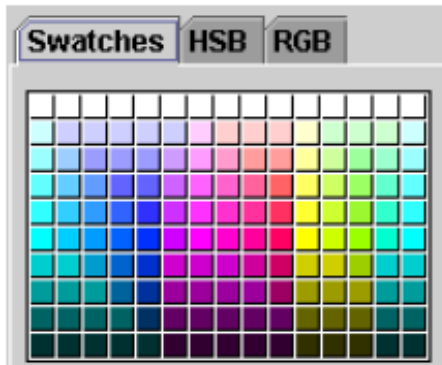

Fenêtre de sélection de fichiers: JFileChooser

- Beaucoup plus complet
- Fenêtre de dialogue permettant de choisir un fichier
- Regarder le tutoriel sur le site de SUN

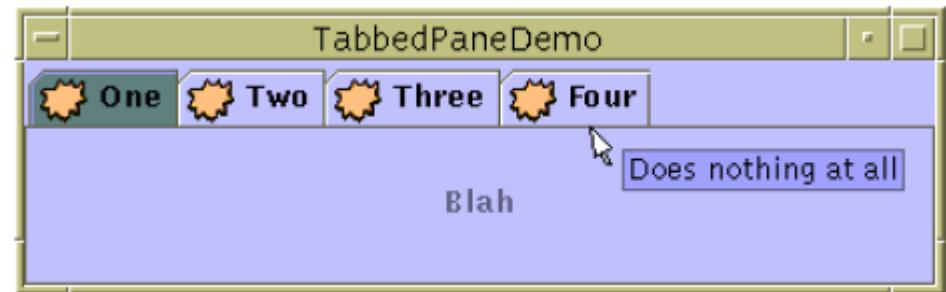


Autres composants utiles

JColorChooser : fenêtre de dialogue permettant de choisir une couleur



JTabbedPane : permet de mettre plusieurs JPanel dans des onglets



JSplitPane : il s'agit d'un double conteneur permettant une interaction entre deux composants



JScrollPane : un conteneur permettant le défilement (ascenseur) si nécessaire

