

JOptionPane

Programació. DAM

Introducción

Un diálogo es una ventana independiente también llamada subventana que pretende mostrar una información temporal al margen de la ventana de la aplicación principal.

Se utilizan cuando creamos aplicaciones visuales con Swing, pero en Greenfoot también podemos utilizarlos.

Tenemos los siguientes diálogos:

showOptionDialog, showInputDialog, showMessageDialog y showConfirmDialog.

Los mas usados son los tres últimos

Librería

- JOptionPane facilita el mostrar una ventana emergente en Java o también se usa para pedir un valor a los usuarios o informarles de algo.
- Para hacer uso de la herramienta JOptionPane se debe de importar la siguiente libreria:
- *javax.swing.JOptionPane;*

Metodos basicos de JOptionPane

- JOptionPane cuenta con estos cuatro metodos basicos para su manejo y uso aunque existen muchos mas solo haremos énfasis en los siguientes:
- `showOptionDialog();`
- `showInputDialog();`
- `showMessageDialog();`
- `showConfirmDialog();`
- Cada uno de ellos **devuelve un tipo de datos diferente**, hemos de fijarnos bien para tratar posteriormente esa información.

showOptionDialog();

- Tenemos un método *JOptionPane.showOptionDialog()* que nos muestra la ventana más configurable de todas, en ella debemos definir todos los botones que lleva. De hecho, las demás ventanas disponibles con *JOptionPane* se construyen a partir de esta. Por ello, al método debemos pasarle muchos parámetros:
- **parentComponent:** A partir de este componente, se intentará determinar cual es la ventana que debe hacer de padre del *JOptionPane*. Se puede pasar *null*, pero conviene pasar, por ejemplo, el botón desde el cual se lanza la acción que provoca que se visualice el *JOptionPane*. De esta manera, la ventana de aviso se visualizará sobre el botón y no se podrá ir detrás del mismo si hacemos click en otro sitio.

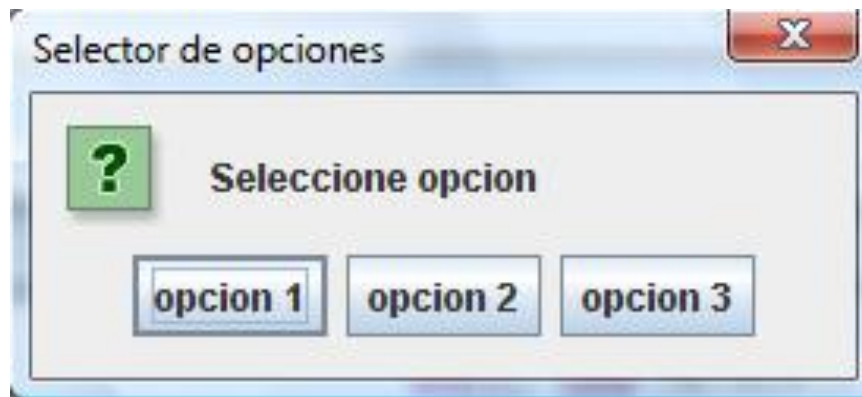
- **message:** El mensaje a mostrar, habitualmente un *String*
- **title:** El título para la ventana.
- **optionType:** Un entero indicando qué opciones queremos que tenga la ventana. Los posibles valores son las constantes definidas en *JOptionPane*: *DEFAULT_OPTION*, *YES_NO_OPTION*, *YES_NO_CANCEL_OPTION*, o *OK_CANCEL_OPTION*.

- **messageType:** Un entero para indicar qué tipo de mensaje estamos mostrando. Este tipo servirá para que se determine qué icono mostrar. Los posibles valores son constantes definidas en *JOptionPane*: *ERROR_MESSAGE*, *INFORMATION_MESSAGE*, *WARNING_MESSAGE*, *QUESTION_MESSAGE*, o *PLAIN_MESSAGE*
- **icon:** Un icono para mostrar. Si ponemos *null*, saldrá el icono adecuado según el parámetro *messageType*.
- **options:** Un array de *objects* que determinan las posibles opciones. Si los objetos son componentes visuales, aparecerán tal cual como opciones. Si son *String*, el *JOptionPane* pondrá tantos botones como *String*. Si se pasa *null*, saldrán los botones por defecto que se hayan indicado en *optionType*.

- **initialValue:** Selección por defecto. Debe ser uno de los *Object* que hayamos pasado en el parámetro *options*. Se puede pasar *null*.
- La llamada a *JOptionPane.showOptionDialog()* devuelve un entero que representa la opción que ha seleccionado el usuario. La primera de las opciones del array es la posición cero. Si se cierra la ventana con la cruz de la esquina superior derecha, el método devolverá -1.

Ejemplo

- `int seleccion = JOptionPane.showOptionDialog(unComponentePadre, "Seleccione opcion", "Selector de opciones", JOptionPane.QUESTION_MESSAGE, null, new Object[] { "opcion 1", "opcion 2", "opcion 3" }, opcion 1);`

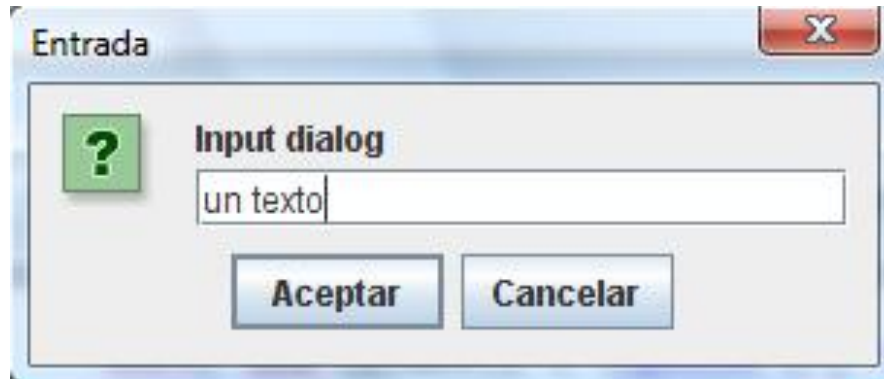


showInputDialog();

- Tenemos varios métodos *JOptionPane.showInputDialog()* y la diferencia entre ellos es que tienen más o menos parámetros, según queramos aceptar o no las opciones por defecto. Los parámetros y sus significados son muy similares a los del método *showOptionDialog()*,

Ejemplo

- String seleccion =
JOptionPane.showInputDialog(null, "Input
dialog", JOptionPane.QUESTION_MESSAGE);



showMessageDialog();

- Esta es la más sencilla de todas, sólo muestra una ventana de aviso al usuario. La ejecución se detiene hasta que el usuario cierra la ventana. Hay varios métodos con el mismo nombre y más o menos parámetros, en función de si aceptamos las opciones por defecto (icono, por ejemplo) o queremos cambiar alguna cosa.

Iconos

- Los iconos usados del JOptionPane son los siguientes y son los que recibe como parametro la mayoría de los metodos que necesitan un icono de mensaje

Icons used by JOptionPane (Java look and feel)



question



information



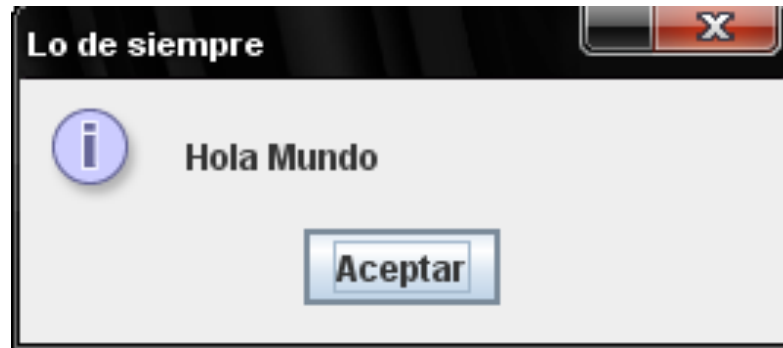
warning



error

Ejemplo

- `JOptionPane.showMessageDialog(null, "Hola Mundo", "Lo de siempre", JOptionPane.INFORMATION_MESSAGE);`



showConfirmDialog();

- Este método muestra una ventana pidiendo una confirmación al usuario y da al usuario opción de aceptar o cancelar. El método devuelve un entero indicando la respuesta del usuario. Los valores de ese entero puede ser alguna de las constantes definidas en *JOptionPane: YES_OPTION, NO_OPTION, CANCEL_OPTION, OK_OPTION, CLOSED_OPTION.*

Ejemplo

- `int confirmado = JOptionPane.showConfirmDialog(null, "¿Lo confirmas?", JOptionPane.QUESTION_MESSAGE);`

