JFileChooser - ChuWiki

chuidiang.org/index.php

ChuWiki 16 de marzo de 2024

Qué es JFileChooser

`JFileChooser` es una clase java que nos permite mostrar fácilmente una ventana para la selección de un fichero.

Cómo usar JFileChooser

Veamos ejemplos de cómo usar `JFileChooser`.

Abrir un fichero : showOpenDialog()

Si queremos abrirlo para leer el fichero, podemos llamarlo así

```
JFileChooser fileChooser = new JFileChooser();
int seleccion = fileChooser.showOpenDialog(parent);
```

`parent` es cualquier componente java de alguna de nuestras ventanas, la que queramos que haga de ventana padre del diálogo `JFileChooser`. Nuestro diálogo `JFileChooser` no se podrá ir detrás de su ventana padre. Habitualmente se suele poner como padre el botón que hace que se abra `JFileChooser`.

La llamada `showOpenDialog()` se quedará bloqueada hasta que el usuario elija un fichero y cierre la ventana. A la vuelta, en `seleccion` tendremos uno de los siguiente valores

- **JFileChooser.CANCEL_OPTION** Si el usuario le ha dado al botón cancelar.
- **JFileChooser.APPROVE_OPTION** Si el usuario le ha dado al botón aceptar
- **JFileChooser.ERROR_OPTION** Si ha ocurrido algún error.

Comprobando que se ha dado al botón aceptar, podemos obtener el fichero seleccionado por el usuario así

```
if (seleccion == JFileChooser.APROVE_OPTION)
{
   File fichero = fileChooser.getSelectedFile();
   // Aquí debemos abrir y leer el fichero.
   ...
}
```

Salvar un fichero: showSaveDialog()

Para seleccionar un fichero para guardar datos, el mecanismo es igual, pero se llama al método `showSaveDialog()`

```
JFileChooser fileChooser = new JFileChooser();
int seleccion = fileChooser.showSaveDialog(areaTexto);
if (seleccion == JFileChooser.APPROVE_OPTION)
{
   File fichero = fileChooser.getSelectedFile();
   // Aquí debemos abrir el fichero para escritura
   // y salvar nuestros datos.
   ...
}
```

La única diferencia entre uno y otro es la etiqueta del diálogo y de los botones. Uno pondrá "**Abrir**" y otro "**Guardar**"

Configuración de JFileChooser

Hay varias cosas que podemos configurar cuando abramos un `JFileChooser`. Veamos algunas de ellas.

Directorio inicial

El `JFileChooser` se abre por defecto en el directorio *HOME* del usuario (C:\Documents and Settings\usuario en Windows, /home/usuario en linux). Podemos elegir el directorio en el que queremos que se abra llamando al método `setCurrentDirectory()` pasando el directorio en cuestión.

Seleccionar un directorio

Por defecto, un `JFileChooser` sólo permite elegir ficheros. Si queremos que permita elegir también directorios, debemos llamar a `setFileSelectionMode()`, pasando como parámetro uno de las siguientes constantes

- **JFileChooser.FILES_ONLY** si queremos que sólo se puedan elegir ficheros. Es la opción por defecto.
- **JFileChooser.DIRECTORIES_ONLY** si queremos que sólo se puedan elegir directorios.
- **JFileChooser.FILES_AND_DIRECTORIES** si queremos que se puedan elegir tanto directorios como ficheros.

Filtrar los ficheros visibles

Si no queremos que el `JFileChooser` muestre todos los ficheros del directorio, podemos añadirle un filtro. Básicamente hay que hacer una clase que herede de `FileFilter` y definir dos métodos:

- método `accept()`. Este método recibe un parémetro `File` y nuestro código debe decidir si pasa o no el filtro, devolviendo `true` o `false`.
- método `getDescription()` que simplemente devuelve una cadena de texto describiendo el filtro. Este texto será visible para el usuario.

Por ejemplo, si sólo queremos ver ficheros *.jpg*, podemos hacer este filtro

```
import javax.swing.filechooser.FileFilter;
...
public class FiltroDeJPG extends FileFilter
{
   public boolean accept (File fichero)
   {
      return fichero.getName().endsWith(".jpg");
   }
   public String getDescription()
   {
      return ("Filtro JPGs");
   }
}
```

Una vez creado nuestro filtro, sólo tenemos que pasarlo a nuestro `JFileChooser`

```
fileChooser.setFilter(new FiltroDeJPG());
```

Filtro por extensión del fichero

Para filtros normales, como verificar la extensión de un fichero, Java ya tiene filtros implemenentados. `FileNameExtensionFilter` es un filtro ya hecho para verificar extensiones de fichero. En el constructor pasamos como primer parámetro el nombre del filtro, visible para el usuario, y luego las extensiones que queramos, sin el punto. El filtro es listo y no distingue mayúsculas de minúsculas.

```
JFileChooser jf = new JFileChooser();
FileNameExtensionFilter filter = new FileNameExtensionFilter("JPG & GIF", "jpg",
"gif");
jf.setFilter(filter);
```

Poner varios filtros

Si queremos varios filtros, podemos llamar a `addChoosableFileFilter()`. Esto irá añadiendo varios filtos que se mostrarán al usuario y podrá utilizar para localizar más fácilmente el tipo de fichero que está buscando.

Por defecto, `JFileChooser` muestra una opción que permite seleccionar todos los ficheros. Podemos deshabilitarla con `setAcceptAllFileFilterUsed(false)`. De esta forma el usuario no podrá saltarse los filtros que le pongamos.

```
fileChooser.setAcceptAllFileFilterUsed(false);
fileChooser.addChoosableFileFilter(new FileNameExtensionFilter("Tablas de
cálculo", "csv", "xlsx"));
fileChooser.addChoosableFileFilter(new FileNameExtensionFilter("Textos", "txt",
"docx"));
fileChooser.addChoosableFileFilter(new FileNameExtensionFilter("Imágenes", "jpg",
"gif"));
```

El filtro visible inicialemnte será el primero que añadamos a `JFileChooser`. Si queremos que sea visible uno determinado, podemos llamar a `setFileFilter()` como antes. Ese método, además de añadir el filtro, igual que `addChoosableFileFilter`, hace que sea el filtro visible por defecto.

Categoría:

Java:SWING