

Tarea 3

Programación avanzada

Emmanuel Peto Gutiérrez

7 de septiembre de 2022

1. Ejercicio arreglo

Primero se explicará qué pasa cuando se igualan los arreglos. Esto es

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

Cuando se ejecuta el comando `new int[3]` se está creando un arreglo, el cual llamaré *nai3* (aunque en realidad no tiene nombre), y se guarda en alguna dirección en la memoria. Del lado izquierdo del comando `int[] arregloUno = new int[3]`, se observa que se está declarando un arreglo llamado `arregloUno`, el cual apunta a *nai3*. Al hacer `arregloDos = arregloUno` lo que ocurre es que `arregloDos` está apuntando a *nai3*; es decir, `arregloUno` y `arregloDos` apuntan al mismo arreglo. Al modificar cualquier variable, `arregloUno` o `arregloDos`, en realidad se está modificando al arreglo *nai3*, por lo cual, al imprimir cualquiera de los dos se muestra lo mismo, aunque sólo se haya modificado uno.

Ahora se explicará el caso en el que se copia `arregloUno`. Es decir, en la línea:

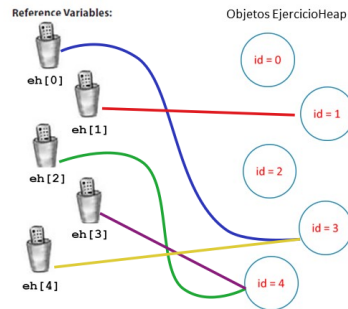
```
int[] arregloTres = arregloUno.clone();
```

Lo que ocurre dentro del método `clone()` es que se crea un arreglo nuevo, se copia cada elemento de `arregloUno` al nuevo y finalmente se devuelve el nuevo. El identificador `arregloTres` apunta al nuevo arreglo, que es una clon del arreglo apuntado por `arregloUno` pero no es el mismo (están en diferentes direcciones de memoria).

2. Ejercicio heap

Ejercicio Heap 2 (TAREA)

¿Cuál es el estado de estas variables con respecto a los objetos?



```
6 public class EjercicioHeap {
7     private int id = 0;
8     public static void main(String[] args) {
9         EjercicioHeap[] eh = new EjercicioHeap[5];
10        int x = 0;
11        while (x < 5) {
12            eh[x] = new EjercicioHeap();
13            eh[x].id = x;
14            x++;
15        }
16        eh[0] = eh[3];
17        eh[3] = eh[1];
18        eh[3].id = 6;
19        eh[2] = eh[4];
20        eh[4] = null;
21        eh[3] = eh[2];
22        eh[4] = eh[0];
23        // continua
24        System.out.println("fin");
25    }
26 }
```

El objeto que inicialmente tenía *id* 1, al final tiene *id* 6.