

2.B. Tipos de datos.

1. Los tipos de datos.

1.2. Tipos de datos primitivos II.

El tipo de dato real permite representar cualquier número con decimales. Al igual que ocurre con los enteros, la mayoría de los lenguajes definen más de un tipo de dato real, en función del número de bits usado para representarlos. Cuanto mayor sea ese número:

- Más grande podrá ser el número real representado en valor absoluto
- Mayor será la precisión de la parte decimal

Entre cada dos números reales cualesquiera, siempre tendremos infinitos números reales, por lo que la mayoría de ellos los representaremos de forma aproximada. Por ejemplo, en la aritmética convencional, cuando dividimos 10 entre 3, el resultado es 3.333333..., con la secuencia de 3 repitiéndose infinitamente. En el ordenador sólo podemos almacenar un número finito de bits, por lo que el almacenamiento de un número real será siempre una aproximación.

Los números reales se representan en coma flotante, que consiste en trasladar la coma decimal a la primera cifra significativa del valor, con objeto de poder representar el máximo de números posible.

Un número se expresa como:

$$\text{Valor} = \text{mantisa} * 2^{\text{exponente}}$$

En concreto, sólo se almacena la mantisa y el exponente al que va elevada la base. Los bits empleados por la mantisa representan la precisión del número real, es decir, el número de cifras decimales significativas que puede tener el número real, mientras que los bits del exponente expresan la diferencia entre el mayor y el menor número representable, lo que viene a ser el intervalo de representación.

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. La mantisa es un valor entre -1.0 y 1.0 y el exponente representa la potencia de 2 necesaria para obtener el valor que se quiere representar. 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 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.

Autoevaluación

Relaciona los tipos primitivos con los bits y rango de valores correspondientes, escribiendo el número asociado en el hueco correspondiente.

Tipo Relación Característica

<code>short</code>	<input type="checkbox"/>	Coma flotante de 64 bits, usando la representación IEEE754-2008
<code>byte</code>	<input type="checkbox"/>	Entero de 32 bits, rango de valores de -2.147.483.648 (-2 ³¹) a 2.147.483.647 (+2 ³¹ -1)
<code>double</code>	<input type="checkbox"/>	Entero de 16 bits, rango de valores de -32.768 (-2 ¹⁵) a +32.767 (+2 ¹⁵ -1)
<code>long</code>	<input type="checkbox"/>	Coma flotante de 32 bits, usando la representación IEEE 745-2008
<code>int</code>	<input type="checkbox"/>	Entero de 8 bits, rango de valores de -128 (-2 ⁷) a +127 (+2 ⁷ -1)
<code>float</code>	<input type="checkbox"/>	Entero de 64 bits, rango de valores de -9.223.372.036.854.775.808 (-2 ⁶³) a 9.223.372.036.854.775.807 (+2 ⁶³ -1)

Resolver

Nota: Además de los anteriores, también son tipos primitivos el tipo de dato boolean, con valores true y false, y el tipo de datos char, que almacena el código Unicode de un carácter.