

PROGRAMACIÓN

ANEXO I: Scanner

ANEXO I: Scanner	1
1. ¿Qué es la clase Scanner?	1
2. ¿Cómo usarla?	1
3. Control de errores.	3
4. Un poco más	4

1. ¿Qué es la clase Scanner?

La clase Scanner es una clase ya implementada dentro del paquete de Java "útil" que nos permite *leer* o capturar datos por la consola de comandos (cmd).

Es decir, en mitad de nuestro programa, podemos pedir al usuario que introduzca un dato, como por ejemplo "Introduzca el número de días que tiene este mes", y dentro del programa, con ese número, que será diferente en cada ejecución, realizar las acciones que consideremos.

2. ¿Cómo usarla?

Lo primero que debemos hacer es decirle al programa que vamos a usar el paquete que contiene a esta clase (y sus métodos) ya definidos; por lo que en la primera línea de código (antes de declarar la propia clase) haremos una importación del paquete.

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

Una vez la hayamos importado ya podremos usarla en nuestro código, ¿cómo la declaramos?

Lo primero que tenemos que hacer es declarar una variable de tipo Scanner:

```
Scanner sc = new Scanner(System.in);
```

Con esta sentencia estaremos recogiendo en “sc” todo lo que introduzcamos por teclado.

Ahora, cuando ejecutemos el programa, este se detendrá en esta línea a la espera de que por teclado introduzcamos una cadena de caracteres.

¿Cómo pasamos esa cadena de datos introducimos a variables de nuestro programa?

Depende del tipo de dato que hayamos introducido; es decir:

Si hemos introducido un dato booleano, usaremos el siguiente método:

nextBoolean()

```
boolean bool = sc.nextBoolean();
```

Si hemos introducido un dato entero, usaremos el siguiente método:

nextInt()

```
int num = sc.nextInt();
```

Si hemos introducido un dato decimal, usaremos el siguiente método:

nextDouble()

```
double num = sc.nextDouble();
```

Si hemos introducido una cadena de texto (String), usaremos el siguiente método:

nextLine()

```
String linea = sc.nextLine();
```

3. Control de errores.

Para que no tengamos una excepción en nuestro programa, debemos asegurarnos de que el valor que hemos introducido es del tipo de datos que le decimos que es, ya que, si vamos a recoger un número, pero por teclado escribimos una palabra vamos a tener una excepción.

Para solucionar esto, antes de recoger un valor, le vamos a preguntar que efectivamente es del tipo que queremos que sea.

(Nota: Por defecto todo va a ser un String, por lo que este tipo de dato no nos va a causar error)

Si queremos leer un número, preguntaremos si es un número con el método:

sc.hasNextInt()

```
if ( sc.hasNextInt() ) { ...
```

Este método nos permitirá continuar (devolverá **true**) si hemos introducido un número por teclado.

Por tanto, todo el código será:

```
if ( sc.hasNextInt() ) {  
    int num = sc.nextInt();  
    ...  
    // Resto del código  
}
```

Lo mismo ocurrirá para el resto de datos, (double, boolean, char, ...)

4. Un poco más

Para ser más finos en nuestro código, hay dos detalles importantes sobre la clase Scanner que podemos tener en cuenta.

El primero es que una vez terminemos de leer datos, la mejor práctica es “cerrar el Scanner”; es decir, decirle al programa que ya no vamos a usarla más.

Esto se realiza introduciendo la siguiente línea

sc.close();

(ojo, no podemos leer si no declaramos otro Scanner nuevo, ya que el anterior lo hemos destruido)

Y el último, si queremos mantener el Scanner abierto para poder leer infinitos datos por teclado (o hasta que introduzcamos por ejemplo un “exit”) podemos usar el siguiente método:

sc.hasNext();

Este método se introducirá dentro de un bucle *while*, como ya veremos más adelante en la siguiente práctica :)