Programación Avanzada

Tarea 03: Ejercicios arreglos y memoria

2023-1

Valeria Jimeno Villegas - vjimenov@ciencias.unam.mx

7 de septiembre de 2022

I. Arreglos

Se tiene un arreglo de enteros de 3 elementos, `arregloUno`, que por *default* se inicializará con ceros por que no le indicamos que elementos contiene:

```
public class Arreglos {
        public static void main(String args[]) {
                // Creamos un primer arreglo
                int arregloUno[] = new int[3]; }}
```

Si ahora queremos crear un arreglo nuevo, `arregloDos`, a partir del arreglo `arregloUno`, es decir igualandolo:

```
                int arregloDos[] = arregloUno;
```

Cuando realizamos lo anterior lo que está sucediendo es **no** estamos creando una copia, en realidad estamos asignando una referencia o apuntador al mismo espacio de memoria. Por lo que al hacerle cambios a `arregloUno` estos cambios también se verán reflejados en `arregloDos` y viceversa.

```
                ...
                // Modificamos el arregloUno
                arregloUno[0] = 2;
                arregloUno[1] = 1;
                arregloUno[2] = 0;
```

```java
// Imprimimos ambos arreglos
System.out.println(Arrays.toString(arregloUno));
System.out.println(Arrays.toString(arregloDos));
```

```
// Output
[2, 1, 0] // arregloUno
[2, 1, 0] // arregloDos
```

Una vez dicho lo anterior, si queremos hacer una copia sin apuntar al mismo espacio de memoria, podemos iterar `arregloUno` y crear una copia de cada uno de los elementos. Con esto garantizamos que si alguno de los dos sufre algún cambio, este no se vera reflejado en el otro arreglo, ya que cada uno está apuntado a un espacio de memoria distinto.

```java
...
// ** Creamos una copia (manual)
int arregloUnoCopyManual[] = new int[3];
for(int i = 0; i < 3; i++){
        arregloUnoCopiaManual[i] = arregloUno[i];
}
// Modificamos el arregloUno
arregloUno[0] = 3;

// Imprimimos ambos arreglos
System.out.println(Arrays.toString(arregloUno));
System.out.println(Arrays.toString(arregloUnoCopyManual));
```

```
// Output
[3, 1, 0] // arregloUno
[2, 1, 0] // arregloUnoCopyManual
```