TEMA 2 - CREACIÓN DE MI PRIMER PROGRAMA

En la definición anterior decimos que un identificador es una secuencia ilimitada de caracteres Unicode. Pero… ¿qué es Unicode? Unicode es un código de caracteres o sistema de codificación, un alfabeto que recoge los caracteres de prácticamente todos los idiomas importantes del mundo. Las líneas de código en los programas se escriben usando ese conjunto de caracteres Unicode.

Esto quiere decir que en Java se pueden utilizar varios alfabetos como el Griego, Árabe o Japonés. De esta forma, los programas están más adaptados a los lenguajes e idiomas locales, por lo que son más significativos y fáciles de entender tanto para los programadores que escriben el código, como para los que posteriormente lo tienen que interpretar, para introducir alguna nueva funcionalidad o modificación en la aplicación.

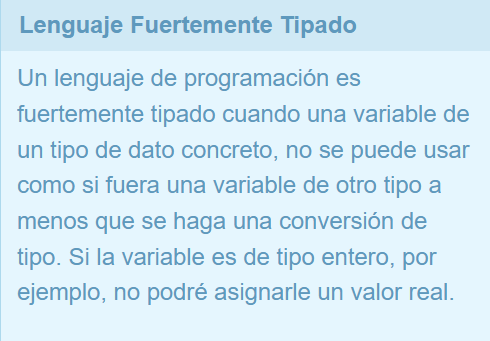
# **Convenios y reglas para nombrar variables.**

* Java distingue las mayúsculas de las minúsculas. Por ejemplo, Alumno y alumno son variables diferentes.
* No se suelen utilizar identificadores que comiencen con «$» o «\_», además el símbolo del dólar, por convenio, no se utiliza nunca.
* No se puede utilizar el valor booleano (true o false) ni el valor nulo (null).
* Los identificadores deben ser lo más descriptivos posibles. Es mejor usar palabras completas en vez de abreviaturas crípticas. Así nuestro código será más fácil de leer y comprender. En muchos casos también hará que nuestro código se autodocumente. Por ejemplo, si tenemos que darle el nombre a una variable que almacena los datos de un cliente sería recomendable que la misma se llamara algo así como FicheroClientes o ManejadorCliente, y no algo poco descriptivo como Cl33.

| **Convenciones sobre identificadores en Java** | | |
| --- | --- | --- |
| **Identificador** | **Convención** | **Ejemplo** |
| Nombre de variable. | Comienza por letra minúscula, y si tienen más de una palabra se colocan juntas y el resto comenzando por mayúsculas. | **numAlumnos, suma** |
| Nombre de constante. | En letras mayúsculas, separando las palabras con el guión bajo, por convenio el guión bajo no se utiliza en ningún otro sitio. | **TAM\_MAX, PI** |
| Nombre de una clase. | Comienza por letra mayúscula. | **String, MiTipo** |
| Nombre de función. | Comienza con letra minúscula. | **modifica\_Valor, obtiene\_Valor** |

**Variables miembro y variables locales, en función del lugar donde aparezcan en el programa. La definición concreta sería:**

* **Variables miembro:**Son las variables que se crean dentro de una [clase](https://ikasaula.educacion.navarra.es/fponline/pluginfile.php/145571/mod_resource/content/3/24_tipos_de_variables.html" \l "t927a49e3-6a6f-6640-b6ea-6456cb692cb4" \o "), fuera de cualquier [método.](https://ikasaula.educacion.navarra.es/fponline/pluginfile.php/145571/mod_resource/content/3/24_tipos_de_variables.html" \l "tc122665f-eb8a-f8f7-63ac-afd3e923842a" \o ") Pueden ser de tipos primitivos o referencias, variables o constantes. En un lenguaje puramente orientado a objetos como es Java, todo se basa en la utilización de [objetos](https://ikasaula.educacion.navarra.es/fponline/pluginfile.php/145571/mod_resource/content/3/24_tipos_de_variables.html" \l "t0045217a-e7fc-7d52-590f-e3adef80336c" \o "), los cuales se crean usando clases. En la siguiente unidad veremos los distintos tipos de variables miembro que se pueden usar.
* **Variables locales:**Son las variables que se crean y usan dentro de un método o, en general, dentro de cualquier bloque de código. La variable deja de existir cuando la ejecución del bloque de código o el método finaliza. Al igual que las variables miembro, las variables locales también pueden ser de tipos primitivos o referencias.



* **Tipos de datos sencillos o primitivos**. Representan valores simples que vienen predefinidos en el lenguaje; contienen valores únicos, como por ejemplo un carácter o un número.
* **Tipos de datos referencia.**Se definen con un nombre o referencia (puntero) que contiene la dirección en memoria de un valor o grupo de valores. Dentro de este tipo tenemos por ejemplo los vectores o arrays, que son una serie de elementos del mismo tipo, o las clases, que son los modelos o plantillas a partir de los cuales se crean los objetos.

**En Java las variables de tipo float** se emplean para representar los números en coma flotante de simple precisión de 32 bits, de los cuales 24 son para la mantisa y 8 para el exponente.

**Por su parte, las variables tipo double** representan los números en coma flotante de doble precisión de 64 bits, de los cuales 53 son para la mantisa y 11 para el exponente.

La mayoría de los programadores en Java emplean el tipo double cuando trabajan con datos de tipo real, es una forma de asegurarse de que los errores cometidos por las sucesivas aproximaciones sean menores. De hecho, Java considera los valores en coma flotante como de tipo double por defecto.

Con el objetivo de conseguir la máxima portabilidad de los programas, Java utiliza el **estándar internacional [IEEE 754](https://ikasaula.educacion.navarra.es/fponline/pluginfile.php/145571/mod_resource/content/3/311_tipos_de_datos_primitivos_ii.html" \l "t967a9fd8-9381-7694-bd20-d980a4c6c1f6" \o ")** para la representación interna de los números en coma flotante, que es una forma de asegurarse de que el resultado de los cálculos sea el mismo para diferentes plataformas.

