

Práctica 2: Java Swing

Asignatura: Sistemas Interactivos

Resumen

Con unas nociones básicas sobre el IDE utilizado obtenidas en la práctica 1, se pretende desarrollar las facultades del alumno enfocadas a utilizar todas las herramientas de Netbeans y Java Swing para la construcción un interfaz, en principio simple y añadiendo complejidad en las iteraciones posteriores al diseño inicial.

1. Introducción.

En el contenido de esta práctica aprenderemos los distintos componentes gráficos que Java Swing nos permite incluir a nuestro diseño, así como el uso de las siguientes Layouts :

- FlowLayout.
- GridLayout.
- BorderLayout.
- JPanel.

No sólo se darán nociones prácticas sino también teórico-prácticas y pautas de cómo está o debe estar correctamente diseñado el interfaz para que cumpla ciertas condiciones que cualquier interfaz debe cumplir; usabilidad, claridad, concisión, modularidad, gestión de errores...etc.

Utilizaremos para el desarrollo y explicación de estas nociones el proyecto [GUIFormExamples](#).

Todo con uso de ejemplos aplicados para finalmente realizar una práctica entregable dónde se pondrá a prueba el aprendizaje y también la innovación del alumno que consistirá en el desarrollo de una calculadora gráfica.

2. Crear una ventana (JFrame)

Como en cualquier lenguaje de programación podemos construir nuestro interfaz mediante código cómo podemos observar en el ejemplo [VentanaTest.java](#).

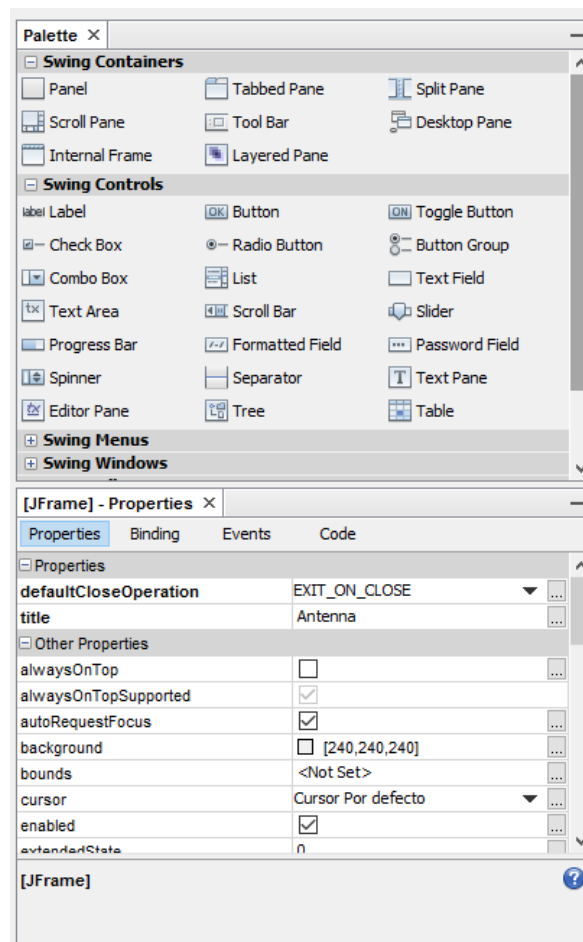
Una ventana en Java Swing viene descrita por un objeto llamado JFrame, que se crea de igual manera que otro objeto y en su constructor puede recibir diversos parámetros que podéis utilizar. En nuestro ejemplo hemos utilizado sólo el nombre de la ventana, otra opción muy común también es pasarle una configuración ya hecha para que la adopte automáticamente, aunque no se va a ver en el desarrollo de la práctica, se deja a interés del alumno.

Una vez creado JFrame podemos utilizar sus funciones para configurarlo a gusto del programador; tamaño, acciones al cierre...etc. Por defecto, todos los JFrame están ocultos, así que es importante usar la operación set ([setVisible](#)) que lo active para ser visualizado.

Pero el objetivo de esta práctica no es del todo usar este método para crear interfaces, sino utilizar las facilidades que nos da NetBeans para crear interfaces.

3. Componentes gráficos.

Junto a la pestaña Source de NetBeans, podemos encontrar Design, que nos abrirá unos nuevos paneles en el lateral como el que se muestra a continuación.



Este panel nos permitirá añadir distintos tipos de elementos gráficos que podamos necesitar. Incluso si fueran necesario más, existen plugins para aumentar las posibilidades. Esto agiliza mucho el trabajo del diseñador y programador, por lo que es una herramienta muy potente tanto en el resultado satisfactorio final del interfaz, como en términos de productividad.

Podemos encontrar botones, cajones de texto para rellenar, deslizadores, tablas, mmenús predefinidos...etc. Multitud de herramientas que se podrían utilizar según nuestras necesidades.

En el ejemplo, [Ventana.java](#) podemos observar en el código cómo sería la programación e inclusión de estos componentes gráficos en el JFrame, organizados en una clase Ventana que hereda de JFrame. Este código es similar el código que NetBeans nos genera automáticamente al utilizar sus herramientas de diseño.

4. Constructor de interfaces de Netbeans.

Ahora vamos a ver cómo creamos un interfaz que se pueda configurar mediante las herramientas que Netbeans nos proporciona. Probaremos a crear un nuevo fichero dentro del proyecto que estamos utilizando dándole botón derecho al package y a New. Elegiremos New JFrame Form. Esto ya nos habilitará una nueva aplicación en el proyecto, que podremos diseñar con uso del constructor.

Llamaremos a esta aplicación [Prueba.java](#) y utilizando la paleta de componentes, crearemos algo parecido a lo siguiente:

[ContactEditor.java](#), [Atenna.java](#) y [Find.java](#) son ejemplos más complejos de este ejemplo que vamos a desarrollar poco a poco, por lo que podrán y deberán tomarse como referencia en caso de duda tras la explicación.

4.1. Añadir eventos a los botones.

Para dar una función a cada botón hay que realizar lo siguiente. En la parte inferior izquierda de NetBeans, tenemos una lista de los componentes de nuestro interfaz. Podemos ir a sus propiedades y editar sus eventos (en el click del ratón o lo que queramos).

También directamente sobre el interfaz que hemos hecho, si le damos botón derecho sobre el componente, podemos editar los eventos directamente. Si pulsamos sobre él, nos lleva a la parte de código dónde se controla la lógica del evento. Aquí es dónde programaremos que nos imprima por consola lo que hemos escrito en el cuadro de texto. Recordamos que la salida por consola se realiza mediante [System.out.println](#). Para utilizar el texto del cuadro de diálogo hay que hacer uso del observador [getText](#).

Para añadirle aún más complejidad ahora editaremos el evento y en lugar de imprimir en pantalla, mostraremos el valor del cuadro de texto, en un mensaje de diálogo con `ShowMessageDialog`.

Las posibilidades son todas las imaginables mientras se sepan programar, así que se deja a interés del alumno que aumente los conocimientos adquiridos sobre eventos y su alcance.

Podemos mezclar distintos contenidos en una misma ventana haciendo buen uso de las ventanas con tabs, utilizando los contenedores `Tabbed Panel`. Si lo incluimos, al principio sólo tendremos el contenedor, pero si volvemos a arrastrar, iremos añadiendo tabs, cada uno con su contenido. Muy útil para no realizar un interfaz con muchísimas ventanas si necesitas mucha y diversa información.

5. Entregable.

Con lo aprendido durante esta práctica se deberá desarrollar una calculadora funcional y con un interfaz que cumpla las características básicas que un sistema interactivo debe tener para que sea usado por cualquier persona; esto implica que sea intuitivo, usable, universal...etc. La entrega realizada por el alumno debe cumplir como mínimo los siguientes objetivos:

- Operaciones básicas de suma, resta, multiplicación y división.
- Realizar funcionalidad de la calculadora de manera continua. Es decir, que no sea necesario reiniciar la aplicación para hacer una nueva operación.
- Que si pudiera existir algún error, se notificara al usuario.
- Mostrar el resultado de la manera que el programador quiera, pero sin usar consola.

Para conseguir calificación extra y con objetivo de aumentar vuestra práctica, pueden añadirse contenidos que no sean contemplados en los objetivos, serán valorados positivamente. Por ejemplo, podría tratarse de una calculadora funcional para un caso específico (infantil).

La entrega se realizará mediante una tarea en la plataforma Moodle de la asignatura con fecha límite XX-XX-2016. El nombre del alumno debe estar reflejado en el comprimido que contenga el Proyecto que recoja ambas actividades y también debe estar reflejado en el código (suele hacerlo automáticamente NetBeans). Se valorará negativamente copias y retrasos en la entrega.