

Tema 1: TIPOS DE DATOS, CONSTANTES Y VARIABLES

Cualquier duda sobre el contenido de este tema se puede enviar al foro **TEORIA1**.

1.1.- ¿Qué es un Programa?

Un PROGRAMA está formado por una serie de INSTRUCCIONES y de estructuras de DATOS, que al ejecutarse en un ordenador, en general, acepta una serie de datos de ENTRADA y produce unos resultados de SALIDA, ejecutando para ello las instrucciones y manejando las estructuras de datos que componen el programa.



PROGRAMA = DATOS + INSTRUCCIONES

1.2.- Tipo de datos

Se caracteriza por:

- ❖ dominio de posibles valores: qué valores puede tomar
- ❖ cómo se representan: cuál es la representación interna y cómo se representan en el lenguaje de programación Java.
- ❖ operadores asociados: qué operaciones o cálculos se pueden realizar con esos datos.

Algunos tipos de datos en Java

simples escalares	{	entero	(byte, short, int, long)
		booleano	(boolean)
		carácter	(char)
		real	(float, double)

Y otros estructurados¹, como por ejemplo las cadenas de caracteres (String)

Más información en: <http://java.sun.com/docs/books/tutorial/java/nutsandbolts/datatypes.html>

¹ Para ser más precisos, `String` es en realidad una clase en Java y no un tipo de datos estructurado.

1.2.- Identificador

Un identificador es cualquier nombre formado por letras, cifras, el carácter `_` o el carácter `$`, que no comience por una cifra (aunque habitualmente suelen comenzarse siempre por una letra y no se suele utilizar el carácter `$`). Las letras en mayúscula o minúscula son diferentes. No se pueden utilizar como identificadores las palabras reservadas en Java (aquellas que tienen significado propio en Java).

Ejemplo de identificadores válidos:

```
edad
dia3
num_mes
```

Ejemplo de identificadores no válidos:

```
5dia (Empieza por una cifra)
int (Es una palabra reservada en Java: corresponde a un tipo de datos entero)
if (Es una palabra reservada en Java: se utiliza en instrucciones condicionales)
```

Más información en: <http://java.sun.com/docs/books/tutorial/java/nutsandbolts/variables.html> (Apartado Naming)

1.3.- Datos constantes

Son valores inalterables durante la ejecución del programa

Ejemplos:	1000	(constante de tipo entero)
	89.3545	(constante de tipo real)
	3.141592	(constante de tipo real)
	-3.15e-5	(constante de tipo real, en notación científica)
	'A'	(constante de tipo carácter)
	"Pedro"	(constante de tipo cadena de caracteres o string)

En un programa se pueden definir constantes a las que se les puede dar un nombre (un identificador). Habitualmente en Java los identificadores que se utilizan para dar nombres a constantes suelen estar formados únicamente por letras mayúsculas.

Ejemplo: definición en Java de una constante de tipo real (`float`) cuyo valor es `3.14` y a la que se le da el nombre `PI`.

```
final float PI = 3.14; // Definición de la constante PI
```

Ventajas de dar nombres a las constantes:

- ❖ mayor legibilidad. Dentro del programa se usará `PI` en vez de `3.14`, lo cual puede dar una idea más clara de qué valor es ese.
- ❖ facilita las modificaciones. Si en un momento determinado se prefiriera usar un valor más preciso para `PI` (`3.14159` en vez de `3.14`) bastaría con cambiarlo una sola vez en todo el programa (donde se defina).

1.4.- Datos variables (o simplemente "variables")

- ❖ Son valores que pueden cambiar durante la ejecución de un programa.
- ❖ Se les asocia un nombre (un identificador) y un tipo que indica qué tipo de valores pueden almacenarse en dicha variable.

Por ejemplo: la variable `edad` que en un momento determinado tiene el valor 17, que puede tomar como posible valor cualquier número entero (`int`)

edad



Debe indicarse el tipo de datos (int en este caso), que define el dominio de posibles valores que puede tomar la variable edad

En Java la variable `edad` de tipo `int` (sin darle como valor 17) se define así:

```
int edad;
/* Definición de una variable llamada edad,
que puede tomar como valor un número entero (int) */
```

Una variable es una abstracción del concepto de celda de memoria (es una "caja" donde se guarda un valor de un tipo determinado)

Siempre que se quiera utilizar más adelante un valor habrá que guardarlo en una variable

En general, toda INSTRUCCIÓN cambia y/o utiliza los DATOS contenidos en variables

1.5.- Comentarios en Java

En todo programa suele ser conveniente añadir comentarios que sirvan para aclarar lo que se desee a quien lea después el programa (el mismo programador al cabo de un cierto tiempo u otro programador distinto que deba modificarlo). Han aparecido en las dos definiciones Java anteriores:

```
final float PI = 3.14; // Definición de la constante PI
```

```
int edad;
/* Definición de una variable llamada edad,
que puede tomar como valor un número entero (int) */
```

Los comentarios son ignorados por el compilador así que no influyen en la eficiencia del programa, esto es, la rapidez con la que se ejecuta el programa.

En Java los comentarios pueden ser de distintos tipos:

```
// Esto es un comentario hasta el final de la línea
/* Esto es un comentario de
tantas líneas como se quiera hasta que se termine así: */
```