

Aplicaciones gráficas en Java

Librería Swing

- Sumario:
 - Breve introducción al paquete Swing
 - Tipos de clases del paquete
 - Definición y manejo de contenedores, diálogos y layouts
 - Diseño mediante IDE (Eclipse) y ejemplos.
 - Resumen y bibliografía



La librería Swing

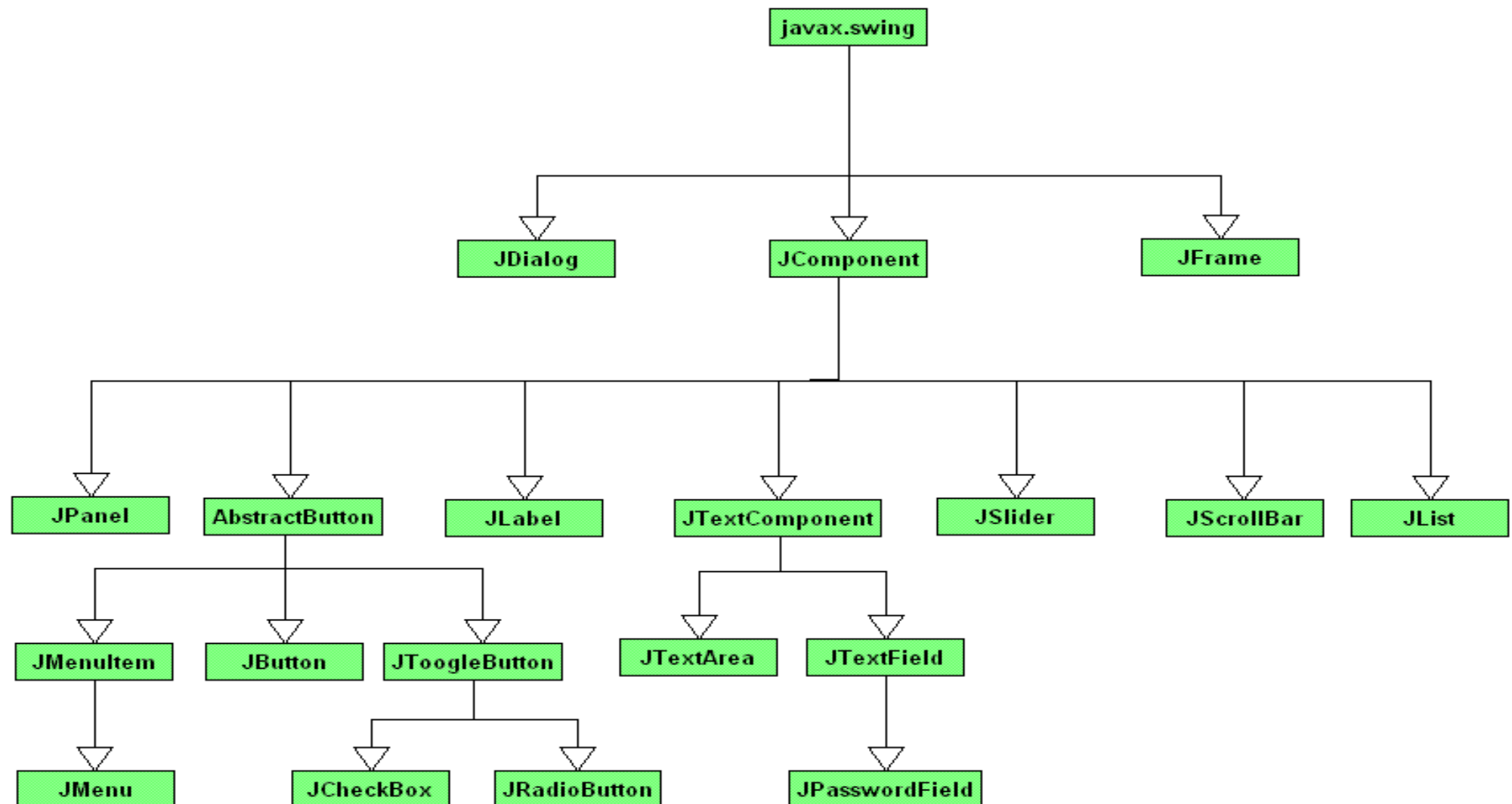
- Pertenece a las JFC (Java Foundation Classes).
- Biblioteca gráfica de Java desde la versión 1.2.
- Contenido en el paquete “javax.swing”.
- Creada a partir de “java.awt”.
- Permite una interfaz adaptada a cada SO sin cambio de código.
- Fácil manejo.

Look & Feel

- Para una interfaz adaptada a cada S.O debemos incluir el siguiente código en el método main:

```
try {  
  
    UIManager.setLookAndFeel (UIManager.getSystemLookAndFeelClassName ())  
    ;  
    }  
    catch (Exception e) {  
        e.printStackTrace ();  
    }  
}
```

Clases del paquete Swing (I)



Clases del paquete Swing (II)

- Todos los componentes heredan de *javax.swing.JComponent*
- JFrame será la base para la aplicación principal.
- JDialog construirá los diálogos (ventanas).
- El resto de clases serán componentes simples.
- Usar en todas las clases *import javax.swing.*;*
y import java.awt.;*
- Todas las componentes permiten fijar un mnemotécnico: *componente.setMnemonic(KeyEvent.VK_letra);*
- Todas las componentes permiten fijar “tooltips”.

Clase JButton (I)

- Es un botón que puede contener texto, gráficos, o ambos.



- Fijar el texto siempre centrado, en caso de contener una imagen, ha de ir a la izquierda o encima del texto.
- Incluir “...” si precisa de más información para llevar a cabo la ejecución.

Clase JButton (II)

- Métodos importantes:
 - setText("Texto");
 - setToolTipText("Tooltip");
 - setBackground(new Color(R, G, B));
 - setForeground(Color.color);
 - setIcon(new ImageIcon("ruta"));
 - setFont(new Font("tipo", estilo, tamaño));
 - setBounds(new Rectangle(posX,posY,tamX,tamY));
- Y sus correspondientes *get*.

Clase JButton (III)

- Ejemplo:

```
JButton boton1 = new JButton();
```

```
boton1.setBounds(new Rectangle(107, 50, 102, 41));
```

```
boton1.setBackground(new Color(91, 238, 89));
```

```
boton1.setForeground(Color.red);
```

```
boton1.setToolTipText("Prueba");
```

```
boton1.setFont(new Font("Comic Sans MS", Font.BOLD, 14));
```

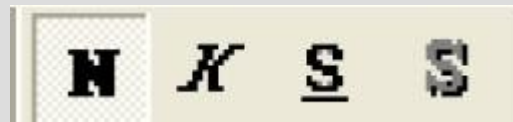
```
boton1.setText("Botón de prueba");
```

```
boton1.setMnemonic(KeyEvent.VK_B);
```

```
.....
```


Clase JToggleButton (I)

- Es un botón que representa dos estados (On y Off).
- Mismas características que el JButton.
- Puede emplearse como dos tipos de opciones.
 - Independientes (Checkboxes).



- Exclusivas (RadioButton).



Clase JToggleButton (II)

- Métodos importantes:
 - Mismos métodos que JButton, pero añadiendo algunos nuevos.
 - isSelected();
 - setSelected(boolean);
- Es posible añadir botones a grupos, para garantizar opciones mutuamente exclusivas.

Clase *JCheckBox*

- Es un control que representa dos estados (On y Off).



- Mismas propiedades y métodos que los anteriores.
- Se pueden emplear Mnemotécnicos.
- Métodos *isSelected()* y *setSelected(boolean)*

Clase JRadioButton (I)

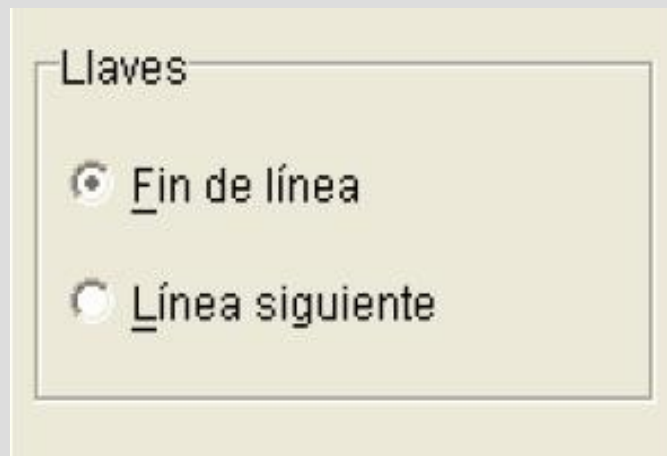
- Permiten seleccionar **una** única opción dentro de un conjunto de opciones relacionadas.



- Sólo puede haber una opción seleccionada a la vez.
- Aunque tiene la misma función que los botones de conmutación agrupados, conviene usar los radio botones en diálogos, y los de conmutación en barras de herramientas.
- Métodos similares a los de JCheckBox

Clase *JRadioButton* (II)

- Los botones de radio (como los checkboxes), suelen aparecer agrupados, con una leyenda identificativa.



- Para ello, en Swing se utiliza un panel, con un borde y un título, agrupándolos en un *ButtonGroup*.

Clase JComboBox (I)

- Esta componente nos permite, al hacer click sobre ella, seleccionar una opción de entre un conjunto, todas ellas mutuamente exclusivas.



- El texto de los ítems ha de ir capitalizado.
- Facilitar el acceso por teclado mediante etiquetas con mnemotécnicos.

Clase JComboBox (II)

- Se pueden generalizar en dos tipos:
 - Editables
 - No editables
- Métodos:
 - `setEditable(boolean);`
 - `addItem(Objeto);`
 - `getSelectedItem();`

Clase JComboBox (III)

- Ejemplo:

```
private String[] contenidos ={"Manzana","Naranja","Platano","Tiburon","Tomate","Trucha"};
```

```
private JComboBox jComboBox1 = new JComboBox(contenidos);
```



```
String nombre= (String) jComboBox1.getSelectedItem();
```

```
jTextField1.setText("Esto es un " + nombre);
```

```
jLabel2.setIcon(new ImageIcon("imagenes/"+nombre+".gif"));
```


Clase JList (I)

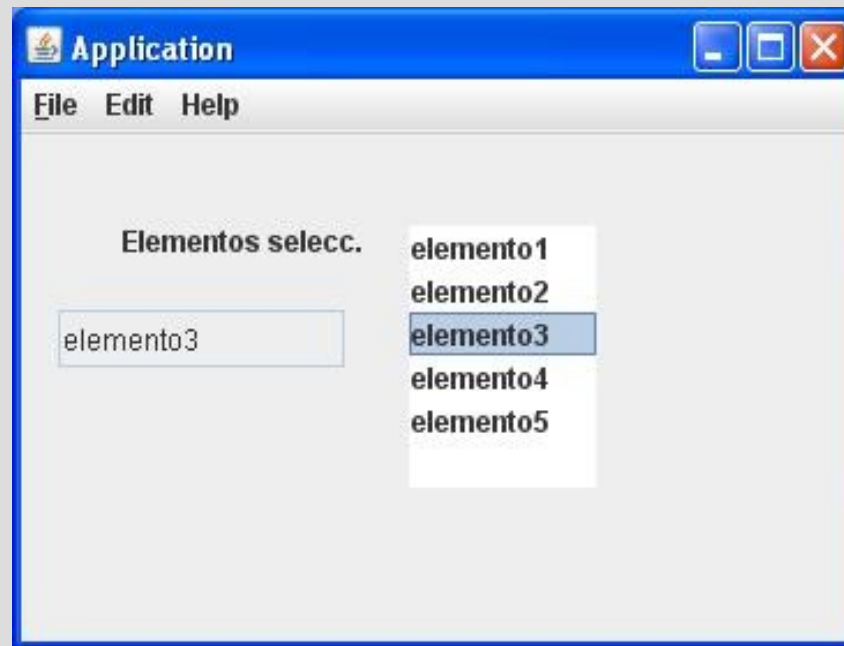
- Es un componente que muestra un conjunto de ítems de texto, gráfico o ambos.
- Permite tres tipos de selección:
 - Ítem único
 - Rango simple
 - Rango múltiple
- Mediante el método:
 - `.setSelectionMode(ListSelectionMode.SELECTION) ;`
 - Donde SELECTION puede ser:
 - `SINGLE_SELECTION`
 - `SINGLE_INTERVAL_SELECTION`
 - `MULTIPLE_INTERVAL_SELECTION`

Clase JList (II)

- Ejemplo:

```
private String[] contenidos={"elemento1",  
    "elemento2","elemento3","elemento4","elemento5"};
```

```
private JList=new JList(contenidos);
```



```
jTextField1.setText((String)jList1.getSelectedValue());
```

Otras clases de componentes

- JProgressBar



- JSlider



Componentes para texto (I)

- JLabel
 - Muestra texto, gráficos o ambos, sólo lectura.
 - Ha de estar desactivado si lo está el componente.
- JTextField
 - Muestra una línea de texto que puede ser editable.
 - Con *setText("Texto")* se le asigna el texto.

Componentes para texto (II)

- JPasswordField
 - Oculta los caracteres introducidos por el usuario.
 - *setEchoChar('char')* indica el carácter de máscara.
 - *getPassword()* recupera el password introducido.
- JTextArea
 - Espacio rectangular en el que ver y editar múltiples líneas de texto.
 - Para que aparezcan barras de scroll debe ir dentro de un JScrollPane.

Menús (I)

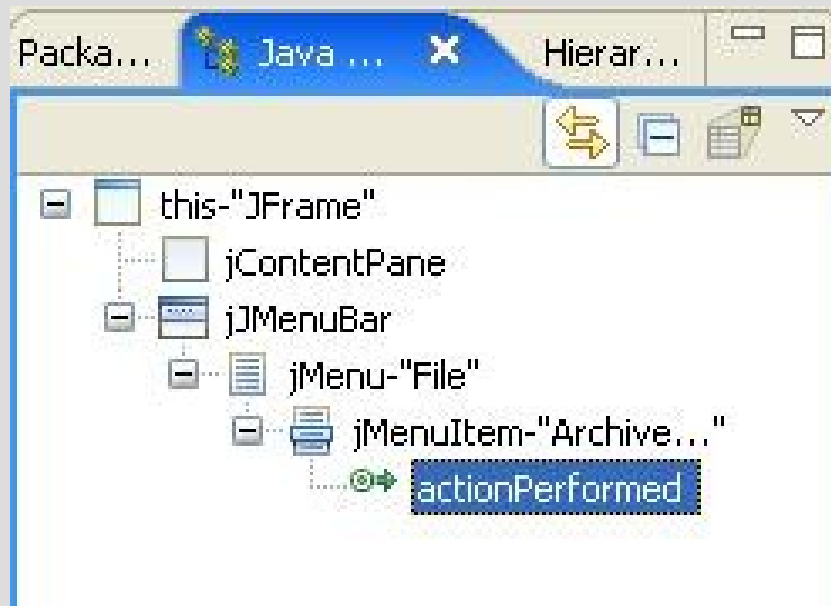
- Los menús han de ir en la ventana principal de la aplicación.
- Es posible asignarles un gráfico.
- Cada título de menú debe tener su mnemotécnico.
- Pueden ser de tres tipos:
 - Drop-Down
 - Submenu
 - Contextuales

Menús (II)

- Los menú “Drop-Down” son los que saldrán al, por ejemplo, hacer click en Archivo.
- Los submenus son aquellos que salen como un grupo de un elemento de menú.
- Los menús contextuales, (clase JPopupMenu) son aplicables a la región en la que está localizado el puntero del ratón.
- Los mnemotécnicos de los menús contextuales han de coincidir con los del menú “Drop-Down”.

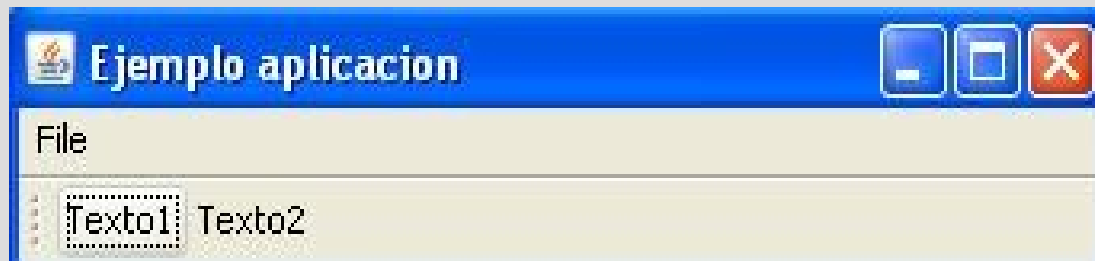
Menús (III)

- Son las clases *JMenuBar*, *JMenu* y *JMenuItem*.
- Han de seguir la siguiente jerarquía:



Contenedores (I)

- Clase *JToolBar*



- Son botones de comando o conmutación.
- Se suelen emplear gráficos.
- Métodos *setFlotable(boolean)* y *addSeparator()*

Contenedores (II)

- Clase *JPanel* :
 - Es un contenedor que agrupa componentes dentro de una ventana.
 - Los layouts permiten una correcta posicionación de los componentes.
- Clase *JTabbedPane*:
 - Es un contenedor que permite tener varios componentes separados por pestañas.
 - Métodos

Diálogos

- Son ventanas mas limitadas que los Frames, y dependientes de estos, si se destruye el Frame, también lo hace el diálogo.
- Pueden ser:
 - No modales: No impiden interactuar con el Frame.
 - Modales: Impiden interactuar con el resto.
- Los más importantes son: JOptionPane y JFileChooser.

Diálogo JOptionPane (I)

- Permite adaptar y crear varias clases de diálogos, especificando por ejemplos los iconos, el título y texto de los diálogos.
- Los iconos estándar son: *question, information, warning y error*.
- Los métodos principales son:
 - showMessageDialog
 - showConfirmDialog

Diálogo JOptionPane (II)

- showMessageDialog:

- `JOptionPane.showMessageDialog(this, "Mensaje", "Titulo", JOptionPane.WARNING_MESSAGE);`

- showConfirmDialog:

- ```
int resp = JOptionPane.showConfirmDialog(this, "Mensaje", "Titulo",
 JOptionPane.YES_NO_OPTION);
```

- ```
int resp =
    JOptionPane.showConfirmDialog(this, "Mensaje", "Titulo", JOptionPane.
    YES_NO_CANCEL_OPTION);
```

- ```
int resp = JOptionPane.showConfirmDialog(this, "Mensaje", "Titulo",
 JOptionPane.OK_CANCEL_OPTION);
```

- ```
if (resp == JOptionPane.YES_OPTION) {
    .....
}
```

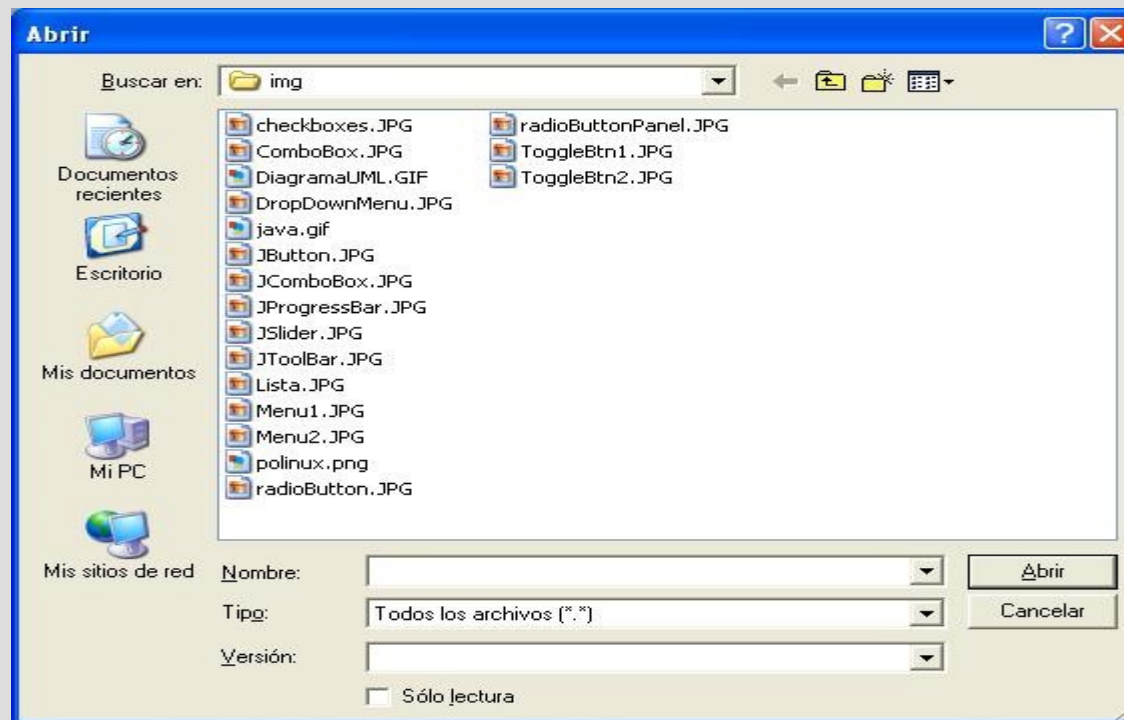
Diálogo *JFileChooser* (I)

- Permite navegar por el sistema de ficheros, y seleccionar uno o varios ficheros.
- Ejemplo:

```
private JFileChooser jFileChooser1 = new JFileChooser();  
  
. . .  
if (jFileChooser1.showOpenDialog(this)==jFileChooser1.APPROVE_OPTION)  
{  
    jTextField1.setText("APPROVE_OPTION");  
    jTextField2.setText(jFileChooser1.getSelectedFile().getName());  
}  
else  
    jTextField1.setText("CANCEL_OPTION");
```

Diálogo JFileChooser (II)

- Métodos importantes:
 - multiSelectionEnabled(boolean);
 - getSelectedFile();



Layouts (I)

- Indican la forma de organizar los componentes dentro de un contenedor, determinando el tamaño y la posición.
- Para su uso:
 - Crear el contenedor.
 - Establecer el layout.
 - Agregar los componentes el contenedor.

Layouts (II)

- Tipos de layouts:
 - FlowLayout
 - BorderLayout
 - GridLayout
 - BoxLayout
 - GridBagLayout
- Por defecto:
 - JPanel -> FlowLayout
 - JFrame, JDialog -> BorderLayout

Layouts (III)

- FlowLayout:
 - Es el más simple, los componentes añadidos a un contenedor se disponen en una o mas filas, de izquierda a derecha y de arriba a abajo.
- BorderLayout:
 - Utiliza 5 áreas para colocar los componentes: Norte, Sur, Este, Oeste y Centro. Si alguna no se ocupa, se expande la contigua.

Layouts (IV)

- GridLayout:
 - El controlador se crea con un determinado numero de filas y columnas.
 - Los componentes se sitúan de forma secuencial, de izquierda a derecha y de arriba a abajo.
 - El tamaño de las celdas es idéntico.
- BoxLayout:
 - Permite organizar los componentes en una línea horizontal o vertical, sin dejar espacio entre los componentes.

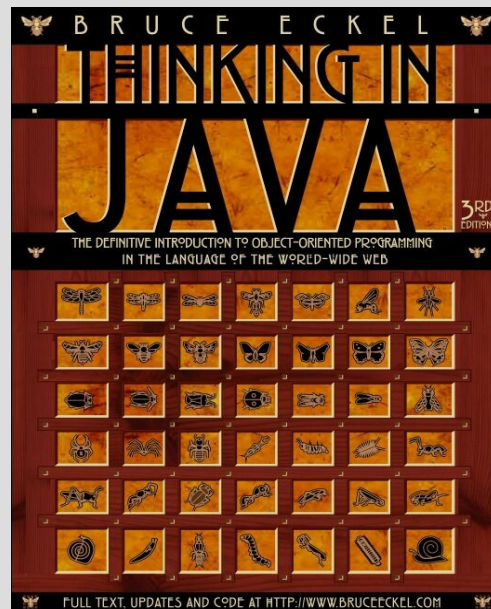
Crear nuestra aplicación: Eclipse

- Para crear las aplicaciones, basándonos en lo explicado, utilizaremos el IDE *Eclipse*, junto con el plugin “*Visual Editor*”.
- Se puede descargar gratuitamente en <http://www.eclipse.org>
- Necesario Java instalado para poder ejecutarlo.



Bibliografia

“Thinking in Java”



“The JFC Swing tutorial : a guide to constructing GUIs”



Material adicional

- Tanto el pdf de la charla como ejemplos de varias componentes se colgarán en:
 - <http://sepho.almaqui.net/swing/>
- Para cualquier duda podeis mandar un correo:
 - jacandel@ei.upv.es

