

# Interface Graphique en Java

## Introduction à Swing

Ilhem BOUSSAID

USTHB

5 octobre 2009

# Plan

- 1 Positionnement des composants
  - BorderLayout
  - FlowLayout
  - GridLayout

# Les layout manager

- Pour placer des composants dans un **container**, Java propose une technique de **Layout**.
- Un layout est une entité Java qui place les composants les uns par rapport aux autres.
- Le layout s'occupe aussi de réorganiser les composants lorsque la taille du container varie.
- Il y a plusieurs layout : **BorderLayout**, **BoxLayout**, **CardLayout**, **FlowLayout**, **GridLayout**, **GridBagLayout**.
- Un layout n'est pas contenu dans un container, il gère le **positionnement**.
- Création de Layout
  - Il est possible de construire son propre Layout
  - Un layout doit implanter l'interface **java.awt.LayoutManager** ou **java.awt.LayoutManager2**

# Les layout manager

- Certains composants ont des gestionnaires de mise en page par défaut :
  - **JPanel** : FlowLayout
  - **contentPane d'une JFrame** : BorderLayout
- S'il ne convient pas, on peut en changer en adressant au composant la méthode : **setLayout**
- **Exemple** : changer le layout du contentPane d'une JFrame :

```
JFrame fen=new JFrame("changement_de_mise_en_page");
Container c=fen.getContentPane();
FlowLayout flow = new FlowLayout();
c.setLayout(flow);
```

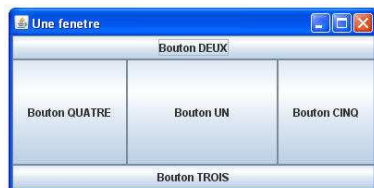
# BorderLayout

- Le **BorderLayout** sépare un container en cinq zones : Nord, Sud, Est et Ouest
- Lorsque l'on agrandit le container, le centre s'agrandit. Les autres zones prennent uniquement l'espace qui leur est nécessaire.
- Les composants y sont ajoutés par la méthode **add(Component, int)** dont le second paramètre indique dans quelle zone doit être placé le composant . Ce paramètre peut prendre les valeurs : **BorderLayout.NORTH**, **BorderLayout.SOUTH**, **BorderLayout.EAST**, **BorderLayout.WEST** ou **BorderLayout.CENTER**

# Exemple

```
import java.awt.* ;
import javax.swing.* ;
class FrameDemo5 extends JFrame {
public FrameDemo5 () {
super("Une_fenetre") ;
setSize(400, 200) ;
Container contenu = this.getContentPane() ;
contenu.setLayout(new BorderLayout()) ;
contenu.add(new JButton("Bouton_UN")) ;
//bouton placé au centre par défaut
contenu.add(new JButton("Bouton_DEUX"), BorderLayout.NORTH) ;
contenu.add(new JButton("Bouton_TROIS"), BorderLayout.SOUTH) ;
contenu.add(new JButton("Bouton_QUATRE"), BorderLayout.WEST) ;
contenu.add(new JButton("Bouton_CINQ"), BorderLayout.EAST) ;
}
public static void main(String args[]) {
JFrame fen = new FrameDemo5() ;
fen.setVisible(true) ;
}
}
```

# Illustration



- Le gestionnaire de mise en forme BorderLayout ne tient pas compte de la taille souhaitée des composants, qui peut être imposée par la méthode `setPreferredSize` de la classe `JComponent`

# FlowLayout

- Un **FlowLayout** permet de ranger les composants dans une ligne. Si l'espace est trop petit, une autre ligne est créée.
- Le **FlowLayout** est le layout par défaut des **JPanel**
- son constructeur accepte trois paramètres selon le modèle suivant **FlowLayout(int,int,int)** :
  - Le premier paramètre indique comment seront alignés les composants, il peut prendre les valeurs suivantes : **FlowLayout.CENTER**, **FlowLayout.LEFT** ou **FlowLayout.RIGHT**.
  - Le deuxième paramètre donne l'espacement horizontal entre composants (en pixels).
  - Le dernier paramètre donne l'espacement vertical entre composants (en pixels).



# Exemple

Exemple :

```
Container contenu = this.getContentPane();
contenu.setLayout(new FlowLayout());
contenu.add(new JButton("Button_1"));
contenu.add(new JButton("2"));
contenu.add(new JButton("Button_3"));
contenu.add(new JButton("Long-Named_Button_4"));
contenu.add(new JButton("Button_5"));
```



# GridLayout

- Un **GridLayout** permet de positionner les composants sur une grille.
- Son constructeur **GridLayout(int,int,int,int)** accepte quatre paramètres :
  - Le premier paramètre est le nombre de lignes du tableau
  - le deuxième paramètre est le nombre de colonnes du tableau
  - Le troisième paramètre donne l'espacement horizontal entre composants (en pixel)
  - Le dernier paramètre donne l'espacement vertical entre composants (en pixels).
- Toutes les cases du tableau ont la même taille (celle déterminée par le composant le plus grand).
- Les composants seront ajoutés par la suite à l'aide de la méthode **add(Component)** qui remplira le tableau ligne par ligne.

# Exemple

```
Container contenu = this.getContentPane();
contenu.setLayout(new GridLayout(0,2));
contenu.add(new JButton("Button_1"));
contenu.add(new JButton("2"));
contenu.add(new JButton("Button_3"));
contenu.add(new JButton("Long-Named_Button_4"));
contenu.add(new JButton("Button_5"));
```

