

Práctica VII - Listas (JList)

Los `JList` funcionan de forma similar a los `JComboBox`, podemos asociar un modelo dinámico del tipo `DefaultListModel` y también podemos controlar el evento de selección mediante `valueChanged`.

Manipular el evento de cambio de elemento de un `JList`

El evento `valueChanged` es invocado cuando se selecciona un valor distinto en el `JList` y podemos recuperar el valor seleccionado mediante `this.jList1.getSelectedValue()`. El código para agregar un nuevo elemento a un modelo `DefaultListModel` debe ser similar al siguiente:

```
private void jList1ValueChanged(javax.swing.event.ListSelectionEvent evt) {  
    String elemento = this.jList1.getSelectedValue();  
    String formato = String.format("Seleccionado: %s", elemento);  
    this.jLabel1.setText(formato);  
}
```

Ajustar el modelo de `JList`

Podemos asociar un modelo de elementos a través de `DefaultListModel`. Si por ejemplo queremos una lista que contenga datos de tipo `String` usamos `DefaultListModel<String>`. El modelo nos permite manipular elementos dinámicos que puedan cambiar en transcurso de la aplicación, por ejemplo si tenemos que agregar o quitar elementos de la lista. El modelo puede ser un atributo de nuestra clase `JFrame` y luego enlazarlo en el constructor con `this.jList1.setModel(elementos)`:

```
public class MiVentana extends javax.swing.JFrame {

    DefaultListModel<String> elementos = new DefaultListModel();

    public MiVentana() {
        initComponents();

        this.jList1.setModel(this.elementos);

        this.elementos.addElement("Hola");
        this.elementos.addElement("Mundo");
    }

    ...
}
```

Así por ejemplo si queremos agregar otro elemento desde un botón en su evento `mouseClicked` hacemos:

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    this.elementos.addElement("Otro elemento");
}
```

Problemas

- Crea dos modelos `DefaultListModel<String>` llamados `modeloA` y `modeloB` como atributos de la clase `JFrame`.
- Crea dos listas `JList` llamadas `jListA` y `jListB`.
- Inserta algunos (al menos 3) elementos al `modeloA`.
- Controla el evento `valueChanged` de `jListA` que recupere el elemento seleccionado en la lista mediante `jListA.getSelectedValue()`, guardalo en la variable local `String elementoSeleccionado`.
- Recupera el índice del elemento seleccionado mediante `jListA.getSelectedIndex()` y guardalo en la variable local `int i`.
- Elimina de `modeloA` el elemento en el índice `i` mediante `modeloA.remove(i)`.
- Inserta el elemento seleccionado a `modeloB` mediante

```
modeloB.addElement(elementoSeleccionado) .
```

- Repite los pasos anteriores pero ahora para `jListB` , es decir, ahora debemos quitar los elementos de la lista `jlistB` y pasarlos de regreso a `jListA` .

Diplomado de Java - Alan Badillo Salas (badillo.soft@hotmail.com)

Instituto Politécnico Nacional - Centro de Investigación en Cómputo