

Paso de Parámetros por Valor

Sucede cuando el control pasa a la función, los valores de los parámetros en la llamada se copian a objetos locales de la función, en este caso las variables declaradas dentro de la misma función.

- ☼ Recibe sólo una copia del valor que tiene la variable, o sea que no la puede modificar.
- ☼ Solo cambia las variables localmente (dentro de su función).
- ☼ Reciben como valores iniciales los valores de parámetros actuales.
- ☼ Otras modificaciones de parámetros diferentes a la función no la afectan.

Paso de Parámetros por Referencia

Las referencias son un tipo especial de variables que permiten alterar el contenido de otra variable. El mecanismo es simple, una variable referencia actúa como un alias de la variable referenciada y cambia valores en todo lugar sin importar si están dentro o fuera de su función. Las referencias:

- ☼ No son variables locales de la función ni nuevas variables, sólo son indicaciones (dependencias) a otra variable, sustituyen implícitamente la variable.
- ☼ Se deben actualizar en el momento de la declaración. (A qué variable pertenecen).
- ☼ Pasan la posición de memoria donde está guardada la variable, por lo que la función que emplean tiene datos de su valor y puede modificarla de cualquier manera.

Un parámetro por referencia tiene por objetivo modificar el contenido de una variable que se le envía a un procedimiento. Si por medio de un parámetro por valor llega un dato a un procedimiento, por un parámetro por referencia se retorna y sale un dato de un procedimiento.



Según el ejemplo de Java realizado, ¿Cómo se consigue que en el método con Paso de Parámetros por Valor modifique el valor en el método Main? Genera otro método que lo implemente y

Se consigue que el paso de parámetros por valor modifique el valor en el método main gracias al paso de parámetros por referencia, para implementarlo se debe hacer lo que en seguida describo.

Primero definimos e inicializamos un arreglo (que es una referencia), sabemos que el valor asignado en la variable apunta a una localidad de memoria en el arreglo que inicializamos, para ese momento el tipo de variable sigue marcando las referencias, pero cuando enviamos una función, entonces se convierte en el valor que ocuparemos, es decir:

Declaración e inicialización de referencia:

```
int [ ] arreglo = {81, 50, 30, 91, 90, 52};
```

Arreglo es la referencia, no es un valor de tipo primitivo int, pero apunta a la localidad en la memoria de los valores int que hay en él, los valores de este arreglo, recordando que los arreglos empiezan en la posición 0, serían los siguientes:

```
arreglo [ 0 ] = 81 ----> (Arreglo en 0 vale 81)
arreglo [ 1 ] = 50 ----> (Arreglo en 1 vale 50)
arreglo [ 2 ] = 30 ----> (Arreglo en 2 vale 30)
arreglo [ 3 ] = 91 ----> (Arreglo en 3 vale 91)
arreglo [ 4 ] = 90 ----> (Arreglo en 4 vale 90)
arreglo [ 5 ] = 52 ----> (Arreglo en 5 vale 52)
```