

Desarrollo de Interfaces: Hoja de Ejercicios 1

- 1- a) Crear una aplicación que muestra una ventana con 3 elementos: un campo de texto, un botón y un JList. La aplicación listará las cadenas de texto introducidas en el campo de texto, al pulsar el botón. (Cada vez que se pulse, se añadirá el texto que aparece en el JTextField).

b) Mismo ejercicio anterior, pero añadiendo las cadenas de texto a un JComboBox.

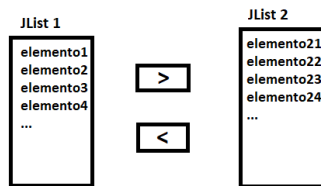
c) Añadir una comprobación para que no permita añadir dos cadenas iguales. Si ya existe la cadena, simplemente no la añade.
- 2- Añadir un botón a la aplicación anterior que me permite eliminar la cadena de texto introducida en el campo de texto de la lista de cadenas del JList. (Si la cadena introducida en el campo de texto, existe en el JList, la elimina, si no hay coincidencia, no hace nada).
- 3- Añadir un botón a la aplicación anterior que permita eliminar la cadena de texto que está actualmente seleccionada en el JList. Si no hay selección, no hace nada.
- 4- Añadir un botón a la aplicación anterior que permita eliminar de una sola vez, todas las cadenas seleccionadas en el JList (Pulsando la tecla **Ctrl** se pueden seleccionar varios elementos del JList). La clase JList tiene un método que permite obtener un objeto *List<>* con todos los elementos seleccionados.
- 5- Ejercicio 1-a, pero el Jlist en lugar de listar cadenas de texto lista objetos de tipo LocalDate. Para añadirlos al JList, en lugar de emplear un campo de texto usaremos un componente DatePicker. La aplicación no debe permitir añadir dos fechas iguales.
- 6- Crear una aplicación que contenga 2 radioButtons (radioButton1 y radioButton2) y dos botones (botón1, boton2). La aplicación funciona del siguiente modo: Al pulsar el botón 1, se seleccionará el radioButton1 y al pulsar sobre el botón 2, se selecciona el radioButton2. En caso de que, al pulsar sobre un botón su radioButton correspondiente ya esté seleccionado, la aplicación me lo indicará mediante un mensaje JOptionPane.
- 7- Crear una aplicación que muestra un dialogo JFileChooser para seleccionar un archivo. Si pulso sobre aceptar en JFileChooser, la aplicación me indicará por consola la ruta del fichero seleccionado, y, si es un directorio o es un archivo. (La clase File tiene métodos para indicar si un objeto File es fichero o directorio, así como para obtener su ruta). No es necesario crear una ventana principal, puedo lanzar el JFileChooser desde el main()).
- 8- Crear una aplicación que contiene dos labels una visible y otra oculta (lblVisible y lblNoVisible). Utilizando los eventos de tipo MouseEvent, implementar el siguiente comportamiento en la aplicación: Cuando paso el ratón encima de la lblVisible, la lblNoVisible se muestra (Visible = true) y cuando quito el ratón fuera de la lblVisible, la lblNoVisible se vuelve a ocultar (Visible = false). (Los métodos listener que debo implementar son mouseEntered() y mouseExited())

- 9- Crear una aplicación que contenga una ventana con un área de texto y un botón que al ser pulsado despliegue un cuadro de diálogo (JDialog). Este cuadro de dialogo tendrá un botón de aceptar y otro de cancelar, y un JList con 10 elementos (estos elementos pueden ser introducidos mediante código al iniciar la aplicación, y pueden ser del tipo que se quiera: String, LocalDate, o cualquier otra clase. Si se prefiere se pueden crear mediante un botón desde ese mismo cuadro de dialogo. El único requisito es que haya un JList con varios elementos en su lista.)

La aplicación debe funcionar del siguiente modo: al desplegar el cuadro de dialogo me permite seleccionar diversos elementos de esa lista (Pulsando la tecla ctrl puede clicar sobre varios elementos para seleccionarlos). Al pulsar sobre el botón aceptar se cierra el cuadro de diálogo y en el área de texto se mostrarán los datos seleccionados en el JDialog. Si pulso sobre cancelar en el cuadro de diálogo no se mostrará nada.

La finalidad de este ejercicio es implementar un método permita devolver datos desde un JDialog, del mismo modo que lo hace JFileChooser u otros tipos de diálogo. El JDialog debería tener un método que haga un return con la lista de datos seleccionada, o permitir obtenerlos de algún modo.

- 10- Crear una aplicación que contenga dos JLists (jList1 y jList2) y dos botones (> y <):



Los elementos de ambos JLists deben ser del mismo tipo, aunque puede ser cualquier tipo. Del mismo modo, pueden añadirse mediante código al iniciar la aplicación.

La aplicación debe permitir que al seleccionar elementos de JList1 y pulsar sobre el botón ">" estos pasen a JList2 y desaparezcan de JList1, y viceversa al pulsar sobre el botón "<". Se pueden seleccionar elementos individuales o varios elementos al mismo tiempo.

- 11- Crear una aplicación que contenga dos elementos: un JSlider y un JSpinner, ambos para gestionar números enteros de 0 a 100. La aplicación debe mostrar en el slider el mismo valor que hay en el spinner, y si modifico el valor desde el slider o desde el spinner, se actualizará en el otro componente. El método listener que debo implementar es el stateChanged().
- 12- Crear una aplicación que contenga una interfaz gráfica, internacionalizada completamente. La interfaz gráfica tendrá un botón con el texto "Seleccionar Archivo" y al pulsarlo desplegará un JFileChooser. Si en el JFileChooser selecciono un fichero y pulso sobre Aceptar, la aplicación mostrará un mensaje por consola con el texto: "Fichero seleccionado: ruta...", donde *ruta...* es la ruta real del fichero en mi ordenador (la clase File tiene un método para obtener la ruta del fichero). Si pulso sobre cancelar en el JFileChooser, se muestra el mensaje "Fichero no seleccionado".

La aplicación debe estar internacionalizada en 3 idiomas: español, inglés y otro cualquiera, tanto los textos que aparecen en la interfaz gráfica, como los que se muestran por consola.

Para poder ver el comportamiento de los textos de mi aplicación, basta con modificar la localización de mi aplicación a la hora de iniciarse, en el método main: `Locale.setDefault("Locale concreto");`

13- Crear una aplicación que contenga un Botón y una JLabel. Al pulsar el botón se despliega un JFileChooser. Al seleccionar un archivo de imagen en ese JFileChooser y pulsar Aceptar, la imagen se mostrará en el elemento JLabel.

14- A partir de la aplicación del ejercicio 5, añadir un botón que despliegue una gráfica JFreeChart, con la cantidad de fechas por cada año. El tipo de gráfica es indiferente. Por ejemplo:

2020-06-05, 2020-01-08, 2020-06-12
2019-11-14
2018-04-02, 2018-06-02

2020 (3 elementos)
2019 (1 elemento)
2018 (2 elementos)

Se puede construir un gráfico de sectores con 3 claves y cada una con sus valores. La finalidad es ser capaz de mostrar un gráfico, da igual cuál y cómo.