

Layouts Java

Eduard Lara

- ☐ Los LayoutManagers se utilizan para organizar los componentes de una manera particular.
- ☐ LayoutManager es una interfaz implementada por todas las clases de gestores de diseño.
- ☐ Un LayoutManager es un objeto que controla el tamaño y la posición de los componentes en un contenedor.
- ☐ Cada objeto Container tiene un objeto LayoutManager que controla su diseño.

Clases que representan layoutManagers:

1. `java.awt.BorderLayout`
2. `java.awt.GridLayout`
3. `java.awt.GridBagLayout`
4. `java.awt.FlowLayout`
5. `java.awt.CardLayout`
6. `javax.swing.BoxLayout`

1. BORDERLAYOUT

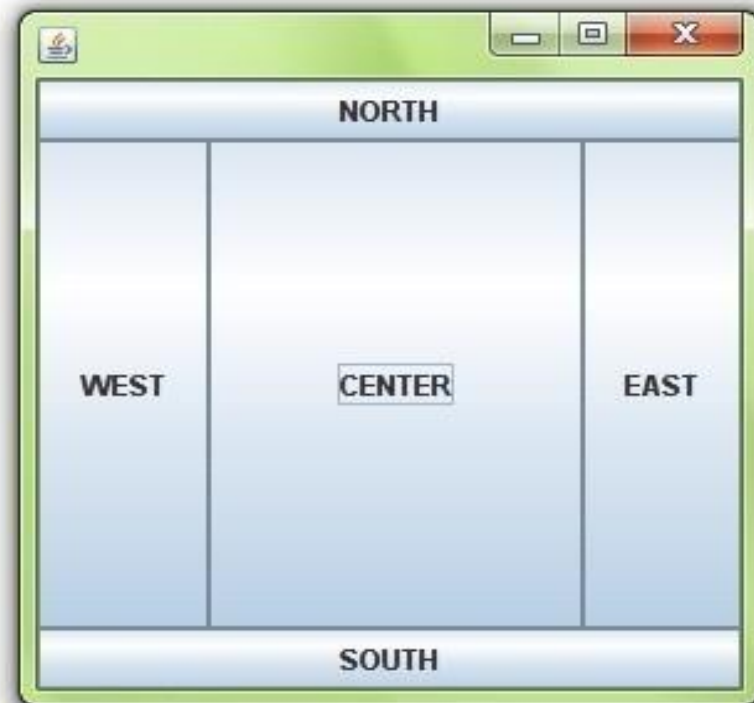
- ❖ The BorderLayout se utiliza para organizar los componentes en 5 regiones: norte, sur, este, oeste y center.
- ❖ Cada region (area) puede contener un componente sólo. Es el default layout de un frame o ventana
- ❖ El BorderLayout proporciona 5 constantes para cada region:
 - ❖ **public static final int NORTH**
 - ❖ **public static final int SOUTH**
 - ❖ **public static final int EAST**
 - ❖ **public static final int WEST**
 - ❖ **public static final int CENTER**

1. BORDERLAYOUT

Constructores de la clase BorderLayout:

BorderLayout(): crea a borderlayout pero sin separaciones entre los componentes.

BorderLayout(int hgap, int vgap): crea un borderlayout con separaciones horizontal y vertical dades entre los componentes.



2. GRIDLAYOUT

- ❑ The GridLayout se utiliza para organizar los componentes en un grid rectangular.
- ❑ Uno de los componentes es mostrado en cada rectángulo.
- ❑ Constructores de la clase GridLayout:
 - 1.GridLayout():** Crea un gridlayout con una columna por componente en una fila
 - 2.GridLayout(int rows, int columns):** Crea un gridlayout con las filas y columnas dadas sin brechas entre los componentes.
 - 3.GridLayout(int rows, int columns, int hgap, int vgap):** Crea un gridlayout con las filas y columnas dadas junto con las brechas horizontales y verticales dadas.

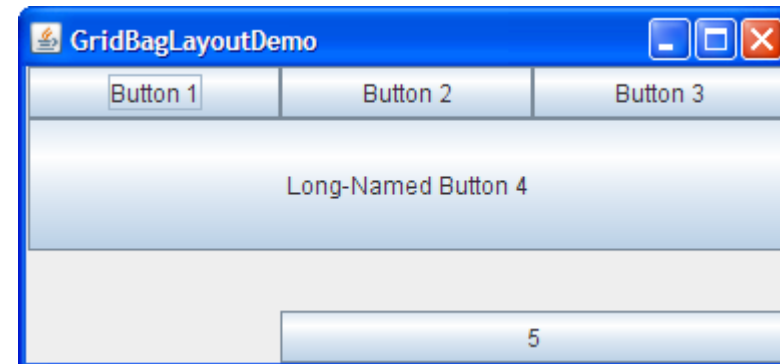
2. GRIDLAYOUT

❑ new GridLayout(3,3)



3. GRIDBAGLAYOUT

- ❑ Es como un GridLayout, pero en lugar de ubicar cada componente dentro de una celda, permite ubicarlos dentro de diferentes celdas combinadas



4. FLOWLAYOUT

- ❑ FlowLayout es usado para organizar los componentes en una linea, uno después de otro (en un flujo).
- ❑ Es el layout por defecto de un applet o panel.
- ❑ Campos de clase FlowLayout:

1. **public static final int LEFT**
2. **public static final int RIGHT**
3. **public static final int CENTER**
4. **public static final int LEADING**
5. **public static final int TRAILING**

4. FLOWLAYOUT

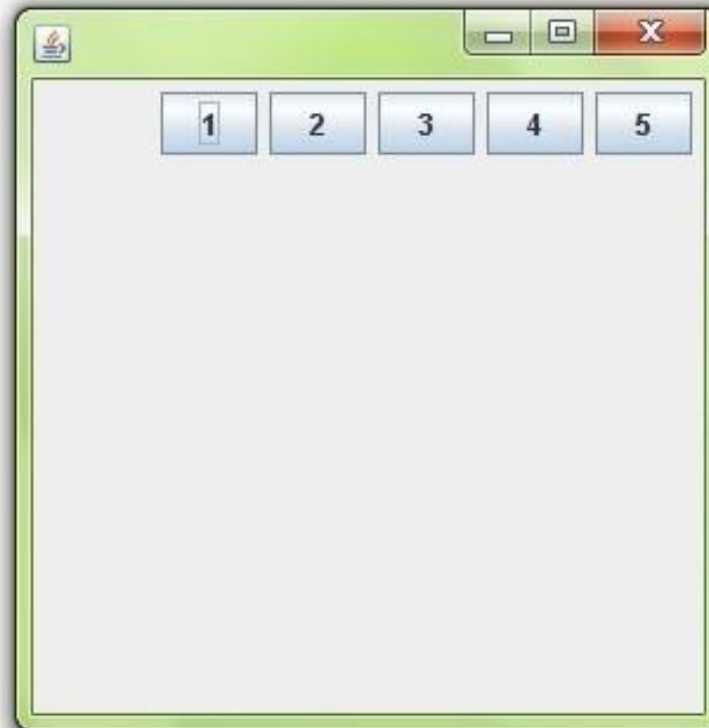
Constructores de la clase FlowLayout:

1.FlowLayout(): Crea un Flowlayout con alineación centrada y una separación horizontal y vertical por defecto de 5 unidades.

2.FlowLayout(int align): Crea un flowlayout con la alineación dada y una separación horizontal y vertical por defecto de 5 unidades.

3.FlowLayout(int align, int hgap, int vgap): Crea un flowlayout con la alineación dada y una separación horizontal y vertical dada.

4. FLOWLAYOUT



5. CARDLAYOUT

- ❑ La clase CardLayout gestiona los componentes de manera que solo un componente es visible al mismo tiempo.
- ❑ Trata cada componente como una carta (sólo muestra un componente), es por eso que es conocido como CardLayout.
- ❑ Constructores de la clase CardLayout

1. CardLayout(): crea a cardlayout sin brechas horizontales ni verticales.

2. CardLayout(int hgap, int vgap): Crea un cardlayout con la separación horizontal and vertical dada.

5. CARDLAYOUT

Métodos

- ☐ `public void next(Container parent)`: Es usado para pasar a la siguiente carta del container dado.
- ☐ `public void previous(Container parent)`: Es usado para flip to the previous card of the given container.
- ☐ `public void first(Container parent)`: Es usado para pasar ala primera carta del container dado.
- ☐ `public void last(Container parent)`: Es usado para pasar a la última carta del container dado.
- ☐ `public void show(Container parent, String name)`: Es usado para pasar a la carta específica con el nombre dado.

6. BOXLAYOUT

- ❑ Permite organizar los componentes según diferentes ejes:

- ❑ Campos de la clase BorderLayout

- ❑ `public static final int X_AXIS`

- ❑ `public static final int Y_AXIS`

- ❑ `public static final int LINE_AXIS`

- ❑ `public static final int PAGE_AXIS`

- ❑ Constructor de la clase BorderLayout:

- ❑ `BoxLayout(Container c, int axis)`: Crea un boxlayout que organiza los components en el eje indicado.

