**Uso de Scanner para permitir entrada de datos en Java.**

Información brindada por: http://puntocomnoesunlenguaje.blogspot.com/

La clase Scanner está disponible a partir de Java 5 y facilita la lectura de datos en los programas Java.

Para utilizar Scanner en el programa tendremos que hacer lo siguiente:

**1.** **Escribir el import**

Se debe incluir al inicio del programa la instrucción:

import java.util.Scanner;

**2.** **Crear un objeto Scanner**

Tenemos que crear un objeto de la clase Scanner asociado al dispositivo de entrada.

Si el dispositivo de entrada es el teclado escribiremos:

Scanner sc = new Scanner(System.in);

Se ha creado el objeto *sc* asociado al teclado representado por *System.in*

Una vez hecho esto podemos leer datos por teclado.

**Ejemplos de lectura:**

Para leer podemos usar el método nextXxx() donde Xxx indica en tipo, por ejemplo nextInt() para leer un entero, nextDouble() para leer un double, etc.

**Ejemplo** de lectura por teclado de un número entero:

int n;

System.out.print("Introduzca un número entero: ");

n = sc.nextInt();

**Ejemplo** de lectura de un número de tipo double:

double x;

System.out.print("Introduzca número de tipo double: ");

x = sc.nextDouble();

**Ejemplo** de lectura de una cadena de caracteres:

String s;

System.out.print("Introduzca texto: ");

s = sc.nextLine();

**Ejemplo de programa Java con lectura de datos con Scanner**:

El programa pide que se introduzca el nombre de la persona y lo muestra por pantalla. A continuación lee por teclado el radio de una circunferencia de tipo double y muestra su longitud. Además lee un entero y muestra su cuadrado.

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in); //crear un objeto Scanner

String nombre;

double radio;

int n;

System.out.print("Introduzca su nombre: ");

nombre = sc.nextLine(); //leer un String

System.out.println("Hola " + nombre + "!!!");

System.out.print("Introduzca el radio de la circunferencia: ");

radio = sc.nextDouble(); //leer un double

System.out.println("Longitud de la circunferencia: " + 2\*Math.PI\*radio);

System.out.print("Introduzca un número entero: ");

n = sc.nextInt(); //leer un entero

System.out.println("El cuadrado es: " + Math.pow(n,2));

}

}

**Funcionamiento la clase Java Scanner.**

De forma resumida podemos decir que cuando se introducen caracteres por teclado, el objeto Scanner toma toda la cadena introducida y la divide en elementos llamados **tokens**.

El carácter predeterminado que sirve de separador de tokens es el espacio en blanco.

Por ejemplo, si introducimos:

Esto es un ejemplo, lectura de datos.

Scanner divide la cadena en los siguientes tokens:

Esto

es

un

ejemplo,

lectura

de

datos.

Si introducimos la cadena:

12 20.001 Lucas w

Los tokens que se crean son:

12

20.001

Lucas

w

A continuación, utilizando los métodos que proporciona la clase Scanner se puede acceder a esos tokens y trabajar con ellos en el programa.

Ya hemos visto el método nextXxx(). Además la clase Scanner proporciona otros métodos, algunos de los métodos más usados son:

|  |  |
| --- | --- |
| **METODO** | **DESCRIPCIÓN** |
| **nextXxx()** | Devuelve el siguiente token como un tipo básico. Xxx es el tipo. Por ejemplo, nextInt() para leer un entero, nextDouble para leer un double, etc. |
| **next()** | Devuelve el siguiente token como un String. |
| **nextLine()** | Devuelve la línea entera como un String. |
| **hasNext()** | Devuelve un boolean. Indica si existe o no un siguiente token para leer. |
| **hasNextXxx()** | Devuelve un boolean. Indica si existe o no un siguiente token del tipo especificado en Xxx, por ejemplo hasNextDouble() |
| **useDelimiter(String)** | Establece un nuevo delimitador de token. |

**Cómo limpiar el buffer de entrada en Java**

Cuado en un programa se leen por teclado datos numéricos y datos de tipo carácter o String debemos tener en cuenta que al introducir los datos y pulsar intro estamos también introduciendo en el buffer de entrada el intro.

Es decir, cuando en un programa introducimos un datos y pulsamos el intro como final de entrada, el carácter intro también pasa al buffer de entrada.

Buffer de entrada si se introduce un 5: 5\n

En esta situación, la instrucción:

n = sc.nextInt();

Asigna a n el valor 5 pero el intro permanece en el buffer

Buffer de entrada después de leer el entero: \n

Si ahora se pide que se introduzca por teclado una cadena de caracteres:

System.out.print("Introduzca su nombre: ");

nombre = sc.nextLine(); //leer un String

El método nextLine() extrae del buffer de entrada todos los caracteres hasta llegar a un salto de línea y elimina el salto de línea del buffer.

En este caso asigna una cadena vacía a la variable nombre y limpia el intro. Esto provoca que el programa no funcione correctamente, ya que no se detiene para que se introduzca el nombre.

Solución:

Se debe **limpiar el buffer** de entrada si se van a leer datos de tipo carácter a continuación de la lectura de datos numéricos.

La forma más sencilla de limpiar el buffer de entrada en Java es ejecutar la instrucción:

sc.nextLine();

Lo podemos comprobar si cambiamos el orden de lectura del ejemplo y leemos el nombre al final:

import java.util.Scanner;

public class JavaApplication335 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String nombre;

double radio;

int n;

System.out.print("Introduzca el radio de la circunferencia: ");

radio = sc.nextDouble();

System.out.println("Longitud de la circunferencia: " + 2\*Math.PI\*radio);

System.out.print("Introduzca un número entero: ");

n = sc.nextInt();

System.out.println("El cuadrado es: " + Math.pow(n,2));

System.out.print("Introduzca su nombre: ");

nombre = sc.nextLine(); //leemos el String después del double

System.out.println("Hola " + nombre + "!!!");

}

}

Si lo ejecutamos obtendremos:

Introduzca el radio de la circunferencia: 34

Longitud de la circunferencia: 213.62830044410595

Introduzca un número entero: 3

El cuadrado es: 9.0

Introduzca su nombre: Hola !!!

Comprobamos que no se detiene para pedir el nombre.

La solución es escribir la instrucción

sc.nextLine();

después de la lectura del int y antes de leer el String:

import java.util.Scanner;

public class JavaApplication335 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String nombre;

double radio;

int n;

System.out.print("Introduzca el radio de la circunferencia: ");

radio = sc.nextDouble();

System.out.println("Longitud de la circunferencia: " + 2\*Math.PI\*radio);

System.out.print("Introduzca un número entero: ");

n = sc.nextInt();

System.out.println("El cuadrado es: " + Math.pow(n,2));

sc.nextLine();

System.out.print("Introduzca su nombre: ");

nombre = sc.nextLine();

System.out.println("Hola " + nombre + "!!!");

}

}

Ahora la ejecución es correcta:

Introduzca el radio de la circunferencia: 23

Longitud de la circunferencia: 144.51326206513048

Introduzca un número entero: 5

El cuadrado es: 25.0

Introduzca su nombre: Lucas

Hola Lucas!!!