

Parámetros

Los parámetros o también llamado con el nombre de argumentos sirven como medio en el cual se puede declarar variables locales de funciones, hacia otras funciones la cual sirven para intercambiar información con los métodos. Los parámetros de un constructor o de un método puede ser de cualquier tipo, primitivos o referencias de objetos. Pueden servir para introducir datos o para obtener o modificar datos tras su ejecución.

Parámetros por referencia:

Cuando el argumento es de tipo referencia (por ejemplo, un array, un objeto,...) el paso por valor significa que no puede modificarse la referencia pero se pueden realizar llamadas a los métodos del objeto y modificar el valor asignado a las variables miembro accesibles del objeto durante la ejecución del método.

Donde se pasa una referencia a la variable original, manteniendo una referencia al objeto, por lo tanto cuando pasamos un objeto como parámetro se está realizando una copia de la referencia.

Parámetros por valor:

Es cuando el parámetro es de tipo primitivo, el paso por valor significa que durante la ejecución del método se reserva un nuevo espacio para el parámetro formal y no se puede modificar el parámetro real durante la ejecución del método. En resumen los parámetros por valor realizan una copia de las variables.

¿Cómo Java utiliza o maneja los parámetros por referencia?

En Java no existe el paso por referencia. El paso por valor (o por copia como algunos llaman) es obligatorio. Se preguntaran cuando se pasa un array por parámetros y se modifica desde el método al que se lo paso, este cambia, no se estará pasando una copia del array

Pareciera que nuestro argumento fallara, pero:

Lo que almacena en una variable no primitiva no es el objeto en sí sino una dirección o identificador del objeto en el espacio dinámico de memoria. Cuando pasas por parámetros la variable, estás pasando una copia de dicha dirección.