



OBJETIVOS

- Construir los primeros programas Java simples.
- Aprender cómo se leen y escriben datos en la consola.

INTRODUCCIÓN

El objetivo de este primer boletín de ejercicios es familiarizarse con el entorno de trabajo y crear nuestros primeros programas Java. Tenga en cuenta que no es recomendable implementar los programas tal y como lo vamos a realizar en este boletín. Los ejercicios de este boletín deben entenderse como un mero aprendizaje de algunos de los conceptos básicos que se usan en cualquier lenguaje de programación, como son: variables, constantes, literales, expresiones o métodos. El esquema de diseño de tipos, que usaremos para realizar correctamente nuestros programas, se introducirá en boletines posteriores.

ESCRITURA DE DATOS EN LA CONSOLA

Para escribir los datos por consola usaremos una llamada al método `println`. Así, por ejemplo, si escribimos la línea de código:

```
System.out.println("El valor de la variable n es " + n + ".");
```

Suponiendo que `n` se ha declarado como variable entera y tiene el valor 5, cuando se ejecuta la línea de código anterior se obtiene por consola el siguiente resultado:

```
El valor de la variable n es 5.
```

Igual que existe el método `println`, hay un método llamado `print`. La diferencia entre ambos es que el primero imprime el carácter fin de línea al final de la cadena que se pasa como argumento. Así, si escribimos las siguientes líneas de código:

```
System.out.print("El valor de la variable n es " + n + ".");  
System.out.println("Fin");
```

La palabra `Fin` se escribirá tras el punto, siendo el resultado

```
El valor de la variable n es 5.Fin
```

Sin embargo, si la llamada a `print` fuera sustituida por una llamada a `println`, el resultado sería

```
El valor de la variable n es 5.  
Fin
```

LECTURA DE DATOS DE LA CONSOLA

Para leer datos de la consola nos apoyaremos en la clase `Scanner`¹ de Java. En este punto, no entraremos en muchos detalles de su funcionamiento, sino que la usaremos de forma intuitiva. Lo primero que tenemos que hacer para leer datos de la consola es crear un objeto de tipo `Scanner` a partir de la entrada estándar, de la siguiente forma:

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

La clase `Scanner` proporciona varios métodos que podemos usar para la lectura de datos. En todos ellos, la entrada de datos terminará cuando se introduzca un retorno de carro. Los métodos más interesantes son los siguientes:

¹ <http://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html>



boolean nextBoolean()	Escanea la entrada y devuelve un valor de tipo boolean .
double nextDouble()	Escanea la entrada y devuelve un valor de tipo double .
float nextFloat()	Escanea la entrada y devuelve un valor de tipo float .
int nextInt()	Escanea la entrada y devuelve un valor de tipo int .
String nextLine()	Escanea la entrada y devuelve una cadena de caracteres.
long nextLong()	Escanea la entrada y devuelve leer un valor de tipo long .

Así, si queremos leer una variable de tipo entero escribiremos las siguientes líneas de código:

```
Scanner sc = new Scanner (System.in);
Integer n = sc.nextInt();
```

Cuando se usa uno de los métodos de `Scanner` mencionado anteriormente, el programa se detiene y espera a que el usuario introduzca información por la consola. Cuando el usuario pulsa la tecla retorno de carro, el método recoge la información y la devuelve convenientemente formateada al tipo devuelto. Así en el ejemplo anterior, el método `nextInt()`, devolvería la información como un entero, que se almacenaría en la variable `n`. Luego, el programa continúa su curso.

Finalmente, merece la pena mencionar que una vez que se termine de trabajar con el objeto `Scanner` hay que cerrarlo mediante el método `close`, de la siguiente forma:

```
sc.close();
```

EJERCICIOS PARA HACER EN EL AULA

Cree un proyecto Java llamado B01.1-aula, añada un paquete llamado `main`, y agréguele las clases que se piden en los siguientes ejercicios:

1. Escriba en una clase Java llamada `HolaMundo` un método `main` que permita mostrar el mensaje “Hola, mundo!” por la consola. Un ejemplo de ejecución del programa es el siguiente:

```
¡Hola, Mundo!
```

2. Escriba en una clase Java llamada `Saludo` un método `main` que solicite el nombre al usuario, y escriba un saludo con su nombre. Un ejemplo de ejecución del programa es el siguiente:

```
Introduzca su nombre:
Jane
¡Hola, Jane!
```

3. Escriba en una clase Java llamada `Nacimiento` un método `main` que solicite la edad de una persona y escriba su año de nacimiento. Un ejemplo de ejecución del programa es el siguiente:

```
Introduzca su edad:
30
Tu año de nacimiento es 1983 si ya ha pasado tu cumpleaños; si no, naciste en 1982.
```

4. Escriba en una clase Java llamada `CuentaCaracteres` un método `main` que solicite una frase y escriba el número de caracteres de la frase. Un ejemplo de ejecución del programa es el siguiente:

```
Introduzca una frase:
En un lugar de la Mancha...
La frase tiene 27 caracteres
```

5. Escriba en una clase Java llamada `IMC` un método `main` que solicite el peso (en kilos) y la estatura de una persona (en metros) y escriba el índice de masa corporal (IMC) de dicha persona. El IMC se calcula como $\frac{\text{peso}}{\text{estatura}^2}$. Un ejemplo de ejecución del programa es el siguiente:



```
Introduzca su peso (en kilos):  
80,0  
Introduzca su estatura (en metros):  
1,80  
Su IMC es: 24.69136
```

6. Escriba en una clase Java llamada `Minuscula` un método `main` que solicite un carácter y escriba un mensaje que indique si el carácter es una minúscula o no. Un par de ejemplos de ejecución del programa son los siguientes:

```
Ejecución 1:  
Introduzca un carácter:  
a  
El carácter a es minúscula
```

```
Ejecución 2:  
Introduzca un carácter:  
B  
El carácter B no es minúscula
```

7. Escriba en una clase Java llamada `Distancia` un método `main` que solicite las coordenadas `x` e `y` de un punto y escriba la distancia de ese punto al origen de coordenadas. Utilice la implementación del tipo `Punto` que se le proporciona en el archivo `FP-B01.1.jar`. Un ejemplo de ejecución del programa es el siguiente:

```
Introduzca la coordenada X:  
3,5  
Introduzca la coordenada Y:  
-1,5  
La distancia de (3.5,-1.5) al origen es 3.8078865529319543
```

8. Exporte el proyecto en un archivo `.zip`.

EJERCICIOS PARA HACER EN CASA

Cree un proyecto Java llamado `B01.1-casa`, añada un paquete llamado `main`, y agréguele las clases que se piden en los siguientes ejercicios:

1. Escriba en una clase Java llamada `Eco` un método `main` que solicite una frase, y escriba esa frase por la consola, como si fuese un eco. Un ejemplo de ejecución del programa es el siguiente:

```
Introduzca una frase:  
En un lugar de La Mancha...  
ECO: En un lugar de La Mancha...
```

2. Escriba en una clase Java llamada `Cuadrado` un método `main` que solicite un número entero, y escriba el cuadrado de ese número por la consola. Un ejemplo de ejecución del programa es el siguiente:

```
Introduzca un número:  
34  
El cuadrado de 34 es 1156
```

3. Escriba en una clase Java llamada `PositivoPar` un método `main` que solicite un número entero, y escriba si el número es positivo, negativo o cero, y también si el número es par o impar. Tres ejemplos de ejecución del programa son los siguientes:

```
Ejecución 1:  
  
Introduzca un número:  
16
```



El número 16 es positivo y par

Ejecución 2:

Introduzca un número:

-3

El número -3 es negativo e impar

Ejecución 3:

Introduzca un número:

0

El número 0 es cero y par

4. Escriba en una clase Java llamada `Perimetro` un método `main` que dados tres puntos $A(-2.0, 2.0)$, $B(1.0, 6.0)$ y $C(6.0, -6.0)$, que forman un triángulo, escriba cuál es el perímetro del triángulo. Utilice la implementación del tipo `Punto` que se le proporciona en el archivo `FP-B01.1.jar`. Un ejemplo de ejecución del programa es el siguiente:

El perímetro del triángulo formado por $A(-2.0, 2.0)$, $B(1.0, 6.0)$ y $C(6.0, -6.0)$ es 29.31370849898476