Programmation orientée objet en langage JAVA

Chapitre 6 : Les gestionnaires d'agencement

Claude Duvallet

Université du Havre
UFR Sciences et Techniques
25 rue Philippe Lebon - BP 540
76058 LE HAVRE CEDEX
Claude.Duvallet@gmail.com
http://litis.univ-lehavre.fr/~duvallet/

Les gestionnaires d'agencement

Introduction (1/2)

- Un gestionnaire de mise en page est une stratégie pour placer les composants dans une fenêtre ou un autre composant (souvent un panneau).
- Dans Java, le contenu de la fenêtre ou tout composant graphique est un Container.
- Un gestionnaire de mise en page est choisi en appelant la méthode `setLayout` du container.
- Les mises en pages sont utilisées pour atteindre un certain degré d'indépendance de platforme et une certaine évolutivité.

Introduction (2/2)

- AWT/Swing supportent plusieurs gestionnaires de mises en pages. En voici quatre :
  - `FlowLayout`
  - `GridLayout`
  - `BorderLayout`
  - `BoxLayout`
- Ces classes implémentent l'interface `java.awt.LayoutManager`.

Les gestionnaires de flux (FlowLayout)

- Place les composants dans une ligne aussi longtemps qu'elle les contient, puis entame une nouvelle ligne.
- Espace correctement les composants, les centre par défaut.
- Laisse chaque composant choisir sa taille naturelle.
- Souvent utilisée pour placer les boutons sur les panneaux.
- Exemple :

```
Container c = getContentPane();
c.setLayout(new FlowLayout());
c.add (new JButton ("Back to Start"));
c.add (new JButton ("Previous Slide"));
c.add (new JButton ("Next Slide"));
c.add (new JButton ("Last Slide"));
c.add (new JButton ("Exit"));
```

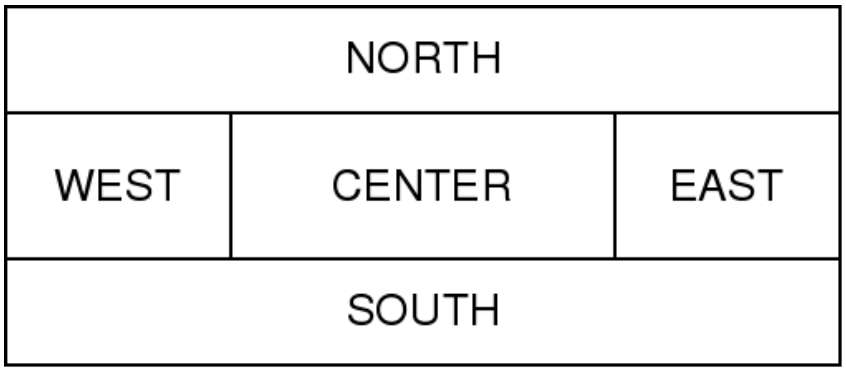
Mise en page en grilles (GridLayout)

- Divise le panneau en une grille avec un nombre donné de lignes et de colonnes.
- Place les composants dans les cellules de la grille.
- Force la taille de chaque composant à occuper toute la cellule.
- Permet des espaces additionnels entre les cellules.
- Exemple :

```
Container c = getContentPane();
c.setLayout (new GridLayout(3, 2,  10, 20 ));
c.add (new JButton ("Back to Start"));
c.add (new JButton ("Previous Slide"));
c.add (new JButton ("Next Slide"));
c.add (new JButton ("Last Slide"));
c.add (new JButton ("Exit"));
```

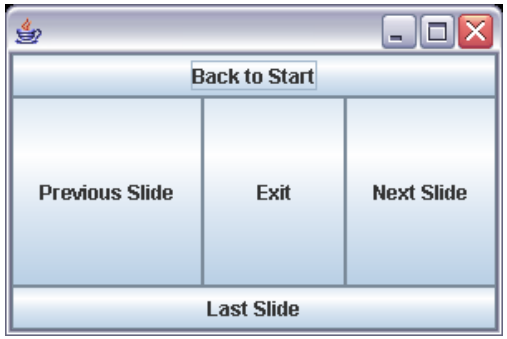
BorderLayout

- Divise l'espace en 5 régions et ajoute un composant à la région spécifiée.



- Force la taille de chaque composant à occuper toute la région.
- Exemple :

```
Container c = getContentPane();
c.setLayout(new BorderLayout());  // optionnel : défaut
c.add (new JButton ("Next Slide"), BorderLayout.EAST);
c.add (new JButton ("Previous Slide"), BorderLayout.WEST);
c.add (new JButton ("Back to Start"), BorderLayout.NORTH);
c.add (new JButton ("Last Slide"), BorderLayout.SOUTH);
c.add (new JButton ("Exit"), BorderLayout.CENTER);
```



Les boîtes (1/2)

- Dans une boîte horizontale, les composants placés horizontallement, de la gauche vers la droite.

- Dans une boîte verticale, les composants sont placés verticallement, du haut vers le bas.

- BoxLayout est la mise en page par défaut de la boîte Box container.
- Le travail avec les boîtes est légèrement différent :

```
Box boite1 = Box.createHorizontalBox();
boite1.add (...);

// ajoute un espace, 60 pixels:
boite1.add(Box.createHorizontalStrut (60));

Box boite2 = Box.createVerticalBox();
...
```

Les boîtes (2/2)

- Exemple :

```
Container c = getContentPane();
c.setLayout(new FlowLayout());
Box boite = Box.createVerticalBox();
boite.add (new JButton ("Next Slide"));
boite.add (new JButton ("Previous Slide"));
boite.add (Box.createVerticalStrut (20) );
boite.add (new JButton ("Exit"));
c.add (boite);
```

Mise en page par défaut

- Chaque composant possède un gestionnaire de mise en page par défaut, qui garde son effet jusqu'à ce que la méthode setLayout du composant soit appelée.
- Les gestionnaires de mise en page par défaut sont :
  - Content pane $\longleftrightarrow$ BorderLayout
  - JPanel $\longleftrightarrow$ FlowLayout
  - Box $\longleftrightarrow$ BoxLayout