

Parámetros en Java

Java no tiene pasaje de parámetros por referencia. Los parámetros se pasan por valor.

Java no pasa objetos como parámetros, sino copias de las referencias a esos objetos.

El pasaje por referencia, y el pasaje de una referencia son dos cosas diferentes.

En Java no hay pasaje por referencia. El modificador var de Pascal no existe en JAVA.

Ejemplos Incorrectos

Se tiene una clase **Maximo** que provee un método de clase público `imprimirMaximo` que recibe un arreglo e imprime el máximo de los elementos.

```
Maximo.java  MaximoArre.java  MaximoTest.java  »16
1 package pruebas;
2
3 public class Maximo {
4     private static void buscarMaximo(int[] a, int max) {
5         for (int i=0;i<a.length;++i)
6             if (a[i]>max)
7                 max=a[i];
8     }
9     public static void imprimirMaximo(int[] a) {
10        int max = 0;
11        //el valor de "max" nunca se modifica
12        buscarMaximo(a, max);
13        // Siempre imprime 0
14        System.out.print("El máximo del arreglo es:"+max);
15    }
16 }
17
```

```
Maximo.java  MaximoTest.java  »17
1 package pruebas;
2
3 public class MaximoTest {
4     public static void main(String[] args) {
5         int[] arre = { 2, 13, 4, 5, 80, 3};
6         Maximo.imprimirMaximo(arre);
7     }
8 }
9
```

max es de tipo primitivo, lo que se pasa ahí es una copia del valor de **max**, es decir 0.
Sería lo mismo que hacer: `buscarMaximo(a, 0);`

¿Qué pasa si pasamos un objeto de tipo **Integer** en vez de un tipo primitivo **int**?

Ejemplos Incorrectos

Se tiene una clase **Maximo** que provee un método de clase público `imprimirMaximo` que recibe un arreglo e imprime el máximo de los elementos.

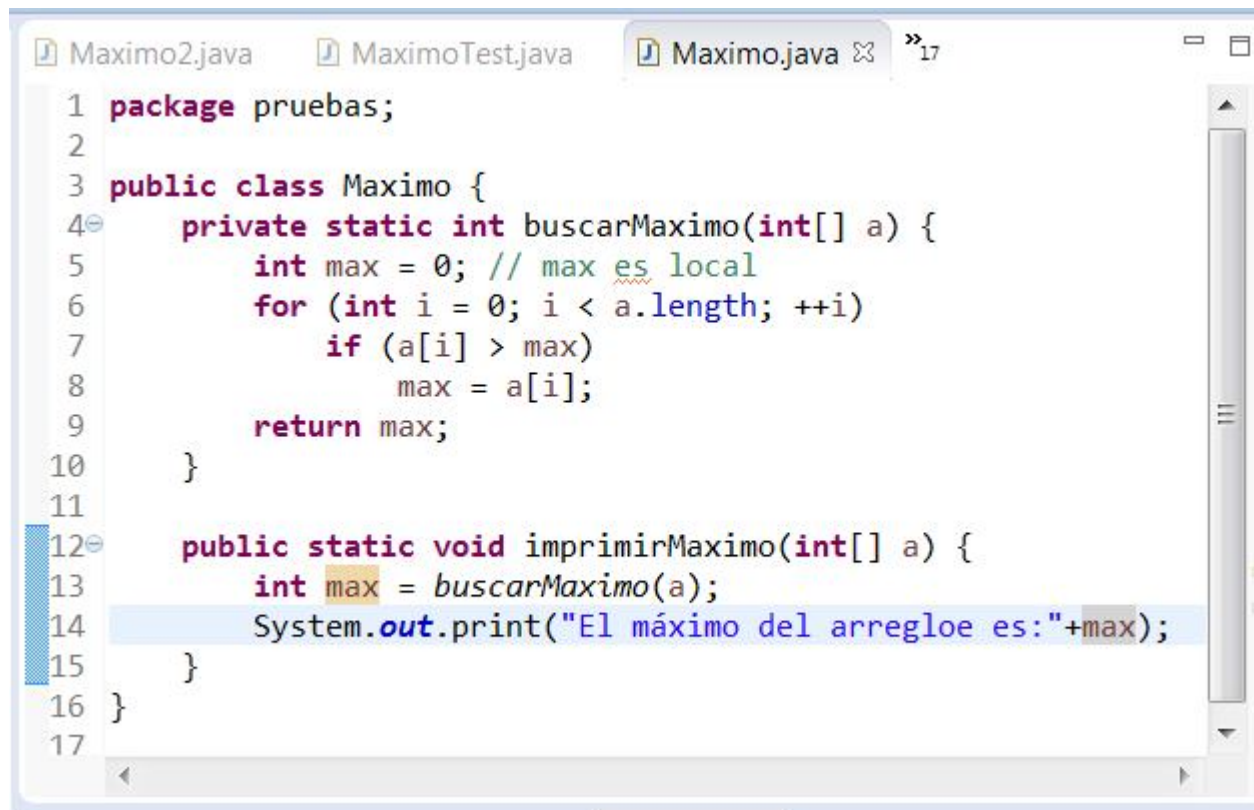
```
Maximo.java  MaximoArre.java  MaximoTest.java  »16
1 package pruebas;
2
3 public class Maximo {
4     private static void buscarMaximo(int[] a, Integer max) {
5         for (int i=0;i<a.length;++i)
6             if (a[i]>max)
7                 max=a[i];
8     }
9     public static void imprimirMaximo(int[] a) {
10        Integer max = 0;
11        //el valor de "max" nunca se modifica
12        buscarMaximo(a, max);
13        // Siempre imprime 0
14        System.out.print("El máximo del arreglo es:"+max);
15    }
16 }
17
```

```
Maximo.java  MaximoTest.java  »17
1 package pruebas;
2
3 public class MaximoTest {
4     public static void main(String[] args) {
5         int[] arre = { 2, 13, 4, 5, 80, 3};
6         Maximo.imprimirMaximo(arre);
7     }
8 }
9
```

Tampoco funciona !!! Los objetos de tipo **Integer** (y todos los *wrappers*) son inmutables

Un primer ejemplo correcto

Una solución de usar la sentencia **return** para devolver un valor. Esta es ideal pero no siempre es posible

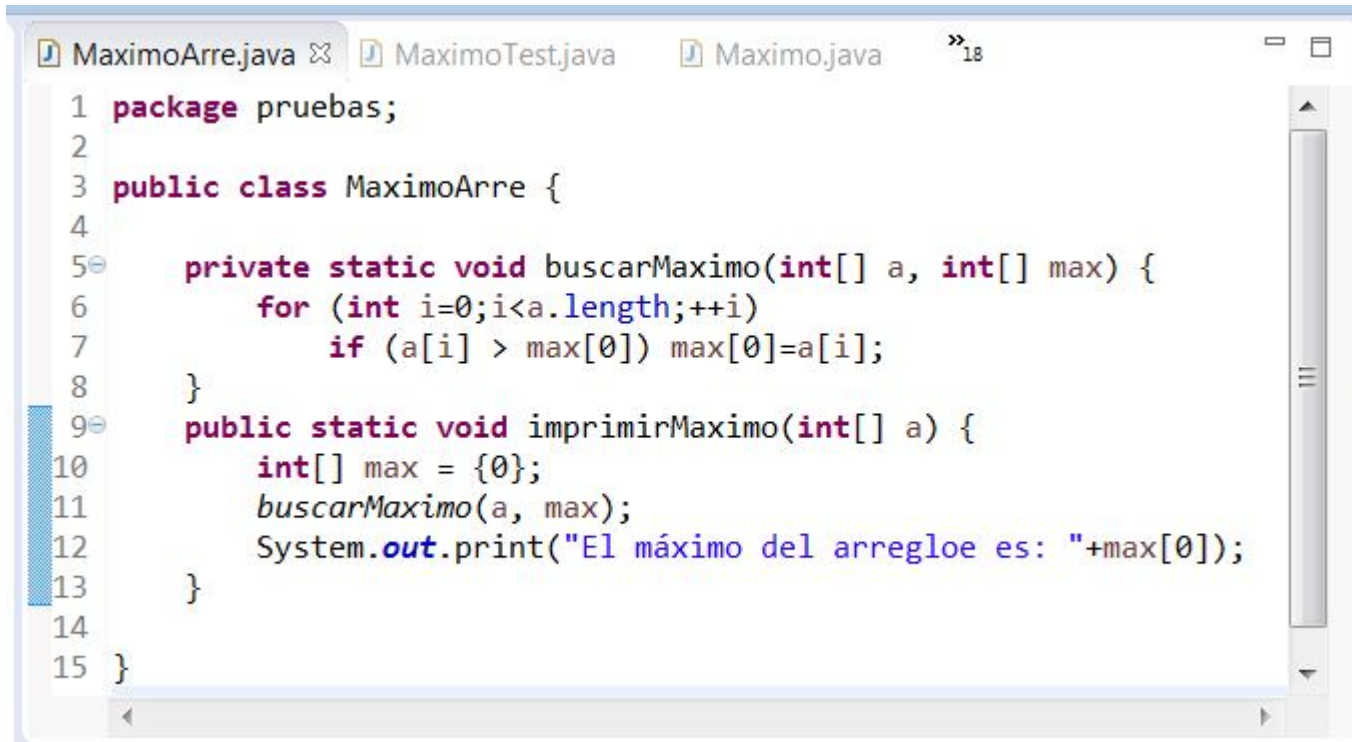


```
1 package pruebas;
2
3 public class Maximo {
4     private static int buscarMaximo(int[] a) {
5         int max = 0; // max es local
6         for (int i = 0; i < a.length; ++i)
7             if (a[i] > max)
8                 max = a[i];
9         return max;
10    }
11
12    public static void imprimirMaximo(int[] a) {
13        int max = buscarMaximo(a);
14        System.out.print("El máximo del arreglo es:" + max);
15    }
16 }
17
```

Otros ejemplos correctos

Otra solución podría ser usar un *objeto simple* para sumar

Usar un arreglo con un elemento para sumar. En este caso, lo que recibe `buscarmaximo(a, max)` en `max` es una copia de la referencia al arreglo. La referencia original y la recibida por parámetro son iguales, apuntan al mismo objeto. De esta manera el valor se actualiza.

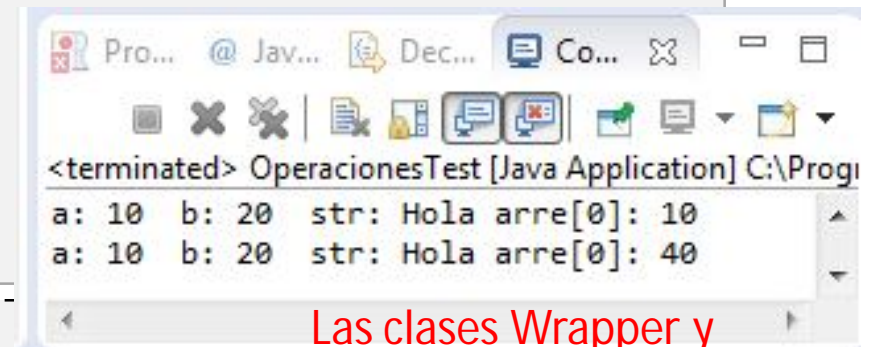


```
MaximoArre.java MaximoTest.java Maximo.java »18
1 package pruebas;
2
3 public class MaximoArre {
4
5     private static void buscarMaximo(int[] a, int[] max) {
6         for (int i=0; i<a.length; ++i)
7             if (a[i] > max[0]) max[0]=a[i];
8     }
9
10    public static void imprimirMaximo(int[] a) {
11        int[] max = {0};
12        buscarMaximo(a, max);
13        System.out.print("El máximo del arreglo es: "+max[0]);
14    }
15 }
```

Otro ejemplo

```
package parametros;
public class Operaciones {
    public static void cambiar(int x, Integer y, String palabra,
                               int[] arre) {

        x = x + 20;
        y = y + 20;
        palabra = palabra + "Juana";
        arre[0] = arre[0] + 30;
    }
}
```



Las clases Wrapper y
los String son inmutables

```
package parametros;
public class OperacionesTest {
    public static void main(String[] args) {
        int a = 10;
        Integer b = new Integer(20);
        String str = new String("Hola");
        int[] arre = {10};
        System.out.println("a: " + a + " b: " + b + " str: " + str);
        Operaciones.cambiar(a, b, str, arre);
        System.out.println("a: " + a + " b: " + b + " str: " + str);
    }
}
```