2.B. Tipos de datos.

1. Los tipos de datos.

1.1. Tipos de datos primitivos I.

Los tipos primitivos son aquéllos datos sencillos que constituyen los tipos de información más habituales: números, caracteres y valores lógicos o booleanos. Al contrario que en otros lenguajes de programación orientados a objetos, en Java no son objetos.

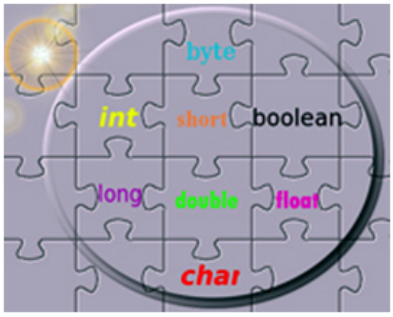

Imagen extraída de curso Programación del MECD.

Una de las mayores ventajas de tratar con tipos primitivos en lugar de con objetos, es que el compilador de Java puede optimizar mejor su uso. Otra importante característica, es que cada uno de los tipos primitivos tiene idéntico tamaño y comportamiento en todas las versiones de Java y para cualquier tipo de ordenador. Esto quiere decir que no debemos preocuparnos de cómo se representarán los datos en distintas plataformas, y asegura la portabilidad de los programas, a diferencia de lo que ocurre con otros lenguajes. Por ejemplo, el tipo `int` siempre se representará con 32 bits, con signo, y en el formato de representación complemento a 2, en cualquier plataforma que soporte Java.

Reflexiona

Java especifica el tamaño y formato de todos los tipos de datos primitivos con independencia de la plataforma o sistema operativo donde se ejecute el programa, de forma que el programador no tiene que preocuparse sobre las dependencias del sistema, y no es necesario escribir versiones distintas del programa para cada plataforma.

Debes conocer

En el siguiente enlace se muestran los tipos de datos primitivos en Java con el rango de valores que pueden tomar, el tamaño que ocupan en memoria y sus valores por defecto.

Tipos de Datos Primitivos en Java

Hay una peculiaridad en los tipos de datos primitivos, y es que el tipo de dato `char` es considerado por el compilador como un tipo numérico, ya que los valores que guarda son el código Unicode correspondiente al carácter que representa, no el carácter en sí, por lo que puede operarse con caracteres como si se tratara de números enteros.

Por otra parte, a la hora de elegir el tipo de dato que vamos a utilizar ¿qué criterio seguiremos para elegir un tipo de dato u otro? Pues deberemos tener en cuenta cómo es la información que hay que guardar, si es de tipo texto, numérico, … y, además, qué rango de valores puede alcanzar. En este sentido, hay veces que aunque queramos representar un número sin decimales, tendremos que utilizar datos de tipo real.

Por ejemplo, el tipo de dato `int` no podría representar la población mundial del planeta, ya que el valor máximo que alcanza es de 2.147.483.647, siendo éste el número máximo de combinaciones posibles con 32 bits, teniendo en cuenta que la representación de los números enteros en Java utiliza complemento a 2. Si queremos representar el valor correspondiente a la población mundial del planeta, cerca de 7 mil millones de habitantes, tendríamos que utilizar al menos un tipo de dato `long`, o si tenemos problemas de espacio un tipo

`float`. Sin embargo, los tipos reales tienen otro problema: la precisión. Vamos a ver más sobre ellos en el siguiente apartado.

Para saber más

Si quieres obtener información sobre cómo se lleva a cabo la representación interna de números enteros y sobre la aritmética binaria, puedes consultar el siguiente enlace:

Aritmética binaria