Architecture de Layout

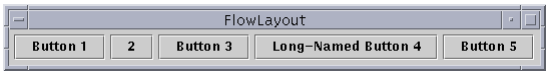
Architecture de Layout

- 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 est un objet.

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



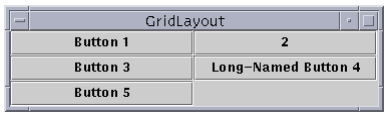
Ex :

```
Container contentPane = getContentPane();
contentPane.setLayout(new FlowLayout());

contentPane.add(new JButton("Button 1"));
contentPane.add(new JButton("2"));
contentPane.add(new JButton("Button 3"));
contentPane.add(new JButton("Long-Named Button 4"));
contentPane.add(new JButton("Button 5"));
```

GridLayout

Un GridLayout permet de positionner les composants sur une grille.



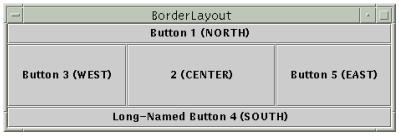
Ex:

```
Container contentPane = getContentPane();
contentPane.setLayout(new GridLayout(0,2));
contentPane.add(new JButton("Button 1"));
contentPane.add(new JButton("2"));
contentPane.add(new JButton("Button 3"));
contentPane.add(new JButton("Long-Named Button 4"));
contentPane.add(new JButton("Button 5"));
```

BorderLayout

Le BorderLayout sépare un container en cinq zones: NORTH, SOUTH, EAST, WEST et CENTER

Lorsque l'on agrandit le container, le centre s'agrandit. Les autres zone prennent uniquement l'espace qui leur est nécessaire.



Ex :

```
Container contentPane = getContentPane();
contentPane.setLayout(new BorderLayout());
contentPane.add(new JButton("Button 1 (NORTH)"), BorderLayout.NORTH);
```

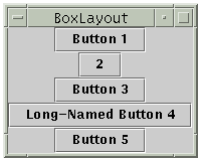
BoxLayout

Un BoxLayout permet d'empiler les composants du container (soit de verticalement, soit horizontalement)

Ce layout essaye de donner à chaque composant la place qu'il demande

Il est possible de rajouter des blocs invisible.

Il est possible de spécifier l'alignement des composants (centre, gauche, droite)



CardLayout

Un CardLayout permet d'avoir plusieurs conteneurs ; les uns au dessus des autres (comme un jeux de cartes).

Ex :

```
JPanel cards;
final static String BUTTONPANEL = "JPanel with JButtons";
final static String TEXTPANEL = "JPanel with JTextField";
cards = new JPanel();
cards.setLayout(new CardLayout());
cards.add(p1,BUTTONPANEL);
cards.add(p2,TEXTPANEL);
```

GridBagLayout

Le GridBagLayout est le layout le plus complexe. Il place les composants sur une grille, mais des composants peuvent être contenus dans plusieurs cases.
Pour exprimer les propriétés des composants dans la grille, on utilise un GridBagConstraints.

Un GridBasConstraints possède :

- gridx, gridy pour spécifier la position
- gridwidth, gridheight pour spécifier la place
- fill pour savoir comment se fait le remplissage
- …

Ex :

```
GridBagLayout gridbag = new GridBagLayout();
GridBagConstraints c = new GridBagConstraints();
JPanel pane = new JPanel();
pane.setLayout(gridbag);
gridbag.setConstraints(theComponent, c);
pane.add(theComponent);
```

Création de Layout

- Il est possible de construire son propre Layout
- Un layout doit implanter l'interface java.awt.LayoutManager ou java.awt.LayoutManager2

END