

Luis Gerardo Estrada García

319013832

Paso de Parámetros por Valor y el Paso de Parámetros por Referencia

Antes de describir como son pasados los argumentos en Java, es mejor entender como Java almacena las variables en memoria. Básicamente hablamos de dos tipos de variables: primitivas y objetos.

Las variables primitivas siempre son almacenadas en la memoria stack, sin embargo, en el caso de los objetos estos son almacenados en dos etapas, los objetos actuales son almacenados en la memoria heap y la referencia a objetos se encuentra almacenada en la memoria stack que apunta a los objetos.

- **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 pasadas a los métodos y no la data original.

En Java, los argumentos son siempre pasados por valor independientemente del tipo de variable. Cada que el método es invocado, pasa lo siguiente:

- Una copia del argumento es pasada en la memoria stack y la copia es pasada en el método
 - Si la variable original es primitiva, es simple, una copia de la variable es creada en el stack y esta se pasa al método
 - Si la variable original no es primitiva, entonces una nueva referencia o apuntador es creado dentro de la memoria stack que apunta al objeto actual y la nueva referencia es pasada al método, (en esta etapa, 2 referencias apuntan al mismo objeto).

En Java, los argumentos son pasados por valor, la copia puede ser una referencia o una variable dependiendo del tipo original de la variable.