

## **Práctica 3: Java Swing 2**

*Asignatura: Sistemas Interactivos*

### **Resumen**

Habiendo aprendido lo básicos sobre Java Swing, se buscará afinar las técnicas en encapsulamiento para una mejor modulación de las aplicaciones creadas por el alumno. Además, en esta práctica, se profundizará en el uso de componentes gráficos.

#### **1. Introducción.**

Ya hemos visto casi todas las características que Java Swing nos permite dar a nuestras aplicaciones. Que si recordamos de la Práctica 1 eran las siguientes:

- Herencia.
- Encapsulación.
- Polimorfismo.
- Sobrecarga de operadores.
- Paquetes Java y convenciones de nombre de archivos y paquetes.

Sin embargo, no hemos visto la encapsulación hasta el momento y es lo que trataremos. La encapsulación dependerá del programador y tratará de crear componentes gráficos básicos y añadirlos a nuestra paleta en Java Swing para posteriormente usarlos de manera automática con una serie de métodos programados, si fueran necesarios (getter, setter, constructores...etc).

## 2. Ejemplo básico de encapsulación

Para comprender la esencia de la encapsulación y sus ventajas, lo veremos rápidamente en un sencillo ejemplo. Para ello crearemos un nuevo proyecto con un JPanel que tenga dos botones.

En la pestaña Source/Código, podremos añadir métodos o atributos a la clase. Por lo que crearemos dos atributos `textbutton1` y `textbutton2`. También pondremos sus correspondientes getter y setter, y si se desea, también el constructor.

Crearemos en el proyecto una carpeta llamada icon, para copiar los iconos que se asociaran a este componente. En moodle tenemos un .Java llamado BotonesBeanInfo.java que nos pondrá los iconos automáticamente como predeterminados una vez lo añadamos a nuestra paleta. Añadiremos este .Java a nuestro proyecto y haremos Clean&Build.

Al hacer esto, generaremos una nueva carpeta en nuestro directorio del proyecto llamada `dist`, donde se ha generado el ejecutable .jar.

Ahora procederemos a añadir este componente a la paleta de componentes, dando click derecho sobre la misma y usando el Palette Manager o Administrador de paleta. Luego Tools Palette y seleccionamos el jar del componente. Podemos darle nombre y añadirlo a la paleta. Una vez hecho esto refrescamos la paleta con botón derecho y Refresh Palette y deberá aparecer el componente en la sección dónde añadimos el mismo.

Si ahora creamos una nueva aplicación y pinchamos y arrastramos dicho componente a la interfaz, se añadirá tal cómo lo creamos y si por código usamos los getter y setter podremos leer el texto del botón y cambiarlo por otro si quisiéramos, sin necesidad de editar el componente a mano y volverlo a actualizar.

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.

### 3. Entregable.

Con lo aprendido durante esta práctica se deberá desarrollar un componente gráfico de una rueda que gire con el uso de unos botones, utilizando encapsulación. La entrega realizada por el alumno debe cumplir como mínimo los siguientes objetivos:

- Control de errores.
- Encapsulación
- Limpieza de código.

Como ayuda para añadir el componente gráfico, debe usarse un Label, y la propiedad "Icon" nos permitirá añadirle una imagen. Podemos cambiar en tiempo de ejecución dicha imagen si añadimos todas las imágenes de la rueda a nuestro proyecto y vamos cogiéndolos con el paquete con uso de `getClass().getResource("/iconos/boton1.jpg")` y después seteando el nuevo icono del Label con el setter correspondiente.

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.