Anexo I.- Entrada de datos por teclado con *Scanner*

1.- Como **primera sentencia del fichero .java** que contiene el código se tiene que poner la sentencia

import java.util.Scanner;

2.- Crear un objeto de la clase *Scanner*. El parámetro System.in representa el dispositivo estándar de entrada de datos, que es el teclado.

Scanner lector=new Scanner(System.in);

3.- Acceder a los métodos de la clase *Scanner* que necesitamos. La clase *Scanner* dispone de un métodos para cada tipo de datos primitivo (de momento hemos visto int y double) que son nextInt() y nextDouble(). Para leer cadenas de texto (tipo de datos *String)* disponemos de dos métodos: next() y nextLine(). La diferencia entre estos dos últimos métodos es que next() lee *tokens* (hasta que encuentra un espacio) y nextLine() incluye espacios y lee hasta que encuentra un salto de línea.

Estos métodos se aplicarán sobre el objeto creado en el apartado 2. Ejemplo de un fragmento de código con utilización de los métodos:

```
Scanner lector = new Scanner(System.in);
String nombre;
int edad;

System.out.print("Inserta el nombre: ");

nombre=lector.nextLine();

System.out.print("Inserta la edad: ");

edad=lector.nextInt();

System.out.println(nombre);
System.out.println(edad);
```

4.- Cuando estamos leyendo un conjunto de datos, es posible que en algún momento determinado los métodos anteriores den lugar a situaciones inesperadas. Por ejemplo, que en un punto del programa queremos leer una línea, pero el sistema no nos deja. Esto suele ocurrir porque previamente tenemos alguna instrucción de leer un dato de tipo numérico. El comportamiento habitual del usuario es que después de introducir el número, se dé a la tecla Enter (salto de línea, \n en Java). Si a continuación, tenemos una llamada al método nextLine() no nos dejará introducir la cadena por teclado, puesto que guardará el \n que sigue al número como String.

```
Scanner lector = new Scanner(System.in);
String nombre;
int edad;

System.out.print("Inserta la edad: ");

edad=lector.nextInt();
```

```
        System.out.print("Inserta el nombre: ");

        nombre=lector.nextLine();

        System.out.println(nombre);
        System.out.println(edad);
```

En el programa anterior, solo se visualizaría el valor de la edad, pero no dejaría opción para introducir el nombre y, por supuesto, tampoco lo mostraría.

Para resolver estos problemas tenemos dos soluciones:

- **Opción A:después de los métodos que leen valores numéricos, hacer un nextLine()**

```
        Scanner lector=new Scanner(System.in);
        String nombre;
        int edad;

        System.out.print("Inserta la edad: ");
        edad=lector.nextInt();
        lector.nextLine();
        System.out.print("Inserta el nombre: ");
        nombre=lector.nextLine();

        System.out.println(nombre);
        System.out.println(edad);
}
```

- **Opción B: utilizar siempre el método nextLine(), independientemente del tipo de datos, y hacer la conversión al tipo deseado.**

```
            Scanner lector=new Scanner(System.in);
            String nombre;
            int edad;

            System.out.print("Inserta la edad: ");
            edad=Integer.parseInt(lector.nextLine());

            System.out.print("Inserta el nombre: ");
            nombre=lector.nextLine();


            System.out.println(nombre);
            System.out.println(edad);
```

El método *parseInt* de la clase *Integer* convertir un *String* (se le tiene que pasar como parámetro un *String*) a un número entero.

Del mismo modo, si queremos convertir un *String* a un *double* utilizaremos el método *parseDouble* de la clase *Double* y le pasaremos el parámetro de tipo *String.*