

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

Marcar como hecha

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

1.1. Conceptos sobre la clase System.

La lectura por teclado es muy importante cuando empezamos a hacer nuestros primeros programas. Para entender mejor en qué consiste la clase `System`, y en particular el objeto `System.in` vamos a describirlo más detenidamente.

Field Summary	
Fields	
Modifier and Type	Field and Description
static <code>PrintStream</code>	<code>err</code> The "standard" error output stream.
static <code>InputStream</code>	<code>in</code> The "standard" input stream.
static <code>PrintStream</code>	<code>out</code> The "standard" output stream.

Imagen extraída de curso Programación del MECD.

En el apartado anterior hemos dicho que `System.in` es un atributo de la clase `System`, que está dentro del paquete `java.lang`. Pero además, si consultamos la Biblioteca de Clases de Java, nos damos cuenta que es un objeto, y como todos los objetos debe ser instanciado. En efecto, volviendo a consultar la biblioteca de clases nos damos cuenta que `System.in` es una instancia de una clase de java que se llama `InputStream`.

En Java, `InputStream` nos permite leer en bytes, desde teclado, un archivo o cualquier otro dispositivo de entrada. Con esta clase podemos utilizar por ejemplo el método `read()` que permite leer un byte de la entrada o `skip(long n)`, que salta n bytes de la entrada. Pero lo que realmente nos interesa es poder leer texto o números, no bytes, para hacernos más cómoda la entrada de datos. Para ello se utilizan las clases:

- `InputStreamReader`. Convierte los bytes leídos en caracteres. Particularmente, nos va a servir para convertir el objeto `System.in` en otro tipo de objeto que nos permita leer caracteres.
- `BufferedReader`. Lee hasta un fin de línea. Esta es la clase que nos interesa utilizar, pues tiene un método `readLine()` que nos va a permitir leer caracteres hasta el final de línea.

ENCAPSULADO DE OBJETOS

Encapsulado de objetos para entrada por teclado

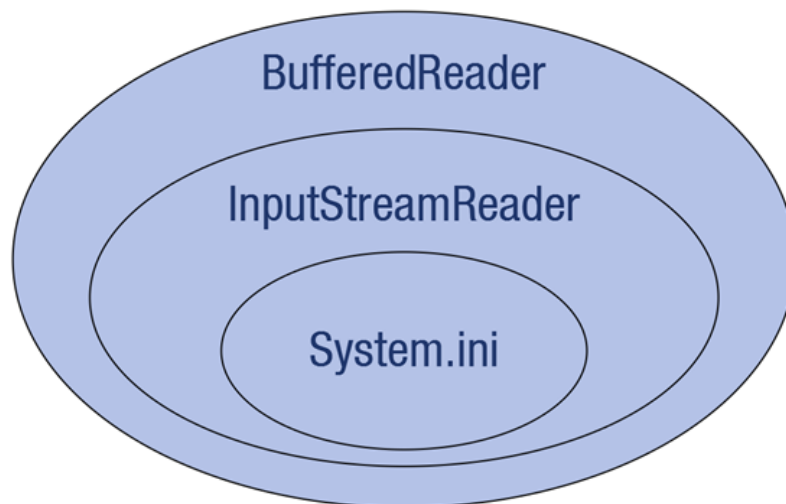


Imagen extraída de curso Programación del MECD.

La forma de instanciar estas clases para usarlas con System.in es la siguiente:

```
InputStreamReader isr = new InputStreamReader(System.in);
```

```
BufferedReader br = new BufferedReader (isr);
```

En el código anterior hemos creado un `InputStreamReader` a partir de `System.in` y pasamos dicho `InputStreamReader` al constructor de `BufferedReader`. El resultado es que las lecturas que hagamos con el objeto `br` son en realidad realizadas sobre `System.in`, pero con la ventaja de que podemos leer una línea completa. Así, por ejemplo, si escribimos una `A`, con:

```
String cadena = br.readLine();
```

Obtendremos en `cadena` una `"A"`.

Sin embargo, seguimos necesitando hacer la conversión si queremos leer números. Por ejemplo, si escribimos un entero `32`, en `cadena` obtendremos `"32"`. Si recordamos, para convertir cadenas de texto a enteros se utiliza el método estático `parseInt()` de la clase `Integer`, con lo cual la lectura la haríamos así:

```
int numero = Integer.parseInt (br.readLine());
```