

Entrada/Salida datos por consola

PROGRAMACIÓN JAVA



ENTRADA DE DATOS DESDE TECLADO

CLASE SCANNER

- ❑ Pertenece al paquete **java.util.Scanner**
- ❑ Crear objeto de la clase Scanner:
Scanner entrada = new Scanner (System.in);
- ❑ Utilizar métodos de la clase Scanner para pedir el tipo de dato que nos interese:
 - next: pedir un String (1 token)
 - nextLine : pedir un String(hasta \n)
 - nextDouble() : Dato numérico de doble precisión
 - nextFloat() : Dato numérico de simple precisión
 - nextInt() : Entrada de dato numérico de tipo entero
 - nextBoolean() : Entrada de dato lógico. (true/false)



ENTRADA DE DATOS DESDE TECLADO

System.in.read()

□ Introducción de un carácter por teclado

```
char <variable>;
```

```
<variable> = (char) System.in.read();
```



ENTRADA DE DATOS DESDE TECLADO

CLASE JOptionPane

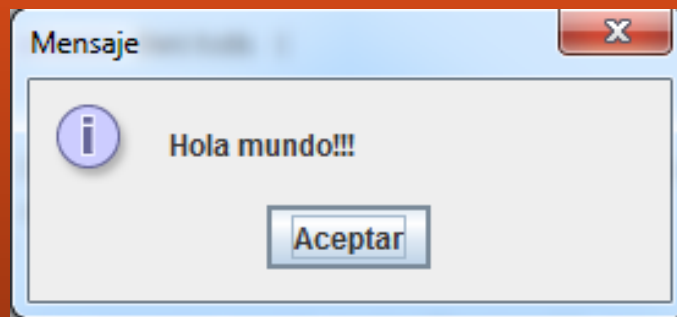
- ❑ Se usa para crear cuadros de dialogo
- ❑ Pertenece al paquete **javax.swing.JOptionPane**
- ❑ Es de tipo static, por tanto no necesitamos crear un objeto de la clase para utilizarla, es un objeto de clase
- ❑ Tipos de cuadros de dialogo:
 - **showConfirmDialog ...:**
 - **showInputDialog:** sirve para introducir datos
 - **showMessageDialog:** sirve para mostrar información (reemplaza a print())
 - **showOptionDialog...;**



ENTRADA DE DATOS DESDE TECLADO

showMessageDialog

- ❑ Sirve para mostrar información (reemplaza a print())
- ❑ `JOptionPane.showMessageDialog(frame, mensaje);`
- ❑ Ejemplo : `JOptionPane.showMessageDialog(null, "Hola mundo!!!");`

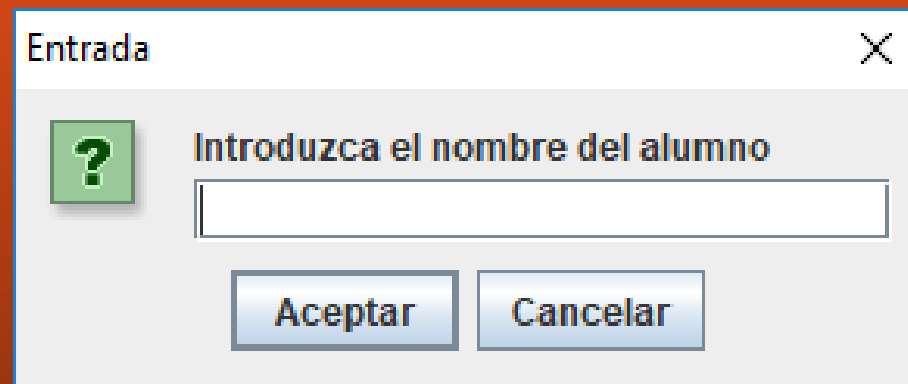




ENTRADA DE DATOS DESDE TECLADO

showInputDialog

- ❑ Sirve para introducir datos.
- ❑ Solo captura String.
- ❑ `JOptionPane.showInputDialog(mensaje);`
- ❑ Ejemplo : `JOptionPane.showInputDialog("Introduzca el nombre del alumno");`





ENTRADA DE DATOS DESDE TECLADO

CLASE JOptionPane

❑ Convertir String al tipo que necesitemos usando los métodos parse de las clases Integer, Double,.....

Ejemplos :

```
int i = Integer.parseInt("2");
```

```
double d = Double.parseDouble("4,5");
```



ENTRADA DE DATOS DESDE TECLADO

clase `BufferedReader`

- ❑ Introducción de un String por teclado
- ❑ Utilizaremos la clase `java.io.*`
- ❑ Instanciaremos un objeto de la clase `BufferedReader` (revisar en la API)

```
BufferedReader <nombre_objeto> =new BufferedReader(new  
InputStreamReader(System.in));
```

```
<variable>=<nombre_objeto>.readLine();
```




SALIDA DE DATOS

Print y Println

o **Visualización de datos.** Para mostrar datos en pantalla o visualizar los resultados. Podemos utilizar:

System.out.print(".....");	sin salto de línea
System.out.println(".....");	con salto de línea

Si queremos concatenar varios elementos en la salida empleamos el signo +

System.out.println("literal "+ variable +"literal"+...);



SALIDA DE DATOS

Print y Println

o **Visualización de datos.** Para mostrar datos en pantalla o visualizar los resultados. Podemos utilizar:

System.out.print(".....");	sin salto de línea
System.out.println(".....");	con salto de línea

Si queremos concatenar varios elementos en la salida empleamos el signo +

System.out.println("literal "+ variable +"literal"+...);



SALIDA DE DATOS

Printf

Printf: nos permite generar una salida en consola con un formato específico.

Printf (“formato”,objeto/variable/valor);

Ejemplo :

```
double precio=5.632
```

```
System.out.printf(“%4.2f”,precio);
```



SALIDA DE DATOS

Printf

`%[posición_dato$][indicador_de_formato][ancho][.precision]carácter_de_conversión`

Los elementos entre corchetes son opcionales.

posición_dato: indica la posición del dato sobre el que se va aplicar el formato. El primero por la izquierda ocupa la posición 1.

ancho: Indica el tamaño mínimo, medido en número de caracteres, que debe ocupar el dato en pantalla.

precision: Indica el número de decimales que serán representados. Solo aplicable a datos de tipo float o double.

SALIDA DE DATOS

indicador_de_formato: es el conjunto de caracteres que determina el formato de salida.

INDICADORES DE FORMATO			
Indicador	Significado	Indicador	Significado
-	Alineación a la izquierda	+	Mostrar signo + en números positivos
(Los números negativos se muestran entre paréntesis	0	Rellenar con ceros
,	Muestra el separador decimal		

carácter_de_conversión: Carácter que indica cómo tiene que ser formateado el dato. Los más utilizados se muestran en la tabla.

CARACTERES DE CONVERSIÓN			
Carácter	Tipo	Carácter	Tipo
d	Número entero en base decimal	X, x	Número entero en base hexadecimal
f	Número real con punto fijo	s	String
E, e	Número real notación científica	S	String en mayúsculas
g	Número real. Se representará con notación científica si el número es muy grande o muy pequeño	C, c	Carácter Unicode. C: en mayúsculas