



## Quelques classes d'agencement

- ◆ BorderLayout
- ◆ BoxLayout
- ◆ FlowLayout
- ◆ GridLayout
- ◆ GroupLayout

et un guide à l'adresse

<http://docs.oracle.com/javase/tutorial/uiswing/layout/index.html>

## Utilisation d'un agencement

- ◆ Le conteneur est instancié avec passage de l'agencement choisi en paramètre du constructeur

```
p = new JPanel (new BorderLayout ());
```

- ◆ Le conteneur est instancié sans passage de paramètre au constructeur. L'agencement est fourni par un appel à **setLayout**

```
p = new JPanel ();  
p.setLayout (new BorderLayout ());
```

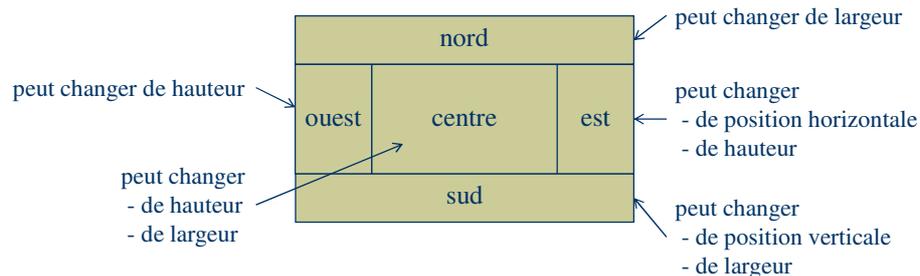
- ◆ Les composants sont ajoutés dans le conteneur par un appel à **add**

```
p.add (composant, contrainte);
```

← dépend de l'agencement

## Agenceur BorderLayout

Un agenceur qui maintient  
les composants périphériques près d'un bord  
le composant central au milieu



## Contraintes pour BorderLayout

- ◆ Espacement entre les composants fourni en arguments du constructeur  
`BorderLayout(int hgap, int vgap)`
- ◆ Position du composant, indiquée à l'ajout dans le conteneur par l'une des contraintes :  
`NORTH, SOUTH`  
`WEST, EAST`  
`CENTER`  
...

## Agenceur FlowLayout

Un agenceur qui maintient (dans la limite de la largeur du conteneur) les composants les uns à côté des autres.

S'il y a trop de composants pour la largeur du conteneur, les composants sont distribués sur autant de « lignes » que nécessaire.

## Contraintes pour FlowLayout

Alignement et espacement entre les composants peuvent être fournis en paramètres du constructeur

```
FlowLayout(int align, int hgap, int vgap)
```

← LEFT, CENTER, RIGH  
LEADING, TRAILING

Type de disposition des composants indiqué par un appel à `setComponentOrientation` qui peut prendre en argument :

```
ComponentOrientation.LEFT_TO_RIGHT
```

```
ComponentOrientation.RIGHT_TO_LEFT
```

## ActionListener

- ◆ L'interface des écouteurs de déclenchement d'action.
- ◆ Lorsqu'un événement de type `ActionEvent` survient, il est transmis à la méthode `void actionPerformed(ActionEvent e)` qui doit être définie.

## ActionEvent

- ◆ La classe de description des événements d'action.
- ◆ `int getModifiers()`  
retourne les touches modificatrices (Alt, Ctrl ...) maintenues enfoncées pendant cet événement.
- ◆ `long getWhen()`  
retourne une horodatation de l'instant où l'événement est survenu.
- ◆ `Object getSource()`  
retourne l'object initiateur de l'événement (un bouton, un item de menu, ...)