



TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO TECNOLÓGICO DEL VALLE DE OAXACA

Ingeniería Informática

ADMINISTRACION Y ORGANIZACION DE DATOS

Tema 4: Sentencias de entrada y salida de datos

DOCENTE: Benedicto Ramírez Santiago

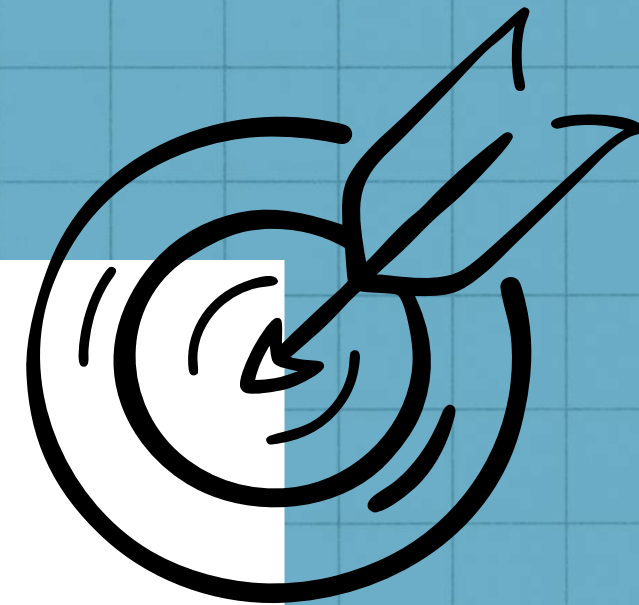
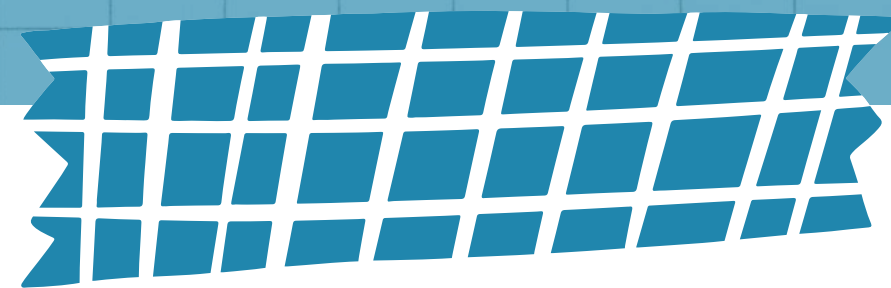
ALUMNOS:

- DIEGO EMILIANO PEÑA ROJAS
- AXEL TADEO VASQUEZ LOPEZ

4^ºA

Introducción

En la programación, la entrada y salida de datos es fundamental para que los usuarios puedan interactuar con un programa y recibir resultados. En Java, existen varias formas de hacerlo: mediante consola usando Scanner, mediante ventanas emergentes con JOptionPane y mediante formularios gráficos con JFrame. Cada método tiene su propio nivel de complejidad y utilidad: la consola es simple y rápida para practicar, JOptionPane ofrece interacciones más visuales sin necesidad de construir un formulario completo, y JFrame permite crear interfaces gráficas completas y más profesionales, donde se pueden usar botones, etiquetas y campos de texto.



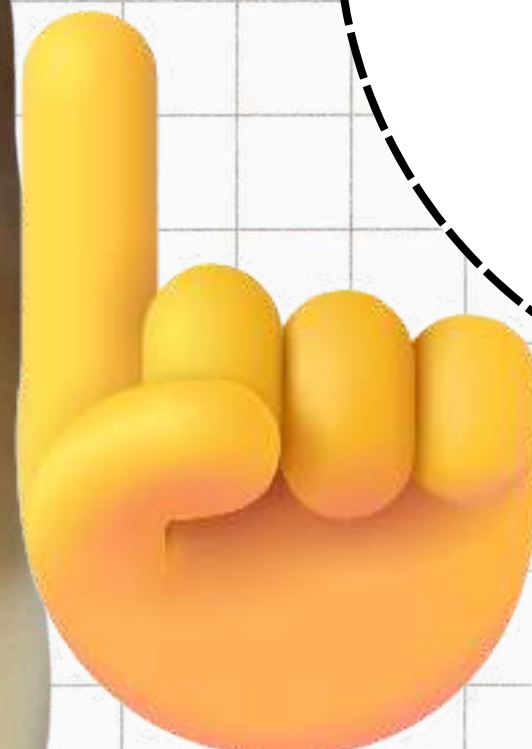
MODDO CONSOLA



¿Qué es Scanner?

```
import java.util.Scanner;
```

En Java, Scanner es una clase que permite leer datos que el usuario escribe en la consola. Es parte de la biblioteca estándar de Java (`java.util.Scanner`) y se usa cuando queremos recibir información sin necesidad de ventanas gráficas.





Sentencias de Entrada y Salida de Datos en Modo Consola



¿QUÉ SON?

Son instrucciones que permiten la interacción entre el usuario y el programa mediante texto. Puede recibir datos que el usuario ingresa desde el teclado (entrada) y también mostrar información o resultados en pantalla mediante la consola (salida).

En Java se utilizan principalmente:

- **Scanner:** Para entrada de datos.
- **System.out.println():** Para salida de datos.

Programa en modo consola

CODIGO

Este programa permite ingresar dos números desde la consola y realiza las cuatro operaciones básicas: suma, resta, multiplicación y división, mostrando los resultados en pantalla.

```
package operacionesbasicas;
import java.util.Scanner;
public class OperacionesBasicas {
    public static void main(String[] args) {
        Scanner objEs = new Scanner(System.in);

        // Entrada de datos
        System.out.print("Ingrese el primer número: ");
        double num1 = objEs.nextDouble();

        System.out.print("Ingrese el segundo número: ");
        double num2 = objEs.nextDouble();

        // Operaciones
        double suma = num1 + num2;
        double resta = num1 - num2;
        double multiplicacion = num1 * num2;

        System.out.println("\n--- RESULTADOS ---");
        System.out.println("Suma: " + suma);
        System.out.println("Resta: " + resta);
        System.out.println("Multiplicación: " + multiplicacion);

        // Validación para evitar división entre cero
        if (num2 != 0) {
            double division = num1 / num2;
            System.out.println("División: " + division);
        } else {
            System.out.println("No se puede dividir entre cero.");
        }

        objEs.close();
    }
}
```

ENTRADA DE DATOS

run:

Ingrese el primer número: 2

Ingrese el segundo número: 2

--- RESULTADOS ---

Suma: 4.0

Resta: 0.0

Multiplicación: 4.0

División: 1.0

BUILD SUCCESSFUL (total time: 6 minutes 35 seconds)

SALIDA DE DATOS

Usamos Scanner para leer datos desde la consola. Primero, creamos un objeto objEs de la clase Scanner, y con los métodos nextDouble() pedimos al usuario que ingrese dos números

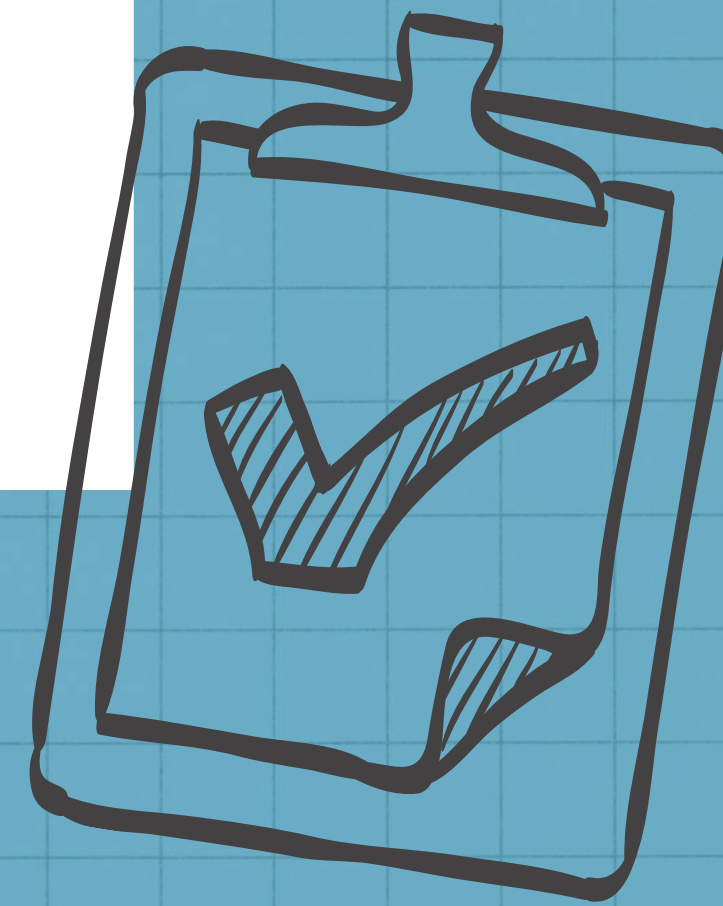
“Después del proceso usamos **System.out.println()** para mostrar los resultados en la consola. Al terminar, cerramos el Scanner con **objEs.close()**, lo cual es buena práctica para liberar recursos”



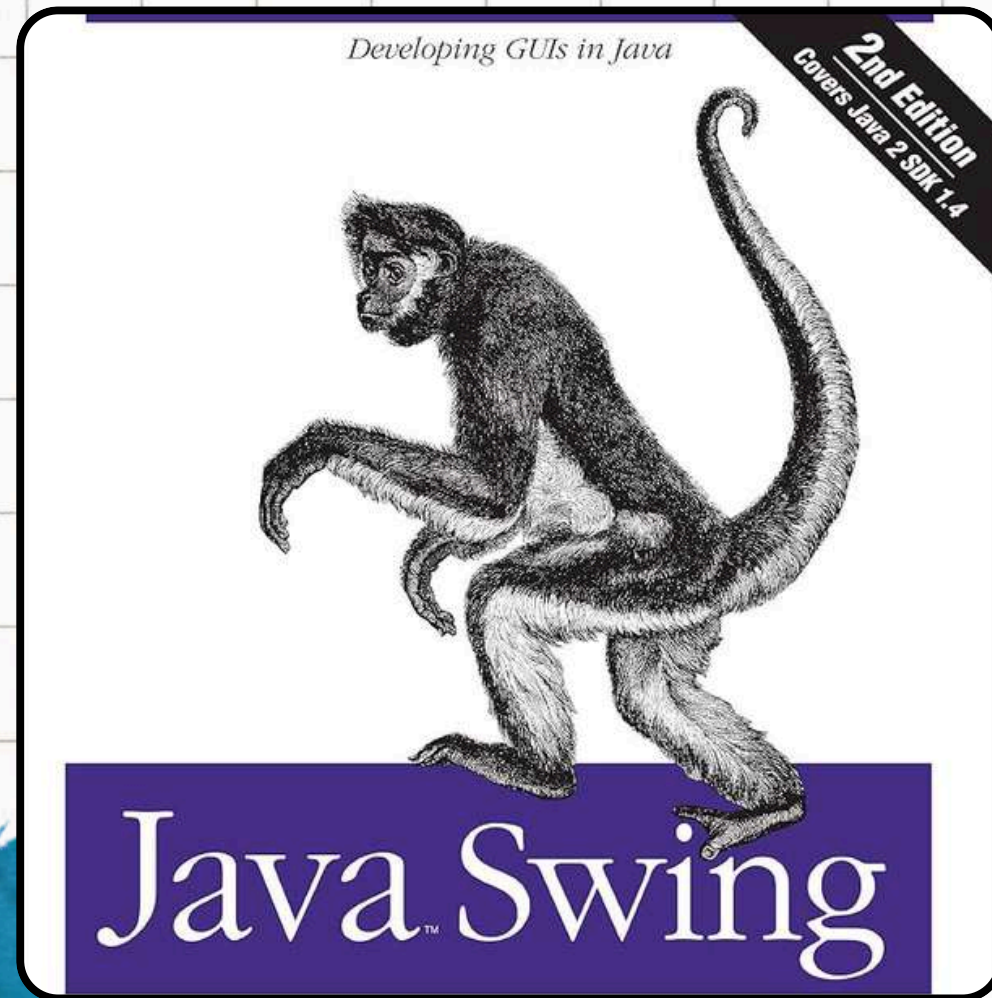


• JOptionPane •

(Entrada/Salida)




```
import javax.swing.JOptionPane;
```



¿Qué es JOptionPane?

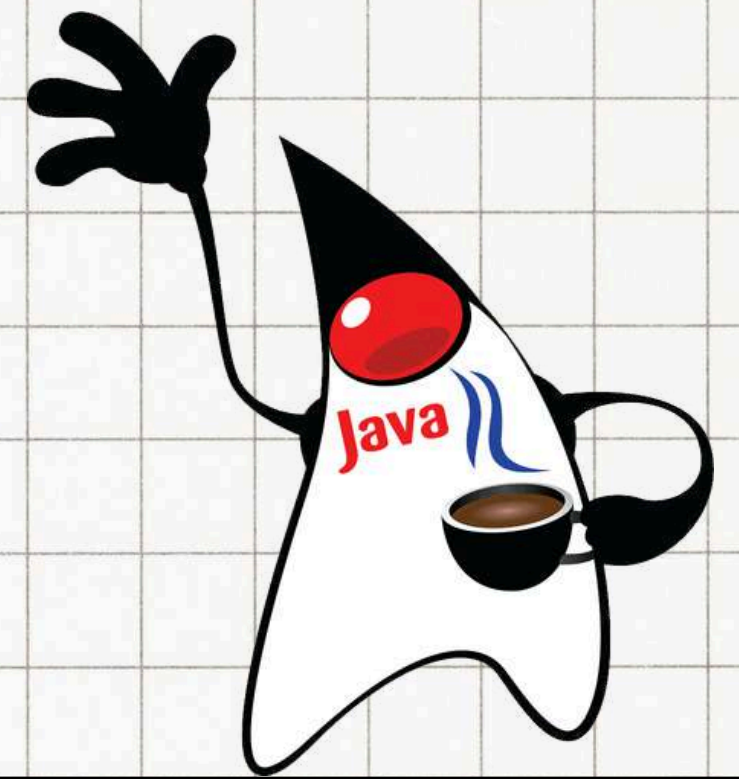
Es una clase de la biblioteca Swing de Java que permite mostrar cuadros de diálogo para interactuar con el usuario sin crear una ventana completa.

Sirve para realizar entrada y salida rápida de datos, mostrando mensajes y pidiendo información mediante ventanas emergentes.

Sentencia de Entrada de datos

showInputDialog()

“Pide datos al usuario”



```
showInputDialog(Object message)
showInputDialog(Component parentComponent, Object message)
showInputDialog(Object message, Object initialSelectionValue)
showInputDialog(Component parentComponent, Object message, Object initialSelectionValue)
showInputDialog(Component parentComponent, Object message, String title, int messageType)
showInputDialog(Component parentComponent, Object message, String title, int messageType, Icon icon, Object[] selectionValues, Object initialSelectionValue)
```

- Muestra una ventana emergente que pide al usuario que escriba un dato.
- Siempre devuelve un String, aunque el usuario escriba números (por eso después hay que convertirlos a int, float o double).
- Puede incluir texto de instrucción, título, ícono y hasta una lista de opciones en versiones avanzadas.



Sentencia de Salida de datos

showMessageDialog()

“Muestra mensajes o resultados”

```
showMessageDialog(Component parentComponent, Object message)  
showMessageDialog(Component parentComponent, Object message, String title, int messageType)  
showMessageDialog(Component parentComponent, Object message, String title, int messageType, Icon icon)
```



- Muestra una ventana emergente para informar o mostrar resultados al usuario.
- No devuelve ningún dato, solo sirve para mostrar texto o mensajes.
- Puede personalizarse con título e ícono según el tipo de mensaje (información, advertencia, error, etc.).

Parámetros principales de `showInputDialog` y `showMessageDialog`

`(Component parentComponent, Object message, String title, int messageType)`

Es la ventana padre sobre la que aparecerá el diálogo

- `null` : aparece centrado en la pantalla.
- Un `JFrame` o `JPanel` aparece centrado sobre esa ventana.

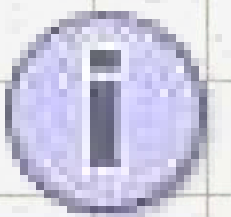
Es el texto que ve el usuario, la instrucción o pregunta.

Ejemplo: "Ingresa el primer número"

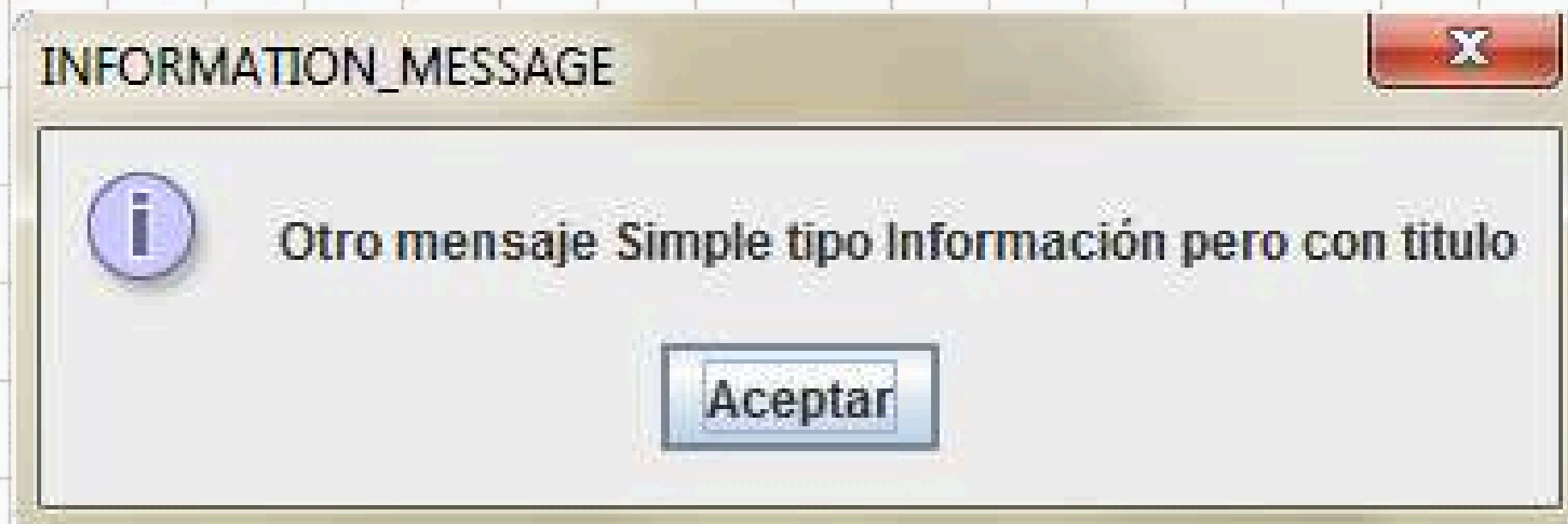
Es el título de la ventana emergente.

Ejemplo: "Datos del usuario"

Define el tipo de ícono que aparece en el cuadro de diálogo.



TIPOS DE MENSAJES/ICONOS EN EL SHOW



Ejemplo Usando JOptionPane para entrada y salida

```
package entradasalidadatos;

import javax.swing.JOptionPane;

public class EntradaSalidaDatos {

    public static void main(String[] args) {
        String num1 = JOptionPane.showInputDialog("Ingresa el primer número:");
        // String edad = JOptionPane.showInputDialog(null, "Ingresa tu edad:", "Datos del usuario", JOptionPane.QUESTION_MESSAGE);
        // Object[] opciones = { "Suma", "Resta", "Multiplicación", "División" };
        // Object operacion = JOptionPane.showInputDialog( null, "Elige una operación:", "Menú de Operaciones",
        //      JOptionPane.QUESTION_MESSAGE, null, opciones, opciones[0]);
        String num2 = JOptionPane.showInputDialog("Ingresa el segundo número:");
        int a = Integer.parseInt(num1);
        int b = Integer.parseInt(num2);
        int suma = a + b;
        int resta = a - b;
        JOptionPane.showMessageDialog(null,
            "Resultados:\n" +
            "Suma: " + suma + "\n" +
            "Resta: " + resta,
            "Operaciones con JOptionPane",
            JOptionPane.INFORMATION_MESSAGE
        );
    }
}
```


Input

?

Ingresa el primer número:

20

OK

Cancel

Input

?

Ingresa el segundo número:

12

OK

Cancel

Operaciones con JOptionPane

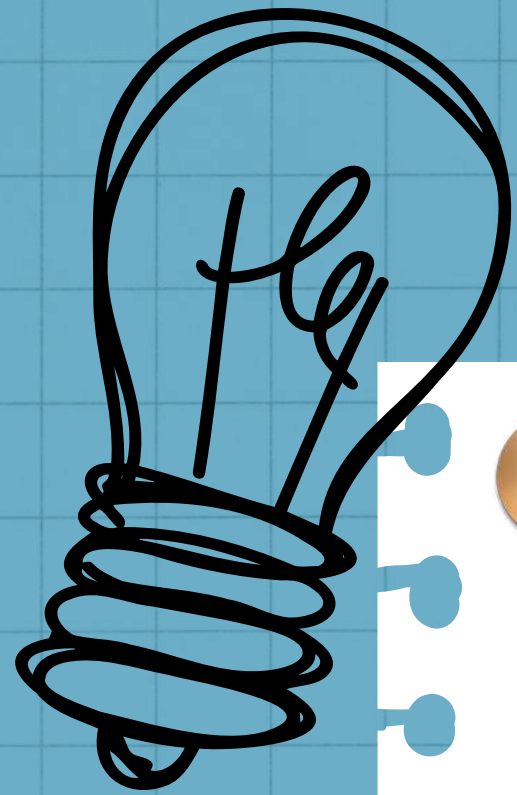
i

Resultados:

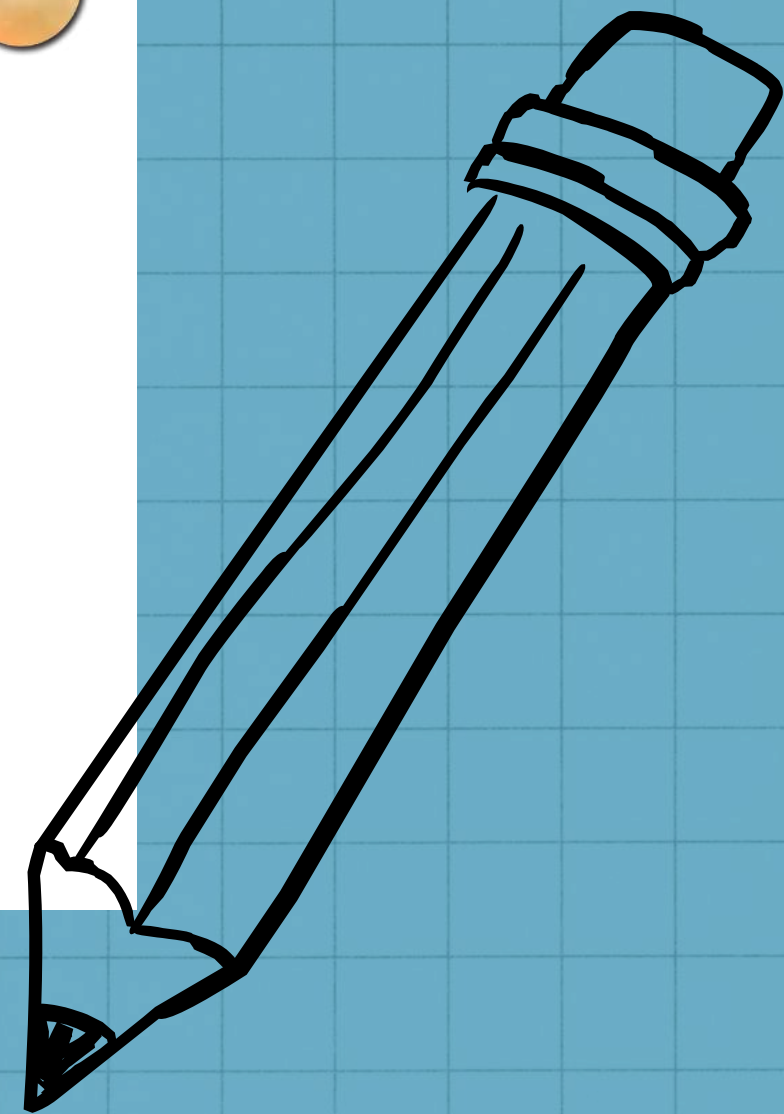
Suma: 32

Resta: -8

OK



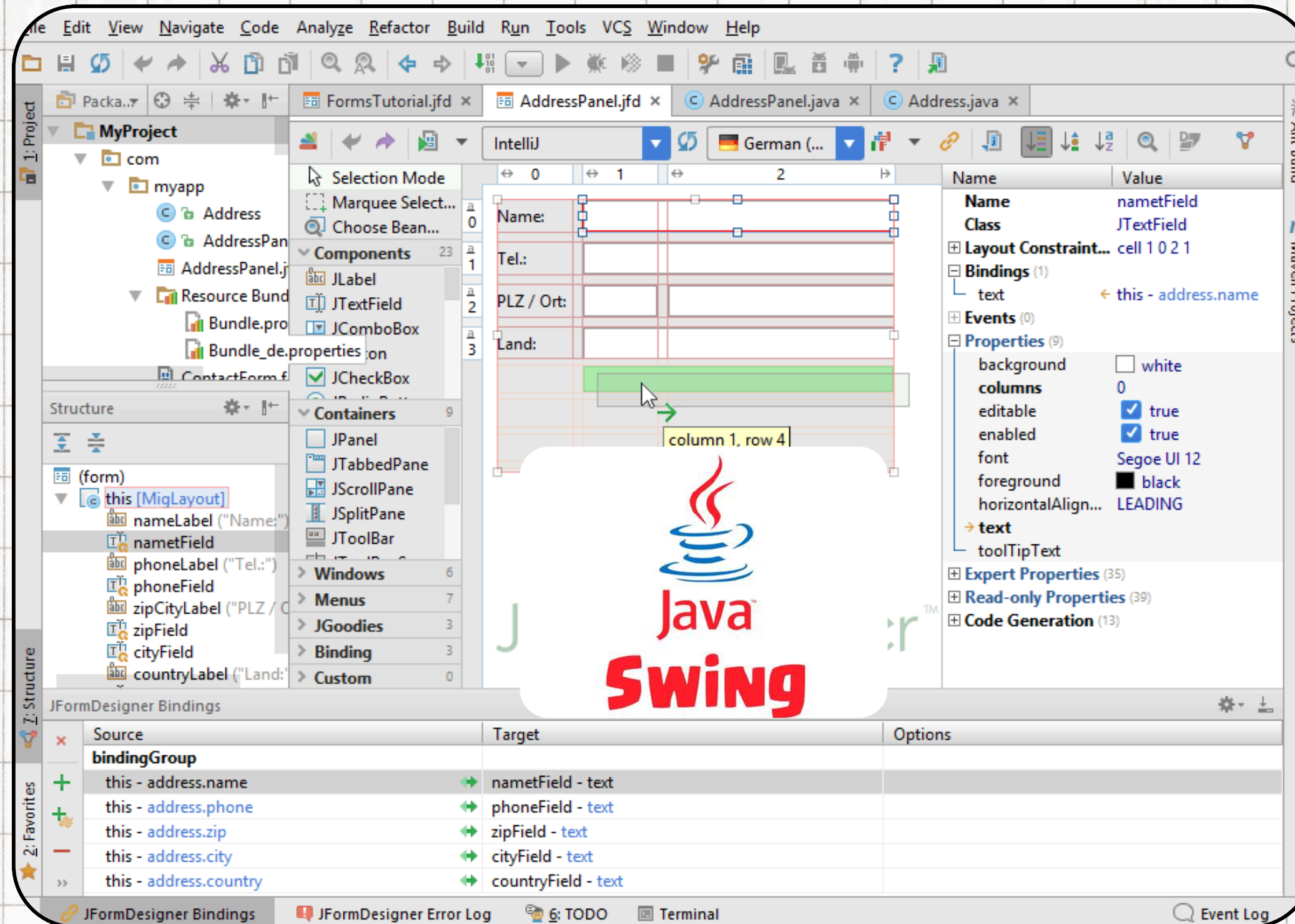
Entrada y salida mediante JFrame



¿Qué Java Swing?

Java Swing es una biblioteca de Java que permite crear interfaces gráficas de usuario (GUI), es decir, ventanas, botones, cuadros de texto, etiquetas y todos los elementos visuales con los que el usuario puede interactuar. Swing forma parte de Java y no necesita instalaciones extra; viene incluido en el propio lenguaje.

Se utiliza cuando queremos crear programas más visuales, completos y amigables, en lugar de trabajar solo con consola o ventanas emergentes básicas como JOptionPane.




¿QUÉ ES UN FORMULARIO EN JAVA SWING?

Un formulario es una ventana (JFrame) que contiene componentes gráficos para interactuar con el usuario, como:

Swing Controls

 Label

 Check Box

 Combo Box

 Text Area

 Button

 Radio Button

 List

 Scroll Bar

 Toggle Button

 Button Group

 Text Field

 Slider

Swing Containers

 Panel

 Tool Bar

 Tabbed Pane

 Desktop Pane

 Split Pane

 Internal Frame

 Scroll Pane

 Layered Pane

Ejemplo un JFrame mediante componentes para entrada y salida

SUMAR Y RESTAR DOS NUMEROS

Num1:

Num2:

Resultado

SUMAR

RESTAR

SUMAR Y RESTAR DOS NUMEROS

Num1:

Num2:

Resultado

SUMAR

RESTAR

```
private void btnSumarActionPerformed(java.awt.event.ActionEvent evt) {  
    float n1 = Float.parseFloat(Num1.getText());  
    float n2 = Float.parseFloat(Num2.getText());  
  
    float resultado = n1 + n2;  
    String r= Float.toString(resultado);  
  
    RESULTADO.setText("La suma es: "+r);  
    // TODO add your handling code here:  
}  
  
private void btnRestarActionPerformed(java.awt.event.ActionEvent evt) {  
    float n1 = Float.parseFloat(Num1.getText());  
    float n2 = Float.parseFloat(Num2.getText());  
  
    float resultado = n1 - n2;  
    String r= Float.toString(resultado);  
  
    RESULTADO.setText("La resta es: "+r);  
}
```

SUMAR Y RESTAR DOS NUMEROS

Num1:

Num2:

Resultado

SUMAR

RESTAR

Entrada de datos:
se captura desde los
JTextField.

Procesamiento:
se realiza dentro del
método
ActionPerformed de
cada botón.

Salida de datos:
se muestra en el
JTextField
RESULTADO.

Conclusión

En resumen, las tres formas de entrada y salida de datos tienen propósitos diferentes. El modo consola usando Scanner es la opción más básica, ideal para practicar lógica y resolver ejercicios sencillos directamente en texto. JOptionPane ofrece una alternativa más visual mediante ventanas emergentes, perfecta cuando solo necesitamos pedir uno o dos datos de forma rápida sin construir una interfaz completa. Finalmente, los formularios con JFrame permiten crear aplicaciones gráficas más profesionales y completas, donde podemos usar botones, etiquetas y cuadros de texto para una interacción más cómoda y organizada. En conclusión, cada método tiene su propio nivel de complejidad y utilidad: Scanner es simple, JOptionPane es práctico y visual, y JFrame es el más completo y flexible.

Bibliografía

- Cápsulas Developer. (2024, 20 noviembre). Enviar y recibir datos entre formularios | Java Swing [Vídeo]. YouTube. <https://www.youtube.com/watch?v=nJmcjqLYiUE>
- Cristian Henao. (2016, 26 enero). ¿ Que es Java Swing ? [Vídeo]. YouTube. <https://www.youtube.com/watch?v=IUHd-UOxTos>
- HunterTech. (2022, 5 noviembre). ¿Cómo utilizar el ShowInputDialog? | Java [Vídeo]. YouTube. <https://www.youtube.com/watch?v=qsOUk5EHtww>
- Perfil, V. (2017, 29 septiembre). Entrada Y Salida De Datos De Datos(JOptionPane). Edu4Java. http://todosobrejavanet.blogspot.com/p/blog-page_39.html
- Prezi, J. Á. O. (s. f.). 6.3 Entrada y Salida de Datos. prezi.com. <https://prezi.com/aoldthw2z5w0/63-entrada-y-salida-de-datos/>
- Programación ATS. (2017a, febrero 6). 8. Programación en Java || Introducción || Entrada y salida de datos por consola [Vídeo]. YouTube. <https://www.youtube.com/watch?v=taSF09mPlcs>
- Programación ATS. (2017b, febrero 7). 9. Programación en Java || Introducción || Entrada y salida de datos con JOptionPane [Vídeo]. YouTube. <https://www.youtube.com/watch?v=VFAG5uedkm4>
- Throphic. (2025, 3 septiembre). Funciones de entrada y salida en Java. DATA SCIENCE. <https://dat-science.com/funciones-de-entrada-y-salida-en-java/>