

En general, para definir elementos de Java Swing debemos eliminar las J inicial y las vocales, y extraer las tres consonantes más significativas del nombre del control. A continuación figura la lista de componentes y sus prefijos asociados:

1. NOMENCLATURA DE SWING CONTAINERS

Contenedor	Prefijo
JInternalFrame	ifrm
JPanel	pan
JScrollPane	scp
JTabbedPane	tpn
JToolBar	tlb

2. NOMENCLATURA DE SWING CONTROLS

Control	Prefijo
JButton	btn
JButtonGroup	btg
JCheckBox	cbx
JComboBox	cmb
JLabel	lbl
JList	lst
JPasswordField	pwd
JProgressBar	pgb
JScrollBar	scb
JTable	tbl
JTextArea	txa
TextField	txt
JTextPane	txp
JTree	jt
JDateChooser	jdc
JCalendar	jcl

JRadioButton	jrb
--------------	-----

3. NOMENCLATURA DE SWING MENUS

Menu	Prefijo
JMenu	mnu
JMenuBar	mnb
JMenuItem	mni

4. NOMENCLATURA DE SWING WINDOWS

Window	Prefijo
JColorChooser	cch
JDialog	dlg
JFileChooser	jfc
JFrame	frm
JOptionPane	opt

5. OTROS

Window	Prefijo
DefaultTableModel	dtm
JDialog	dlg
JFileChooser	jfc
JFrame	frm
JOptionPane	opt

documentación java swing

Crear diálogos modales en Java Swing

28 diciembre, 2015

Un diálogo modal respecto al sistema es una ventana que no va ceder el foco a ninguna otra aplicación hasta que se produzca una determinada acción sobre él. Por ejemplo, podríamos utilizarlos en caso de que dispongamos de un JFrame que abre una nueva ventana y al que no podemos acceder a menos que cerremos dicha ventana.

Para ello, hay que tener en cuenta que esa nueva ventana que abrimos no puede ser un JFrame, sino un JDialog. El procedimiento es el siguiente:

- Hemos definido un JDialog con los componentes que deseemos. Su constructor dispondrá de dos parámetros: parent y modal, que indican el padre del JDialog, y si la ventana es modal (hasta que no se cierre no podemos acceder al padre) o no modal.

```
public DialogSalonHabana(java.awt.Frame parent, boolean modal) {  
    super(parent, modal);  
    initComponents();  
}
```

- La ventana principal dispone de un evento que permite abrir el nuevo JDialog, con el siguiente código:

```
private void mniNuevaVentanaActionPerformed(java.awt.event.ActionEvent evt) {  
    NuevaVentanaDialog nv = new NuevaVentanaDialog(this,true);  
    JDialog jd = new JDialog(nv, "Dialogo modal", Dialog.ModalityType.DOCUMENT_MODAL);  
    nv.setVisible(true);  
}
```

De este modo, cuando mediante el evento mniNuevaVentanaActionPerformed abramos una ventana NuevaVentanaDialog, no podremos volver a la ventana principal hasta que ésta no sea cerrada.

java JDialog JFrame Swing

Eventos y Listeners en Java Swing

14 diciembre, 2015

Aquí os dejo la presentación de clase sobre Eventos y Listeners en Java Swing.

Está disponible en SlideShare, en el siguiente enlace: <http://www.slideshare.net/laurafolgadogalache/eventos-y-listeners-en-java-swing>

EVENTOS Y LISTENERS EN JAVA SWING

CICLO: Desarrollo de Aplicaciones Multiplataforma
MÓDULO: Desarrollo de Interfaces

LAURA FOLGADO GALACHE | WWW.LAURAFOLGADO.ES

1 of 10

Eventos y listeners en Java Swing from [Laura Folgado Galache](#)

eventos java listeners swing



ISO DE JAVA SWING | Laura Folgado Galache
CARTAS PARA RALLY-COACH

Repaso de Java Swing con Rally Coach

27 noviembre, 2015

En esta presentación podéis encontrar algunos ejercicios de repaso para realizar con la estructura de Aprendizaje Cooperativo llamada Rally-Coach.

Consiste en varias cartas (quitando la portada, las 8 primeras pueden imprimirse de 4 en 4 en dos páginas horizontales y las 8 últimas como reverso) que se reparten a parejas de alumnos (una baraja de 8 por cada pareja). El alumno que hace de Coach en primer lugar levanta una carta y realiza el ejercicio mientras explica a su compañero los pasos que va dando; éste asiente o corrige. Luego el alumno que ha hecho de Coach pasa a ser el que escucha, y el que escuchaba pasa a hacer de Coach, levantando otra carta y haciendo el ejercicio correspondiente.



Repaso de Java Swing usando Rally-Coach from [Laura Folgado Galache](#)

java presentación repaso swing

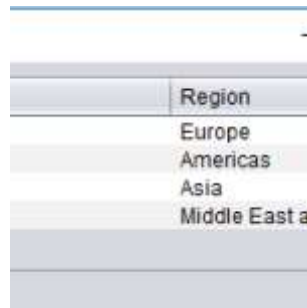
Proyecto conjunto con GitHub y HR

17 noviembre, 2015

Basándonos en el componente de acceso a datos de HR creado en el módulo Acceso a Datos, y poniendo en práctica todo lo aprendido hasta ahora con Java Swing debéis poneros de acuerdo para crear un equipo de desarrollo donde integréis todos los elementos utilizando el repositorio GitHub.

Los diferentes perfiles de desarrollo que utilizaréis son los siguientes:

- **Diseñador-programador** (3 personas): es el trabajador encargado del diseño de los componentes de Java Swing de modo que todos tengan un aspecto y funcionalidad similar y cumplan con los estándares de usabilidad. No deben olvidar la asignación de nombres siguiendo los estándares para cada uno de los elementos (deben coordinarse para seguir la misma nomenclatura). Una vez diseñados estos componentes deberán continuar su trabajo de programadores.
- **Revisor-programador** (3 personas): una vez desarrollados los componentes, se encargarán de comprobar que funcionan correctamente. Además, deben realizar la función de programadores.
- **Jefe de operación** (3 personas): encargado de planificar el trabajo a realizar, la asignación de recursos humanos en cada puesto de trabajo y en cada momento. Además, realiza los procesos de programador.



Region
Europe
Americas
Asia
Middle East a

Cargar datos de una base de datos en un JTable

16 noviembre, 2015

Ya que en Acceso a Datos trabajamos con la base de datos HR, vamos a reutilizar uno de sus métodos para completar un jTable de nuestro formulario. Para ello debemos añadir en un JFrame un jTable y crearnos un modelo de datos de tipo DefaultTableModel en la clase de nuestro JFrame.

```
DefaultTableModel dtmModelo;
```

Además, dentro del método public nombreJPanel () además de llamar al método initComponents; llamaremos también a cargarTabla();

El método cargarTabla será público y no devolverá nada.

En primer lugar deberá definir el modelo DefaultTableModel para que no pueda ser editado por el usuario.

```
this.dtmModelo = new DefaultTableModel() {  
    @Override  
    public boolean isCellEditable(int fila, int columna) {  
        return false; //Con esto conseguimos que la tabla no se pueda editar  
    }  
};
```

A continuación asignamos dicho modelo a la tabla:

```
this.tblTabla = new JTable(dtmModelo);
```

Al modelo le añadimos los nombres de columnas que vamos a necesitar. En este caso, de Regiones tomaremos el ID y el nombre de la región.

```
this.dtmModelo.addColumn("ID");  
this.dtmModelo.addColumn("Region");
```

Seguidamente realizamos la conexión a la base de datos HR y obtenemos, en este caso, el ArrayList de regiones que se lo asignaremos a una variable de tipo ArrayList.

```
Hr hr = new Hr();  
ArrayList<Region> listaRegiones = hr.listarRegions();
```

Seguidamente hacemos un bucle para recorrer esta lista de regiones, definiéndonos una fila en la que indicaremos el número de columnas que vamos a utilizar y asignándole a las posiciones de la fila la información extraída del array. Finalmente, en cada pasada del bucle añadiremos la fila al modelo:

```
for (int i = 0; i < listaRegiones.size(); i++) {  
    Object[] fila = new Object[2];  
    fila[0]= listaRegiones.get(i).getRegionId();  
    fila[1]= listaRegiones.get(i).getRegionName();  
    this.dtmModelo.addRow(fila);  
}
```

Para actualizar la información será necesario realizar el método updateUI a la tabla y a continuación ejecutar el método setViewportView del JScrollPane para actualizar la información que contiene.

```
this.tblTabla.updateUI();  
this.scpScrollPane.setViewportView(tblTabla);
```

¡No olvides implementar el **try-catch** correspondiente! De lo contrario te dará error.

base de datos java JTable swing

Cómo hacer que un elemento sea transparente (Tip)

10 noviembre, 2015

Pues no ha echado horas ni nada Dani para poner los iconos de sus aplicaciones en un JDesktop pane... eso sí, se le resistió cómo poner el fondo de las imágenes transparentes y aquí tenemos la solución:

Clic derecho sobre el elemento -> Events -> PropertyChange -> PropertyChange

En el código de dicho evento:

```
element.setContentAreaFilled(false);
```

¡Gracias, Dani!