

Introducción a Java

Módulo 1

Sintaxis y semántica de Java

Variables

¿Qué es una variable?

Una **variable** es un nombre que se asocia con una porción de la memoria del ordenador en la que se guarda **el valor asignado a dicha variable**. Consiste en un elemento al cual **le damos un nombre y le atribuimos determinado tipo de información**.

Las variables pueden ser consideradas como la base de la programación. Los datos que se manejan en nuestro programa **se almacenan en variables**.

El concepto de variable debe verse como un **contenedor de información**, de este modo podríamos escribir en un lenguaje ficticio:

a="perro"

b="ladra"

La variable que nosotros llamamos "a" posee un elemento de información de tipo texto que es "perro". Asimismo, la variable "b" contiene el valor "ladra".

Podríamos definir una tercera variable que fuese la suma de las dos variables anteriores:

c=a+b

Si introdujeramos una petición de impresión de esta variable en nuestro lenguaje ficticio:

Imprimir (c)

El resultado podría ser:

Perro ladra

Podríamos de la misma forma trabajar con variables que contuviesen números y construir nuestro programa:

a=3

b=4

c=a+b

Imprimir (c)

El resultado sería: **7**

Hay varios tipos de variables que requieren distintas cantidades de memoria para guardar datos.

Todas las variables deben declararse antes de usarse. La **declaración** consiste en **una sentencia en la que figura el tipo de dato y el nombre que asignamos a la variable**. Una vez declarada se le podrán asignar valores.

Identificador

Un **identificador** es un nombre que **identifica a una variable, a un método o función miembro, a una clase**. Cada lenguaje tiene reglas para componer los identificadores, que veremos a continuación.

Definición de variables en Java

Identificador (nombre)

- Puede comenzar con "_" o "\$".
- No puede utilizar caracteres "%" o "*" o "@" pues son reservados para otras operaciones.
- Puede incluir, pero no comenzar por un número.
- No puede incluir el carácter espacio en blanco.
- Distingue entre mayúsculas y minúsculas.
- No se pueden utilizar las palabras reservadas como identificadores.

Tipos de variables

Una **variable** es *algo que cambia o varía*.

En Java, una variable almacena datos. Los tipos de variables definen **el tipo de dato que puede ser almacenado en una variable y los límites de los datos**.

A toda variable que se use en un programa se le debe asociar (generalmente, al principio del programa) **un tipo de dato específico**.

Un tipo de dato define todo el posible rango de valores que una variable puede tomar al momento de ejecución del programa y a lo largo de toda su vida útil.

Tipos de dato:

- **Primitivos**
Int, long, float, double, etc.
- **Objetos ***

(*) Se profundizará en Java Standard e Introducción al Paradigma con orientación a Objetos.

Ejemplos de declaración y definición:

- de una **variable simple**:

```
int var = 5;
```

- de un **array**:

```
int[] vec = new int[10]; vec[0] = 150; vec[1] = 500;
```

```
int vec[] = new int[10]; vec[0] = 150; vec[1] = 500;
```

```
int[] vec = {150,500,3,4,5,6};
```

Un *array* es un objeto que tiene una *variable miembro* llamada *length*.

Tipos de datos primitivos

Boolean 1 byte. Valores true y false

Char 2 bytes. Unicode. Comprende el código ASCII

Byte 1 byte. Valor entero entre -128 y 127

Short 2 bytes. Valor entero entre -32768 y 32767

Int 4 bytes. Valor entero entre -2.147.483.648 y 2.147.483.647

Long 8 bytes. Valor entre -9.223.372.036.854.775.808 y 9.223.372.036.854.775.807

Float 4 bytes (entre 6 y 7 cifras decimales equivalentes).
De -3.402823E38 a -1.401298E-45 y de 1.401298E-45 a 3.402823E38

Double 8 bytes (unas 15 cifras decimales equivalentes). De -1.79769313486232E308 a -4.94065645841247E-324 y de 4.94065645841247E-324 a 1.79769313486232E308

Carácter

- En Java los caracteres no están restringidos a los ASCII si no son Unicode.
- Un carácter está siempre rodeado de comillas simples como 'A', '9', 'ñ', etc.
- El tipo de dato *char* sirve para guardar estos caracteres.

Boolean

Una variable booleana solamente puede guardar uno de los dos posibles valores: *true* (verdadero) y *false* (falso).



Enteros

Una variable entera consiste en **cualquier combinación de cifras precedidas por el signo “+” (opcional), para los positivos, o el signo “-”, para los negativos.**

Ejemplo: 10, -12, 432, -128

Como ejemplos de declaración de variable enteras tenemos:

```
Int. Numero = 8451;
```

```
Int. x,y;
```

Int. es la palabra reservada para declarar una variable entera.

En el primer caso, el compilador reserva una porción de 32 bits de memoria en el que guarda el número 8451.

Se accede a dicha porción de memoria mediante el nombre de la variable **“número”**.

Enteros Largos

Son valores enteros que van desde

-9.223.372.036.854.775.808 a

9.223.372.036.854.775.807

```
long x;
```

```
long y=50L;
```

En la primera línea **definimos la variable x como long**.

En la segunda línea asignamos a la variable **y el valor 50, que es un número de tipo int**. Por defecto, le ponemos el sufijo **L, en mayúsculas o minúsculas, para indicar que es de tipo long**.



Flotante

Las variables del tipo float o double (coma flotante) se usan para **guardar números en memoria que tienen parte entera y parte decimal**.

```
double PI=3.14159;
```

```
double g=9.7805, c=2.9979e8;
```

El primero es una aproximación del número real π , el segundo es la aceleración de la gravedad a nivel del mar, el tercero es la velocidad de la luz en m/s, que es la forma de escribir $2.9979 \cdot 10^8$. El carácter punto “.”, separa la parte entera de la parte decimal, en vez del carácter coma “,” que usamos habitualmente en nuestro idioma.

Ejemplo:

```
float x=16.2f;
```

```
float y=9f;
```

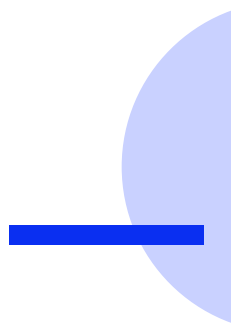
```
double z=21.0;
```

```
double g=7d;
```



Conceptualmente, hay infinitos números de valores entre dos números reales. Ya que los valores de las variables se guardan en un número prefijado de bits, algunos valores no se pueden representar de forma precisa en memoria. Por lo tanto, los valores de las variables en coma flotante en un ordenador **solamente se aproximan a los verdaderos números reales en matemáticas.**

La aproximación es tanto mejor cuanto **mayor sea el tamaño de la memoria que reservamos para guardarlo.** De este hecho, surgen las variables del tipo `float` y `double`.



Palabras reservadas

Las **palabras reservadas** no pueden ser usadas como nombre de variable.

El lenguaje Java tiene la siguiente lista de palabras reservadas:

<code>abstract</code>	<code>double</code>	<code>int</code>	<code>strictfp</code>	<code>boolean</code>	<code>else</code>
<code>interface</code>	<code>super</code>	<code>break</code>	<code>extends</code>	<code>long</code>	<code>switch</code>
<code>byte</code>	<code>final</code>	<code>native</code>	<code>synchronized</code>	<code>case</code>	<code>finally</code>
<code>new</code>	<code>this</code>	<code>catch</code>	<code>float</code>	<code>package</code>	<code>throw</code>
<code>char</code>	<code>for</code>	<code>private</code>	<code>throws</code>	<code>class</code>	<code>goto</code>
<code>protected</code>	<code>transient</code>	<code>const</code>	<code>if</code>	<code>public</code>	<code>try</code>
<code>continue</code>	<code>implements</code>	<code>return</code>	<code>void</code>	<code>default</code>	<code>import</code>
<code>short</code>	<code>volatile</code>	<code>do</code>	<code>instanceof</code>	<code>static</code>	<code>while</code>

**¡Sigamos
trabajando!**

