

Universidad Autónoma de Tlaxcala



Introducción al lenguaje JAVA

Java

M.C. José Juan Hernández Mora

M.C. José Juan Hernández Mora

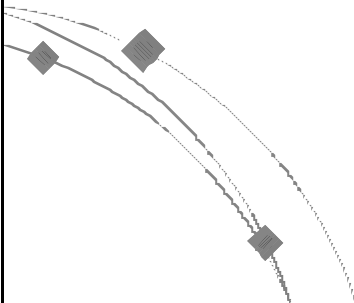
Cuarta sesión

1. Imágenes en Java
2. Animaciones sencillas con imágenes en Java
3. Interfaz de usuario en Java
 - Etiquetas
 - Botones
 - Casillas
 - Grupos de casillas
 - Listas
 - Menú
 - Áreas de Texto
 - Gestores de Organización
4. Entornos de desarrollo

M.C. José Juan Hernández Mora

Componente

- **Componente** es una clase abstracta que representa todo lo que tiene una posición, un tamaño, puede ser pintado en pantalla y puede recibir eventos.



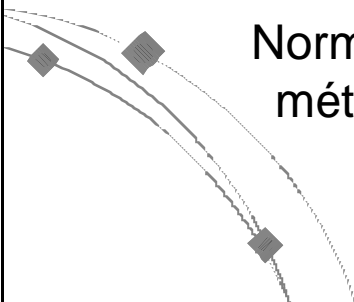
M.C. José Juan Hernández Mora

Adicionando componentes

- Para adicionar un componente usamos el método:

```
add(componente);
```

Normalmente dentro del método `init()`



M.C. José Juan Hernández Mora

Algunos de los Objetos derivados Abstract Window Toolkit :

- Label
- Button
- Checkbox
- Choice
- List
- TextField

M.C. José Juan Hernández Mora

Etiquetas

- Una etiqueta (***Label***) proporciona una forma de colocar texto estático en un panel, para mostrar información fija, que no varía (normalmente), al usuario.

Etiqueta

M.C. José Juan Hernández Mora

Constructores de Label

- Label()
- Label(String lbl)
- Label(String lbl, int align)

La clase ***Label*** define las constantes

- Label.CENTER
- Label.LEFT
- Label.RIGHT

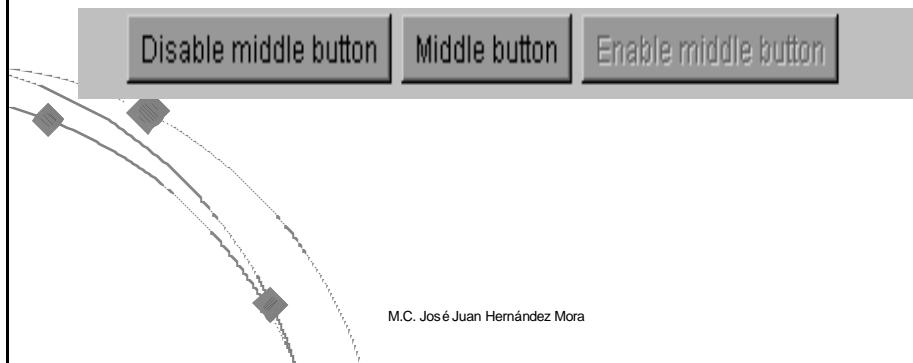
M.C. José Juan Hernández Mora

Establecer u obtener el texto del Label

- setText(String txt)
- Establece el texto de la etiqueta
- String getText()
- Obtiene el texto de la etiqueta

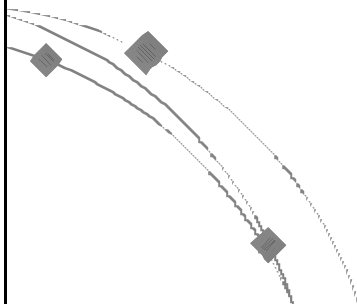
M.C. José Juan Hernández Mora

Botones



Botones de Pulsación

- Los botones de pulsación (***Button***), son un sencillo control que genera un evento `Action()` cuando el usuario lo pulsa.



Constructores

- Button(String label)
- Button()

Métodos

- setLabel(String str)
- ◆ ● Permite establecer la etiqueta del Button
- String getLabel()
- Permite obtener la etiqueta del Button

M.C. José Juan Hernández Mora

Ejemplo

- pregunta.java

M.C. José Juan Hernández Mora

Component Checkbox y clase CheckboxGroup

- | | |
|------------------------------------------------|----------------------------------|
| <input type="checkbox"/> Checkbox 1 | <input type="radio"/> Checkbox 4 |
| <input type="checkbox"/> Checkbox 2 | <input type="radio"/> Checkbox 5 |
| <input checked="" type="checkbox"/> Checkbox 3 | <input type="radio"/> Checkbox 6 |

M.C. José Juan Hernández Mora

Component Checkbox y clase CheckboxGroup

- Los objetos de la clase **Checkbox** son **botones de opción o de selección** con dos posibles valores:

true y false.

M.C. José Juan Hernández Mora

Constructores de Checkbox

- `Checkbox()`
 - Casilla sin etiqueta
- `Checkbox(String)`
 - Casilla con etiqueta
- `Checkbox(String, CheckboxGroup, boolean)`
 - Establece la etiqueta, si pertenece a un Grupo y si está o no seleccionado

M.C. José Juan Hernández Mora

Métodos

- `setLabel(String)`,
 - Establece la etiqueta del componente
- `String getLabel()`
 - Obtiene la etiqueta del componente
- `setState(boolean)`
 - Establece el estado (true o false, según esté seleccionado o no)
- `boolean getState()`
 - Obtiene el estado (true o false, según esté seleccionado o no)

M.C. José Juan Hernández Mora

Ejemplo

- pregunta2.java

M.C. José Juan Hernández Mora

Menus emergentes “*Choice*”



M.C. José Juan Hernández Mora

Menu emergente “*Choice*”

- La clase *Choice* proporciona una lista de opciones al estilo menú, a la que se accede por un botón distintivo. Cuando el usuario elige un ítem de la lista, la clase *Choice* genera un evento *Action*.

M.C. José Juan Hernández Mora

Constructor de Choice

- `Choice()`

Métodos

- `addItem(String)`
 - Añade un elemento a la lista
- `insert(String label, int index)`
 - Inserta un elemento con un label en la posición indicada

M.C. José Juan Hernández Mora

Choice

- `int getSelectedIndex()`
 - Obtiene el index del elemento
- `String getSelectedItem()`
 - Obtiene el label del elemento elegido de la lista
- `int getItemCount()`
 - Obtiene el número de elementos
- `String getItem(int)`
 - Obtiene el label a partir del index

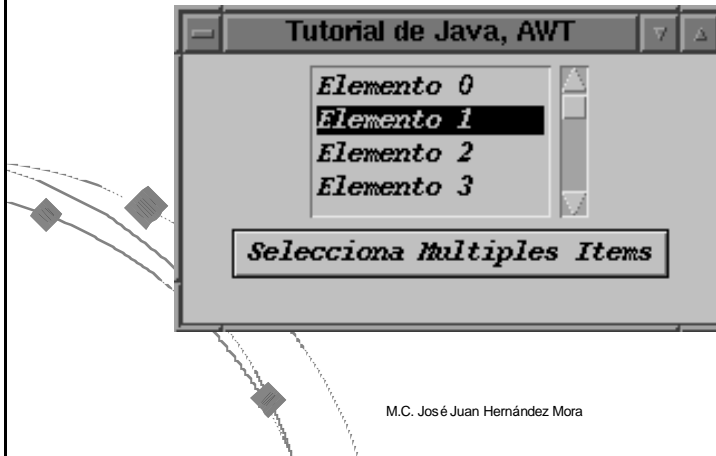
M.C. José Juan Hernández Mora

Ejemplo

- `pregunta3.java`

M.C. José Juan Hernández Mora

La clase *List*



M.C. José Juan Hernández Mora

La clase *List*

- La clase *List* viene definida por una zona de pantalla con varias líneas, de las que se muestran sólo algunas, y entre las que se puede hacer una **selección simple o múltiple**.

M.C. José Juan Hernández Mora

La clase *List*

- Las *List* se diferencian de las *Choices* en que muestran varios items a la vez y que permiten hacer selecciones múltiples.

M.C. José Juan Hernández Mora

Constructores

- List()
- List(int nl)
- List(int nl, boolean mult)
 - Donde:
 - nl es el número de líneas desplegadas
 - Mult si es true permite multiselección

M.C. José Juan Hernández Mora

Métodos

- addItem(String)
 - Añadir un ítem al final
- addItem(String, int)
 - Añadir un ítem indicando la posición

M.C. José Juan Hernández Mora

Métodos

- int getSelectedIndex()
 - Obtiene el índice del elemento seleccionado
- String getSelectedItem()
 - Obtiene la etiqueta del elemento seleccionado
- int[] getSelectedIndexes()
 - Obtiene los índices de los elementos seleccionados
- String[] getSelectedItems()
 - Obtiene las etiquetas de los elementos seleccionados

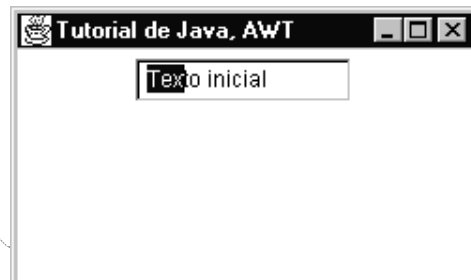
M.C. José Juan Hernández Mora

Ejemplo

- pregunta4.java

M.C. José Juan Hernández Mora

Campos de texto "TextField"



M.C. José Juan Hernández Mora

Campos de texto “TextField”

- Para la entrada directa de datos se suelen utilizar los campos de texto, que aparecen en pantalla como pequeñas cajas que permiten al usuario la entrada por teclado de una línea de caracteres.
- También permite al usuario editar ese texto.

M.C. José Juan Hernández Mora

Constructores de TextField

- TextField()
 - Crea un área de texto por defecto
- TextField(int ncol)
 - Crea un área de texto con n columnas
- TextField(String s)
 - Crea un área de texto con un texto
- TextField(String s, int ncol)
 - Crea un área de texto con un texto y con n columnas

M.C. José Juan Hernández Mora

TextField

- `String getText()`
 - Permiten obtener el texto del componente
- `setText(String str)`
 - Permiten establecer el texto del componente

M.C. José Juan Hernández Mora

Ejemplo

- `Leedatos.java`

M.C. José Juan Hernández Mora

Los *layout managers* o *manejadores de composición*

- Ayudan a adaptar los diversos Componentes que se desean incorporar a un **Panel**.
- Especifican la apariencia que tendrán los Componentes a la hora de colocarlos sobre un Contenedor, controlando tamaño y posición (*layout*) automáticamente

M.C. José Juan Hernández Mora

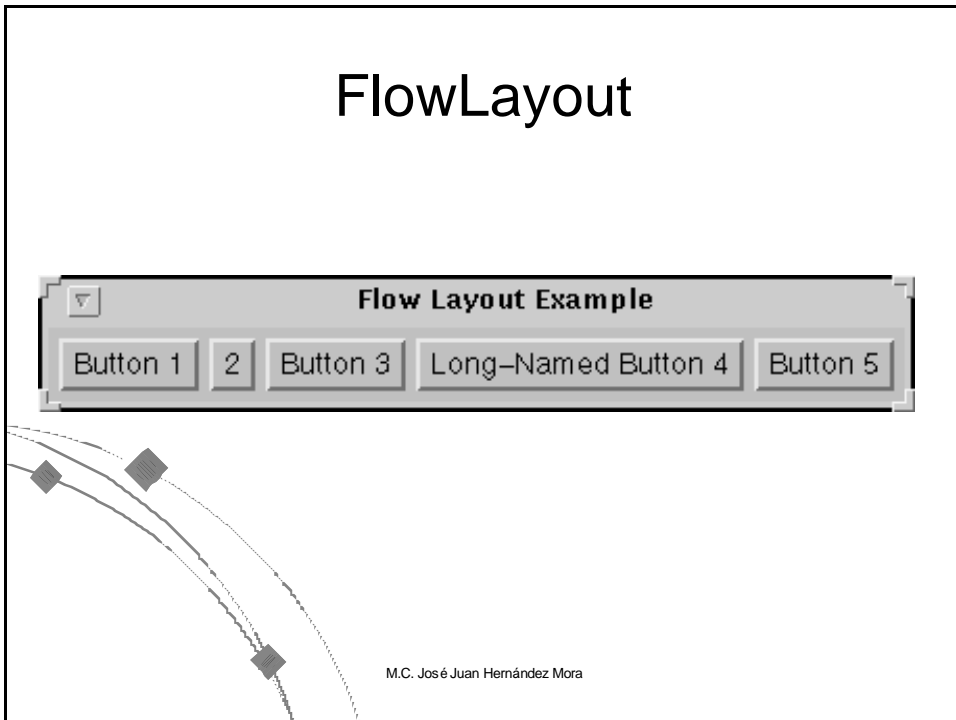
setLayout()

El método `setLayout(LayoutManager)` se encarga de establecer el gestor de organización

```
Void init() {  
    setLayout(new LayoutManager());  
    ...  
    ...  
}
```

M.C. José Juan Hernández Mora

FlowLayout



FlowLayout

- Es el más simple y el que se utiliza por defecto en todos los Paneles si no se fuerza el uso de alguno de los otros.
- Los Componentes añadidos a un **Panel** con **FlowLayout** se encadenan en forma de lista. La cadena es horizontal, de izquierda a derecha, y se puede seleccionar el espaciado entre cada Componente.

La clase ***FlowLayout*** tiene tres constructores:

- `FlowLayout()`;
- `FlowLayout(int alineacion)`;
- `FlowLayout(int alineacion, int horizontal, int vertical)`;

Donde:

- Alineacion.- es la alineación de los componentes
- Horizontal.- es el espacio horizontal de los componentes
- Vertical.- es el espacio vertical de los componentes

M.C. José Juan Hernández Mora

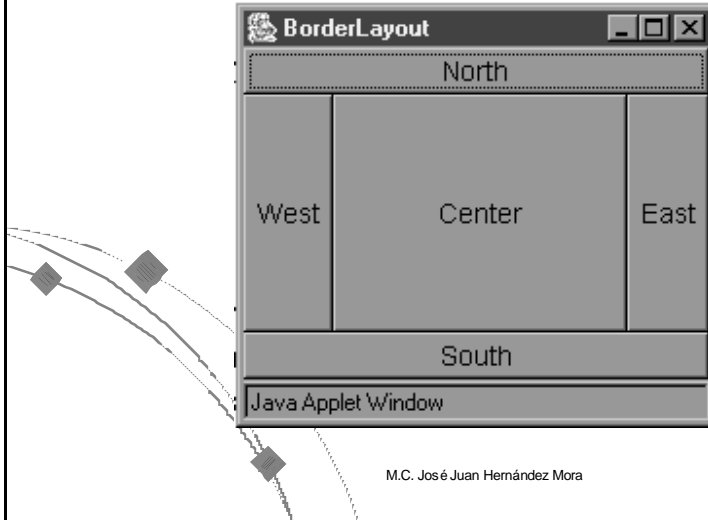
Alineación

Se puede establecer la alineación de los componentes (centrados, por defecto), por medio de las constantes:

- `FlowLayout.LEFT`
- `FlowLayout.CENTER`
- `FlowLayout.RIGHT`.

M.C. José Juan Hernández Mora

BorderLayout



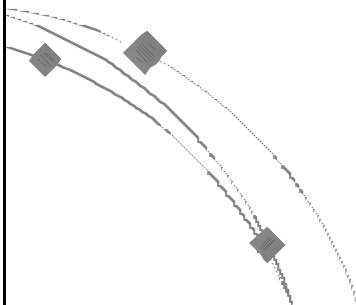
BorderLayout

- **BorderLayout** define cinco áreas: **North**, **South**, **East**, **West** y **Center**.
- Si se aumenta el tamaño de la ventana todas las zonas se mantienen en su mínimo tamaño posible excepto **Center**, que absorbe casi todo el crecimiento.

M.C. José Juan Hernández Mora

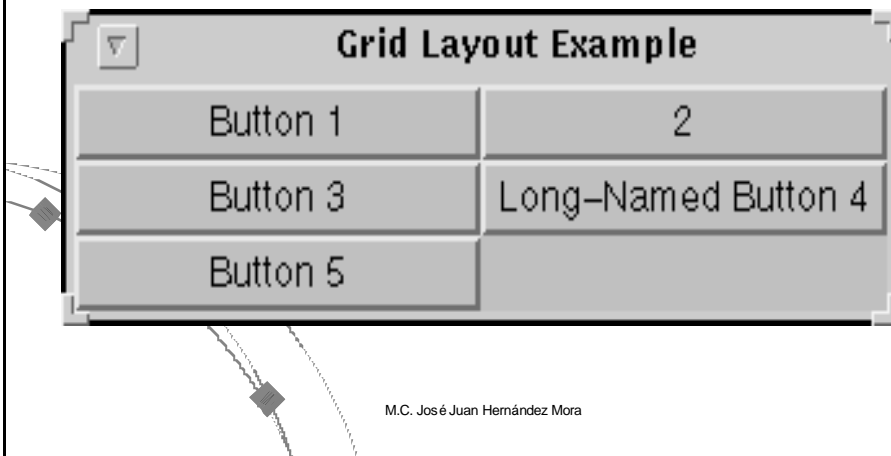
Ejemplo

- `bordes.java`



M.C. José Juan Hernández Mora

GridLayout



M.C. José Juan Hernández Mora

GridLayout

- Con ***GridLayout*** las componentes se colocan en una matriz de celdas.
- Todas las celdas tienen el mismo tamaño.
- Cada componente utiliza todo el espacio disponible en su celda, al igual que en ***BorderLayout***.

M.C. José Juan Hernández Mora

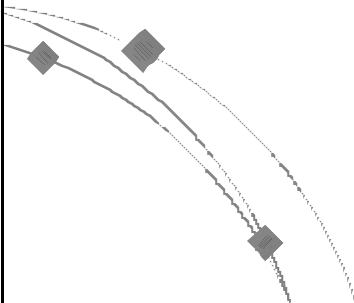
GridLayout tiene dos constructores:

- `GridLayout(int nfil, int ncol)`
 - Divide el Panel en *nfil* filas y *ncol* columnas
- `GridLayout(int nfil, int ncol, int horizontal, int vertical)`
 - Divide el Panel en *nfil* filas y *ncol* columnas y con un espacio horizontal y vertical dado en pixeles

M.C. José Juan Hernández Mora

Ejemplo

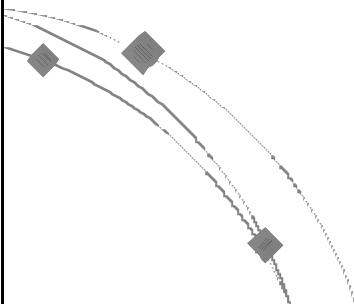
- Malla.java



M.C. José Juan Hernández Mora

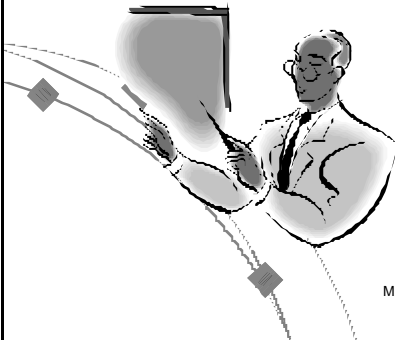
Examen

- Programas de examen



M.C. José Juan Hernández Mora

Por su Atención



¡¡Gracias!!

M.C. José Juan Hernández Mora