

Introducción a las clases, los objetos, los métodos y las cadenas

Parte 2

Ejemplo de GUI y gráficos: uso de cuadros de diálogo

Este ejemplo práctico está diseñado para quienes desean empezar a conocer las poderosas herramientas de Java para crear **interfaces gráficas de usuario (GUI)** y gráficos compatibles. Este ejemplo práctico incluye la tecnología **Swing** consolidada de Java, que sigue siendo popular.

En las primeras secciones, usted creará sus primeras aplicaciones gráficas. En las secciones posteriores, utilizará los conceptos de programación orientada a objetos.

Cómo mostrar texto en un cuadro de diálogo

Los programas que hemos presentado hasta ahora muestran su salida en la ventana de comandos. Muchas aplicaciones utilizan ventanas, o cuadros de diálogo (también llamados diálogos) para mostrar la salida. Por ejemplo, los navegadores Web como Chrome, Firefox, Internet Explorer, Safari y Opera muestran las páginas Web en sus propias ventanas. Los programas de correo electrónico le permiten escribir y leer mensajes en una ventana. Por lo general, los cuadros de diálogo son ventanas en las que los programas muestran mensajes importantes a los usuarios. La clase `JOptionPane` cuenta con cuadros de diálogo prefabricados, los cuales permiten a los programas mostrar ventanas que contengan mensajes; a dichas ventanas se les conoce como diálogos de mensaje. La figura 3.12 muestra el objeto `String` "Bienvenido a Java" en un diálogo de mensaje.

```
// Fig. 3.12: Dialogo1.java
// Uso de JOptionPane para mostrar varias líneas en un cuadro de diálogo.
import javax.swing.JOptionPane;

public class Dialogo1
{
    public static void main(String[] args)
    {
        // muestra un diálogo con un mensaje
        JOptionPane.showMessageDialog(null, "Bienvenido a Java");
    }
} // fin de la clase Dialogo1
```



Fig. 3.12 Uso de **JOptionPane** para mostrar varias líneas en un cuadro de diálogo

El método `static showMessageDialog` de la clase `JOptionPane`

La línea 3 indica que el programa utiliza la clase `JOptionPane` del paquete `javax.swing`. Este paquete contiene muchas clases que le ayudan a crear interfaces gráficas de usuario (GUI). Los componentes de la GUI facilitan la entrada de datos al usuario del programa, además de la presentación de los datos de salida. La línea 10 llama al método `showMessageDialog` de `JOptionPane` para mostrar un cuadro de diálogo que contiene un mensaje. El método requiere dos argumentos. El primero ayuda a la aplicación de Java a determinar en dónde colocar el cuadro de diálogo. Por lo general, un diálogo se muestra desde una aplicación de GUI con su propia ventana. El primer argumento hace referencia a esa ventana (conocida como ventana padre) y hace que el diálogo aparezca centrado sobre la ventana de la aplicación. Si el primer argumento es null, el cuadro de diálogo aparece en el centro de la pantalla de la computadora. El segundo argumento es el objeto `String` a mostrar en el cuadro de diálogo.

Introducción de los métodos `static`

El método `showMessageDialog` de la clase `JOptionPane` es lo que llamamos un método `static`. A menudo, dichos métodos definen las tareas que se utilizan con frecuencia. Por ejemplo, muchos programas muestran cuadros de diálogo, y el código para hacer esto es el mismo siempre. En vez de “reinventar la rueda” y crear código para realizar esta tarea, los diseñadores de la clase `JOptionPane` declararon un método `static` que realiza esta tarea por usted. La llamada a un método `static` se realiza mediante el uso del nombre de su clase, seguido de un punto (.) y del nombre del método, como en `NombreClase.nombreMétodo(argumentos)`.

Observe que no tiene que crear un objeto de la clase `JOptionPane` para usar su método `static` llamado `showMessageDialog`.

Introducción de texto en un cuadro de diálogo

La figura 3.13 utiliza otro cuadro de diálogo `JOptionPane` predefinido, conocido como diálogo de entrada, el cual permite al usuario introducir datos en un programa. El programa pide el nombre del usuario y responde con un diálogo de mensaje que contiene un saludo y el nombre introducido por el usuario.

```
// Fig. 3.13: DialogoNombre.java
// Entrada básica con un cuadro de diálogo.
import javax.swing.JOptionPane;

public class DialogoNombre
{
    public static void main(String[] args)
    {
        // pide al usuario que escriba su nombre
        String nombre = JOptionPane.showInputDialog("Cual es su nombre?");

        // crea el mensaje
```

```
String mensaje =  
    String.format("Bienvenido, %s, a la programacion en Java!", nombre);  
  
    // muestra el mensaje para dar la bienvenida al usuario por su nombre  
    JOptionPane.showMessageDialog(null, mensaje);  
} // fin de main  
} // fin de la clase DialogoNombre
```



Fig. 3.13 *Cómo obtener la entrada del usuario mediante un cuadro de diálogo*

El método static `showInputDialog` de la clase `JOptionPane`

La línea 10 utiliza el método `showInputDialog` de `JOptionPane` para mostrar un diálogo de entrada que contiene un indicador (*prompt*) y un campo (conocido como *campo de texto*), en el cual el usuario puede escribir texto. El argumento del método `showInputDialog` es el indicador que muestra lo que el usuario debe escribir. El usuario escribe caracteres en el campo de texto, y después hace clic en el botón **Aceptar** u oprime la tecla **Intro** para devolver el objeto `String` al programa. El método `showInputDialog` devuelve un objeto `String` que contiene los caracteres escritos por el usuario. Almacenamos el objeto `String` en la variable `nombre`. Si oprime el botón **Cancelar** en el cuadro de diálogo u oprime **Esc**, el método devuelve `null` y el programa muestra la palabra `null` como el nombre.

El método static `format` de la clase `String`

Las líneas 13 y 14 utilizan el método static `String` llamado `format` para devolver un objeto `String` que contiene un saludo con el nombre del usuario. El método `format` es similar al método `System.out.printf`, excepto que `format` devuelve el objeto `String` con formato, en vez de mostrarlo en una ventana de comandos. La línea 17 muestra el saludo en un cuadro de diálogo de mensaje, como hicimos en la figura 3.12.