



Bài 3: Layout Manager

Mục tiêu bài học

Giới thiệu về quản lý bố cục

- FlowLayout
- BorderLayout
- GridLayout
- CardLayout
- BoxLayout
- GridBagLayout
- SpringLayout
- GroupLayout
- TabbedPaneLayout
- SplitPaneLayout

Layout Manager

- Một Container là một Component có thể chứa các Component khác
- JFrame, JDialog, JScrollPane, JPanel, JDesktopPane, JInternalFrame
- getContentPane().add để thêm Component vào Container
- Mỗi Container có một đối tượng Layout Manager
- Layout Manager là một đối tượng quyết định cách sắp xếp vị trí của các Component bên trong một Container.
- Các Layout Manager “implements” từ interface LayoutManager hoặc LayoutManger2.

FlowLayout



FlowLayout

- Flow Layout bố trí các Component trong Container theo dòng, từ trái sang phải theo thứ tự thêm vào.
- Tạo dòng mới khi kích thước dòng còn lại không đủ chứa Component thêm vào.
- Flow Layout bố trí vị trí các Component phụ thuộc vào kích thước của Container.
- Mỗi dòng của các Component được window mặc định canh giữa theo chiều ngang . Có thể điều chỉnh canh trái hoặc phải



FlowLayout

■ Khởi tạo

- `public FlowLayout ()`
 - `align: FlowLayout.CENTER`
 - `vgap: 5px, hgap: 5px`
- `FlowLayout (int align)`
 - `align`: canh lề
 - `FlowLayout.CENTER` : Canh giữa
 - `FlowLayout.LEFT`; : Canh trái
 - `FlowLayout.RIGHT`; : Canh phải
- `FlowLayout(int align, int vgap, int hgap)`
 - `align` : canh lề
 - `vgap` : kích thước chiều ngang
 - `hgap`: chiều dọc

FlowLayout

- Phương thức
- `public void setAlignment(int align)`
- `public void setHgap(int hgap)`
- `public void setVgap (int vgap)`
- `public int getAlignment()`
- `public int getHgap ()`
- `public int getVgap ()`

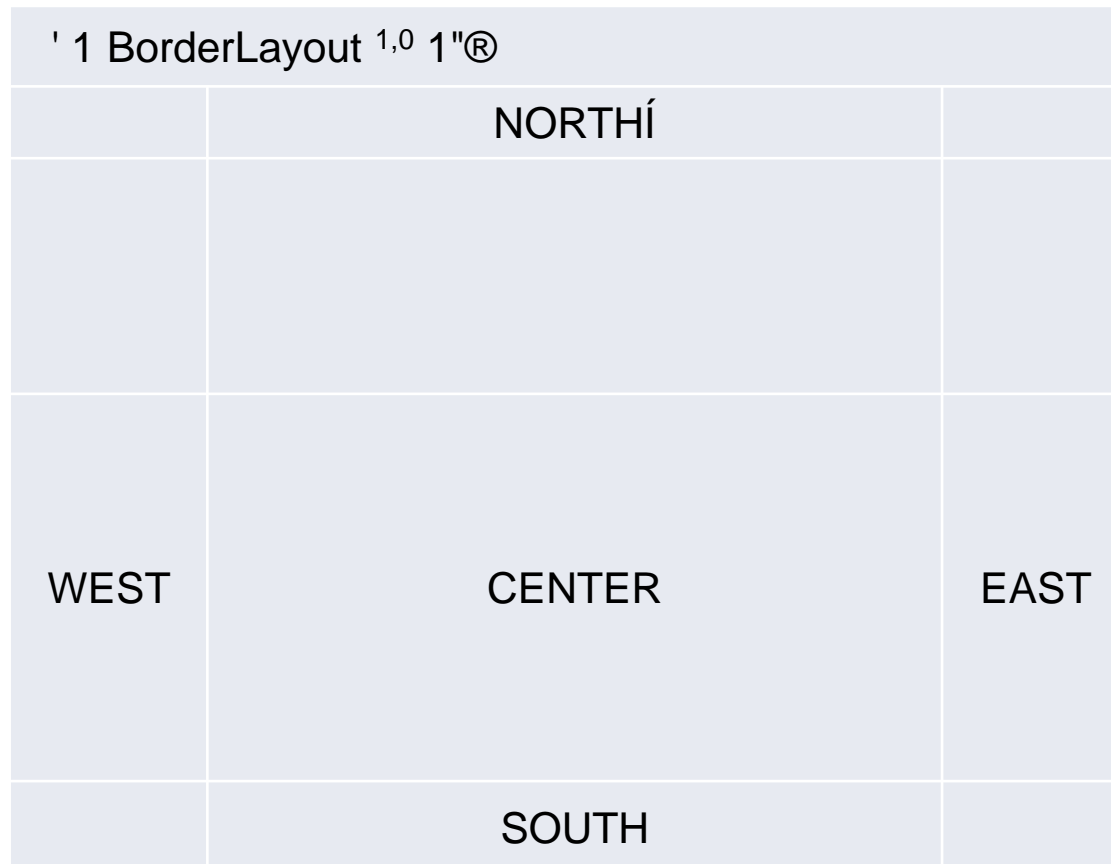
FlowLayout

- Ví dụ:

```
1 FlowLayout layout=new FlowLayout();  
2 layout.setAlignment(FlowLayout.LEFT);  
3 this.setLayout(layout);  
4
```


BorderLayout

- BorderLayout bố trí các Component bên trong Container theo 5 vùng:
- "North", "South", "East", "West", "Center".



BorderLayout

- khởi tạo:
 - `public BorderLayout ()`
 - `hgap = 0`
 - `vgap = 0`
 - `public BorderLayout (int hgap, int vgap)`
 - `hgap`: chiều ngang
 - `vgap`: chiều dọc



BorderLayout

```
4 private JButton jbtNorth; private JButton jbtSouth;
5 private JButton jbtEast; private JButton jbtWest;
6 private JButton jbtCenter;
7 //Phương thức khởi tạo public BorderLayoutFrame() {
8 this.initComponents();
9 //Phương thức khởi tạo các thành phần
10 private void initComponents() {
11 //MainFrame
12 this.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
13 this.setTitle("BorderLayout");
14 BorderLayout layout = new BorderLayout();
15 this.setLayout(layout);
16 //jbtNorth
17 this.jbtNorth = new JButton("NORTH");
18 this.getContentPane().add(this.jbtNorth, BorderLayout.NORTH);
19 //jbtWest
20 this.jbtWest = new JButton("WEST");
21 this.getContentPane().add(this.jbtWest, BorderLayout.WEST);
22 //jbtEast
23 this.jbtEast = new JButton("EAST");
24 this.getContentPane().add(this.jbtEast, BorderLayout.EAST);
25 //jbtSouth
26 this.jbtSouth = new JButton("SOUTH");
27 this.getContentPane().add(this.jbtSouth, BorderLayout.SOUTH);
28 //jbtCenter
29 this.jbtCenter = new JButton("CENTER");
30 this.getContentPane().add(this.jbtCenter, BorderLayout.CENTER);
31 pack();
32 }
```

CardLayout

- Card Layout quản lý nhiều Card cùng một không gian hiển thị
- Card Layout giúp quản lý hai hay nhiều Component (thường là JPanel) để chia sẻ cùng một không gian hiển thị.
- Chỉ duy nhất Top Card được hiển thị.
- Mỗi "Card" có thể sử dụng Layout Manager riêng.
- Card nào cũng có thể là Top Card
- Có thể sử dụng **JTabbedPane** để thay cho **Card Layout**

CardLayout

- khởi tạo :

- `public CardLayout ()`
 - `hgap = 0`
 - `vgap = 0`
- `public CardLayout (int hgap, int vgap)`
 - `hgap`: chiều ngang
 - `vgap` : chiều dọc

CardLayout

■ Phương thức

- `public void setHgap(int hgap)`
- `public void setVgap(int vgap)`
- `public int getHgap()`
- `public int getVgap()`
- `public void next (Container parent)`
- `public void previous(Container parent)`
- `public void first(Container parent)`
- `public void last(Container parent)`
- `public void show(Container parent, String name)`

■

CardLayout



GridLayout

- Grid Layout bố trí các Component của Container vào bên trong một Grid với các Row và Column.
- Mỗi Component được bố trí trong 1 Cell của lưới.
- Tất cả các Cell có cùng kích thước bên trong Grid.
- Các Component được thêm vào Container theo thứ tự từ trái sang phải, từ trên xuống dưới (mặc định).
- Kích thước của mỗi Cell được xác định bởi kích thước của Container.



GridLayout

- Khởi tạo
 - GridLayout()
 - rows: 1
 - -1 Component / 1 Column
 - GridLayout(int rows, int cols)
 - GridLayout(int rows, int cols, int hgap, int vgap)

GridLayout

- GridBag Layout bố trí các Component trong một Grid với các Row và Column.
- Mỗi Component bên trong Grid được RowSpan và ColumnSpan (giống table HTML)
- Width và Height của các Row/Column có thể khác nhau.
- GridBag Layout là một Layout Manager rất linh động cho việc bố trí các Component bên trong Container theo dạng Grid.
- GridBag Layout là một trong các Layout Manager **thường sử dụng nhất** mà Java Platform cung cấp.

GridLayout

- Các thuộc tính của GridBagConstraints
 - gridx ,gridy : vị trí dòng, vị trí cột
 - gridheight , gridwidth : số lượng dòng, số lượng cột
 - ipadx, ipady :
 - Insets
 - weightx, weighty
 - fill :
 - NONE, HORIZONTAL, VERTICAL, BOTH

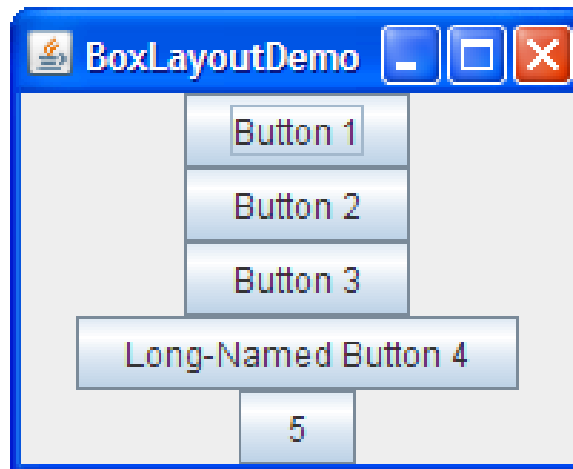
GridLayout

- Các thuộc tính của GridBagConstraints
 - anchor:

```
-----  
| FIRST_LINE_START    PAGE_START    FIRST_LINE_END |  
|  
| LINE_START          CENTER         LINE_END |  
|  
| LAST_LINE_START     PAGE_END       LAST_LINE_END |  
-----
```

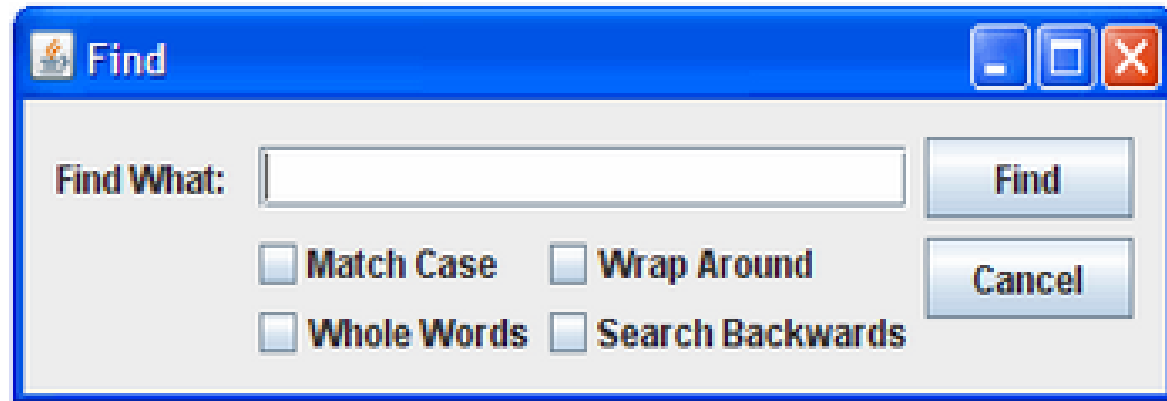
BoxLayout

- Box Layout bố trí các Component bên trong Container theo 1 dòng theo trục X, hoặc là trục Y.
- `BoxLayout(Container container, int align)`
- `container`: chứa các Component
- `axis`:
- `BoxLayout.X_AXIS` : Trục X
- `BoxLayout.Y_AXIS` : Trục Y



GroupLayout

- GroupLayout bố trí các Component bên trong Container theo chiều ngang và chiều dọc.
- Sự bố trí được thực hiện theo mỗi chiều riêng lẻ
-



XIN CẢM ƠN!

