

Diferencia entre parámetros Por valor y Por referencia

Por valor: cuando los argumentos son pasados por valor a los métodos, significa que se realiza una copia de la variable y esta es enviada al método y no la original, entonces todos los cambios realizados dentro del método solo afectan a la copia actual.

Por referencia: cuando los argumentos son pasados por referencia, significa que la referencia o el puntero a la variable original son pasados a los métodos y no la data original.

Forma en que Java maneja los parámetros por referencia:

En Java, los manejos explícitos de memoria no existen, él es quien se encarga de realizar todo el trabajo sucio por ti (dejemos a los programadores preocuparse de cosas más importantes). Cuando tú instancias objetos en este lenguaje, dichos objetos se crean en algún lugar de la memoria; ellos no son algo que puedas tomar y guardarlo en una variable, necesitas de una entidad que te permita comunicarte con dicho objeto para poder trabajar sobre él. Esta entidad es precisamente lo que se conoce como **referencia** en Java.

Aunque las referencias de Java son totalmente diferentes de sus homónimas en C++, sí que guardan cierta similitud con los apuntadores. No obstante, veamos dos de las diferencias más importantes entre apuntadores y referencias de Java:

No hay aritmética de apuntadores

No es posible manipular el valor de una referencia de Java, es decir, no se le podrá sumar ni restar valor alguno a dicha referencia a fin de acceder a una región de memoria cercana a la que tenemos originalmente.

Las referencias tienen tipado fuerte

En Java no es posible que sus referencias se comporten como algo que no son. Por ejemplo, a una referencia de tipo *Object* solo se le podría realizar un *cast* a *String* si el objeto referido es efectivamente un *String*.

Para obtener la referencia de un objeto, Java utiliza el operador `new`; una vez obtenida, la podemos almacenar en una variable y, a partir de ese momento,

podemos utilizar a dicha variable como si prácticamente fuera el objeto que se ha creado en memoria.

Como te puedes dar cuenta, un apuntador, una referencia de C++ y una referencia de Java son, en esencia, la misma cosa. Los tres son meras implementaciones del concepto de referencia que definimos en un principio, es decir, son valores que se refieren a otro valor. Cada implementación tiene su propia particularidad, sus ventajas y desventajas (siendo la desventaja más importante el tener como nombre un término totalmente genérico).

Si tú pasas cualquiera de estas referencias a una función, no puedes estar equivocado, es un paso por referencia. Sin embargo, un último reto nos espera y radica en reconocer que, **dependiendo del tipo de referencia que se trate, esta también podría ser pasada por valor o por referencia**. En la siguiente sección trataremos este dilema.