

3.F. Entrada y salida de datos por consola.

Marcar como hecha

1. Programación de la consola: entrada y salida de la información.

1.2. Entrada por teclado. Clase System.

A continuación vamos a ver un ejemplo de cómo utilizar la clase `System` para la entrada de datos por teclado en Java.



Como ya hemos visto en unidades anteriores, para compilar y ejecutar el ejemplo puedes utilizar las órdenes `javac` y `java`, o bien crear un nuevo proyecto en `Eclipse` o `Netbeans` y `copiar el código` que se proporciona en el archivo anterior.

```

1 import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

/*
 * Ejemplo de entrada por teclado con la clase System
 */

1 /**
 *
 * @author FMA
 */
public class entradateclado {
1 | public static void main(String[] args) {
    try
    {
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(isr);

        System.out.print("Introduce el texto: ");
        String cad = br.readLine();

        //salida por pantalla del texto introducido
        System.out.println(cad);

        System.out.print("Introduce un numero: ");
        int num = Integer.parseInt(br.readLine());

        // salida por pantalla del numero introducido
        System.out.println(num);

    } catch (Exception e) {
        // System.out.println("Error al leer datos");
        e.printStackTrace();
    }
}
}

```

El mismo código copiable:

```
import java.io.BufferedReader;
```

```
import java.io.InputStreamReader;
```

```
/*
```

```
 * Ejemplo de entrada por teclado con la clase System
```

```
 */
```

```
/**
```

```
 *
```

```
 * @author FMA
```

```
 */
```

```

public class EntradaTecladoSystem {

    public static void main(String[] args) {

        try

        {

            InputStreamReader isr = new InputStreamReader(System.in);

            BufferedReader br = new BufferedReader(isr);


            System.out.print("Introduce el texto: ");

            String cad = br.readLine();


            //salida por pantalla del texto introducido

            System.out.println(cad);


            System.out.print("Introduce un numero: ");

            int num = Integer.parseInt(br.readLine());


            // salida por pantalla del numero introducido

            System.out.println(num);


        } catch (Exception e) {

            // System.out.println("Error al leer datos");

            e.printStackTrace();

        }

    }

}

```

Observa que hemos metido el código **entre excepciones try-catch**. Cuando en nuestro programa falla algo, por ejemplo la conversión de un **String** a **int**, Java nos avisa lanzando excepciones. Si "capturamos" esa excepción en nuestro programa, podemos avisar al usuario de qué ha pasado. Esto es conveniente porque si no tratamos la **System** excepción seguramente el programa se pare y no siga ejecutándose. El control de excepciones lo veremos en unidades posteriores, ahora sólo nos basta saber que en las llaves del **try** colocamos el código que puede fallar y en las llaves del **catch** el tratamiento de la excepción.

◀ 3.E. Librerías y paquetes.

Ir a...

