Módulo 9
Arreglos

Arreglos

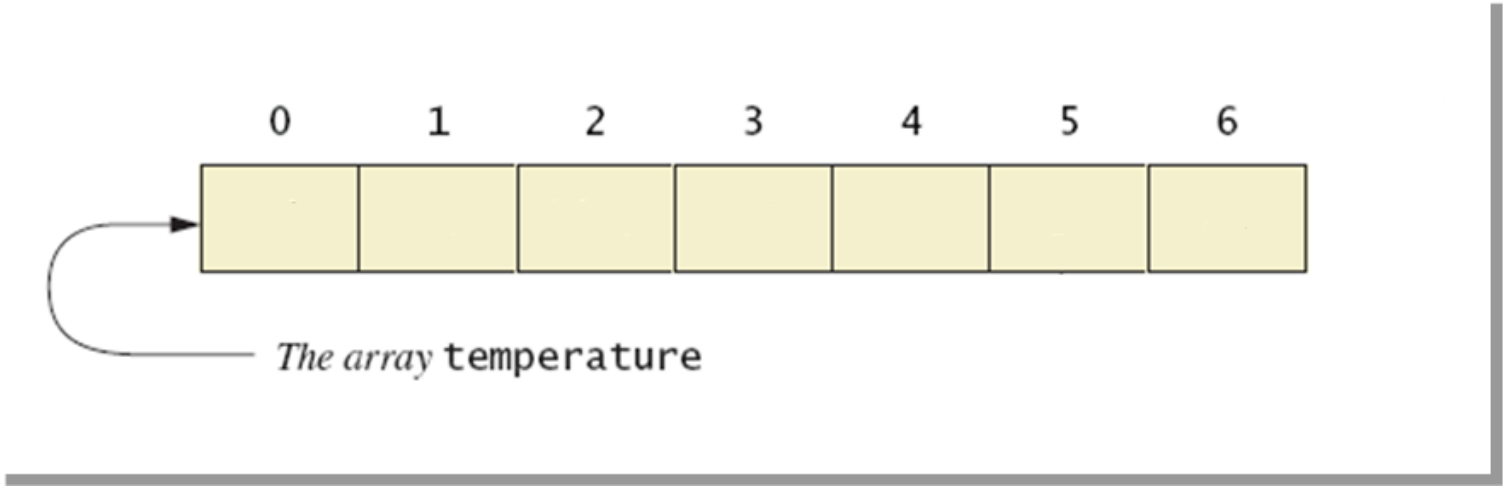
Un arreglo es una colección de variables del mismo tipo de datos. Para declarar un arreglo, podemos hacerlo de la siguiente forma:

```
int[] arr1; //declaration form #1
arr1 = new int[100]; //instantiation

int arr2[]; //declaration form #2
arr2 = new int[100]; //instantiation

int[] arr3 = new int[20]; //declaration and instatiation form #3

int arr4[] = new int[20]; //declaration and instatiation form #4
```

Visualizando Arreglos

Podemos visualizar un arreglo de la siguiente manera:

```
double[] temperature = new double[7];
```



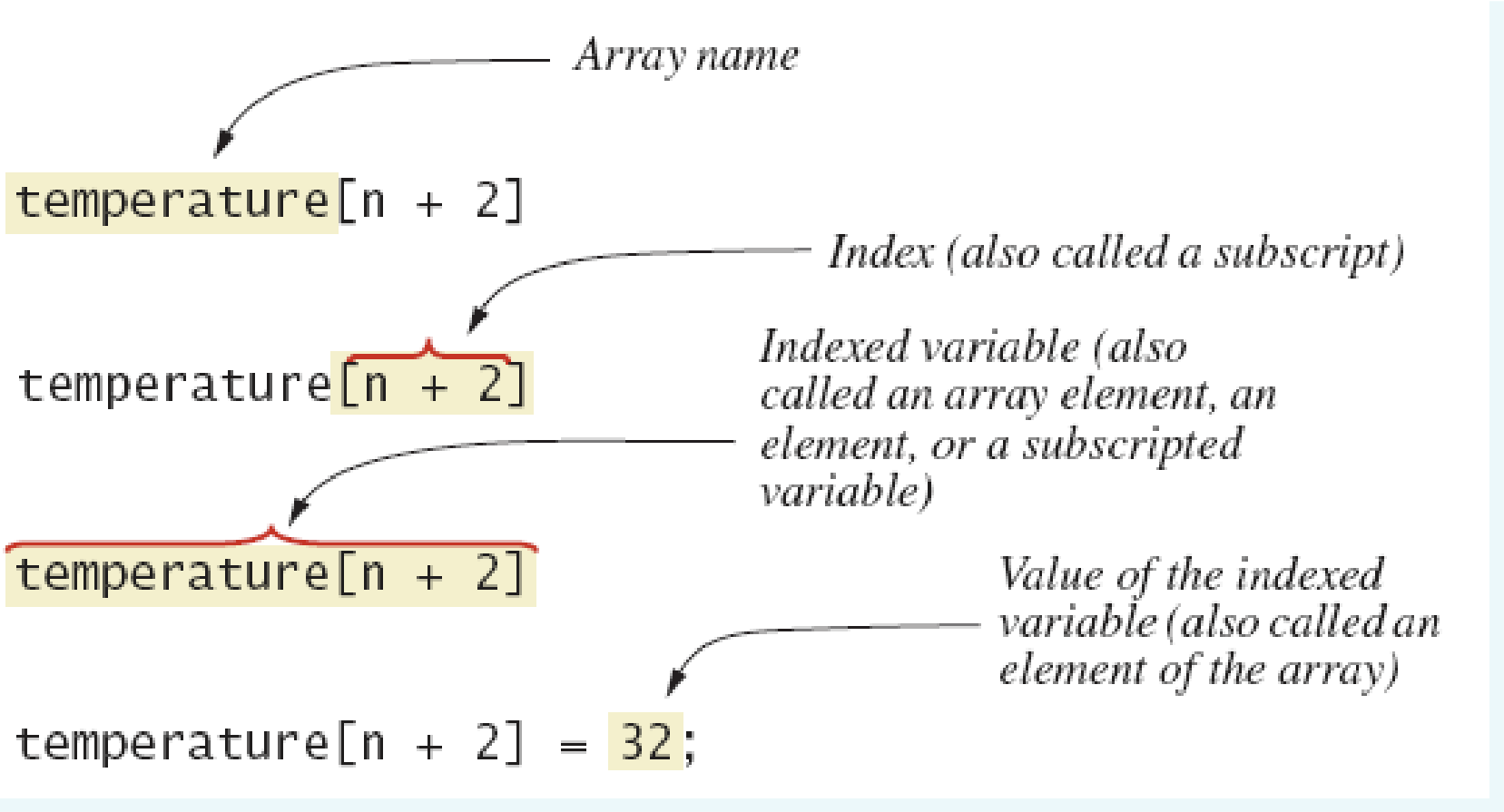
Sintaxis

Todos los arreglos deben estar asociado a un solo tipo de dato. No se puede combinar valores dentro de un mismo arreglo. La sintaxis para declarar e instanciar un arreglo es la siguiente:

```
data_type[] array_name = new data_type[array_size];

data_type array_name[] = new data_type[array_size];
```

El tamaño del arreglo debe ser un número positivo mayor o igual a cero.

Nomenclatura



Acceder un elemento de un arreglo

Para acceder un elemento de un arreglo, utilizamos los corchetes. La primera posición de un arreglo siempre será el índice 0. La última posición de un arreglo será `array_name.length - 1`.

```
double[] temperature = new double[7];
temperature[0] = 6.9; //first element index 0
temperature[1] = 30;
temperature[2] = 25.7;
temperature[3] = 26;
temperature[4] = 34;
temperature[5] = 31.5;
temperature[6] = 29; //last element index (temperature.length-1)
```

Acceder un elemento de un arreglo

Podemos inicializar un arreglo de cualquiera de las siguientes dos formas:

1. Instanciando el arreglo con el operador `new`, y escribiendo cada elemento del arreglo.
2. Instanciando explícitamente cada posición del arreglo.

1

```
double[] temperature = new double[7];
temperature[0] = 6.9;
temperature[1] = 30;
temperature[2] = 25.7;
temperature[3] = 26;
temperature[4] = 34;
temperature[5] = 31.5;
temperature[6] = 29;
```

2

```
double[] temperature = {6.9, 30, 25.7, 26, 34, 31.5, 29};
```

Tamaño del arreglo

Para conocer el tamaño de un arreglo, podemos utilizar el atributo `length`.

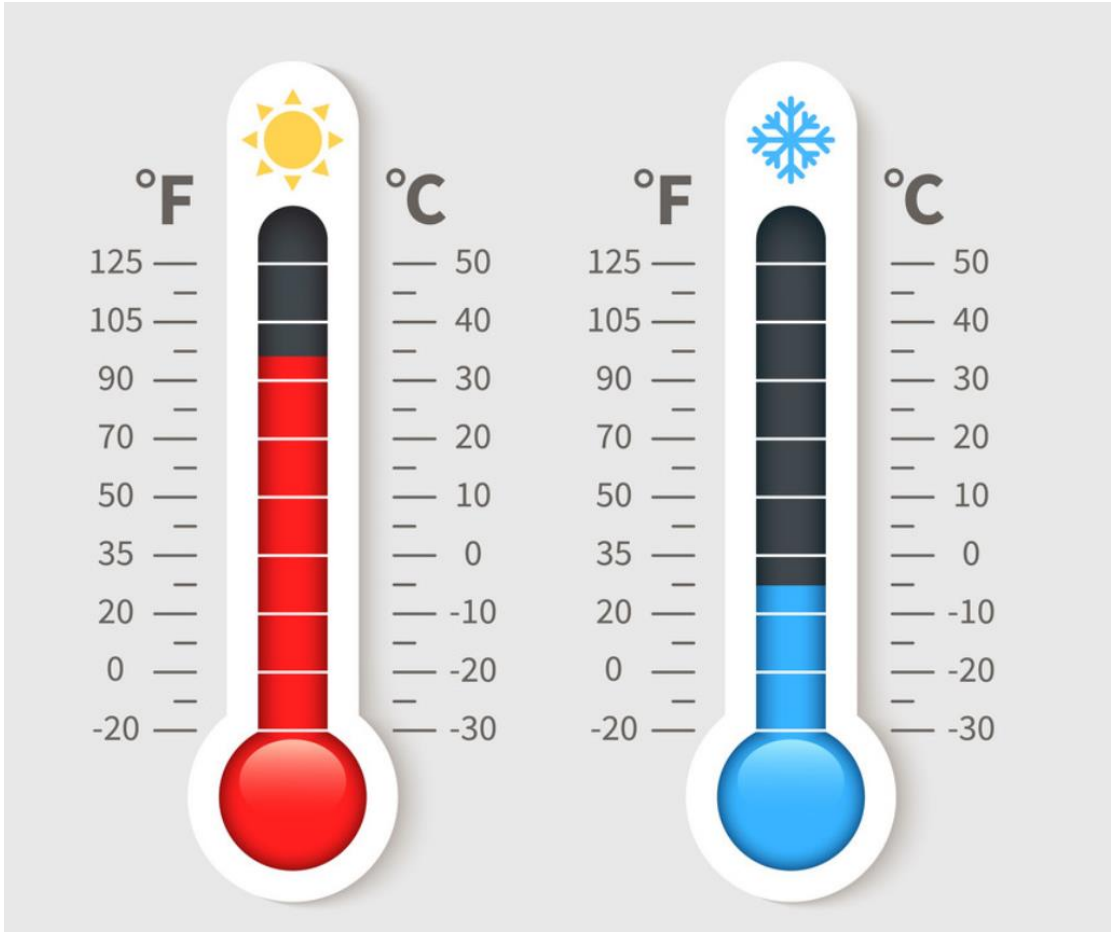
El tamaño de un arreglo es final, lo que significa que no es modificable después de su instanciación.

```
double[] temperature = new double[7];
int len = temperature.length;
System.out.println(len);  //prints 7
```

El último elemento de un arreglo está en el índice `length - 1`.

```
temperature[temperature.length - 1] = 29;
```

Ejercicio!



Crea un programa que lea 7 temperaturas del teclado, e indique cuáles estén por encima y por debajo del promedio.

```java
import java.util.*;

public class ArrayOfTemperatures {
    public static void main(String[] args) throws Exception {
        double[] temperatures = new double[7];
        Scanner keyboard = new Scanner(System.in);
        System.out.println("Enter 7 temperatures");

        double sum = 0;
        for(int i=0; i<temperatures.length; i++){
            temperatures[i] = keyboard.nextDouble();
            sum = sum + temperatures[i];
        }

        double average = sum / 7;
        System.out.println("The average is: " + average);
        System.out.println("The temperatures are:");
        for(int i=0; i<temperatures.length; i++){
            if (temperatures[i] > average){
                System.out.println(temperatures[i] + " is above average.");
            }
            if (temperatures[i] < average){
                System.out.println(temperatures[i] + " is below average.");
            }
            if (temperatures[i] == average){
                System.out.println(temperatures[i] + " is the average.");
            }
        }
        keyboard.close();
    }
}
```

Nuevos tipos de ciclos! FOR-EACH

```java
int[] numbers = {1,2,3,4,5,6,7,8,9,10};
for (int item : numbers) {
    System.out.println("Count is: " + item);
}
```

Este ciclo iterará sobre cada elemento del arreglo numbers, asignando el contenido en la variable `int item`.

```
Count is: 1
Count is: 2
Count is: 3
Count is: 4
Count is: 5
Count is: 6
Count is: 7
Count is: 8
Count is: 9
Count is: 10
```

Arreglos son objetos

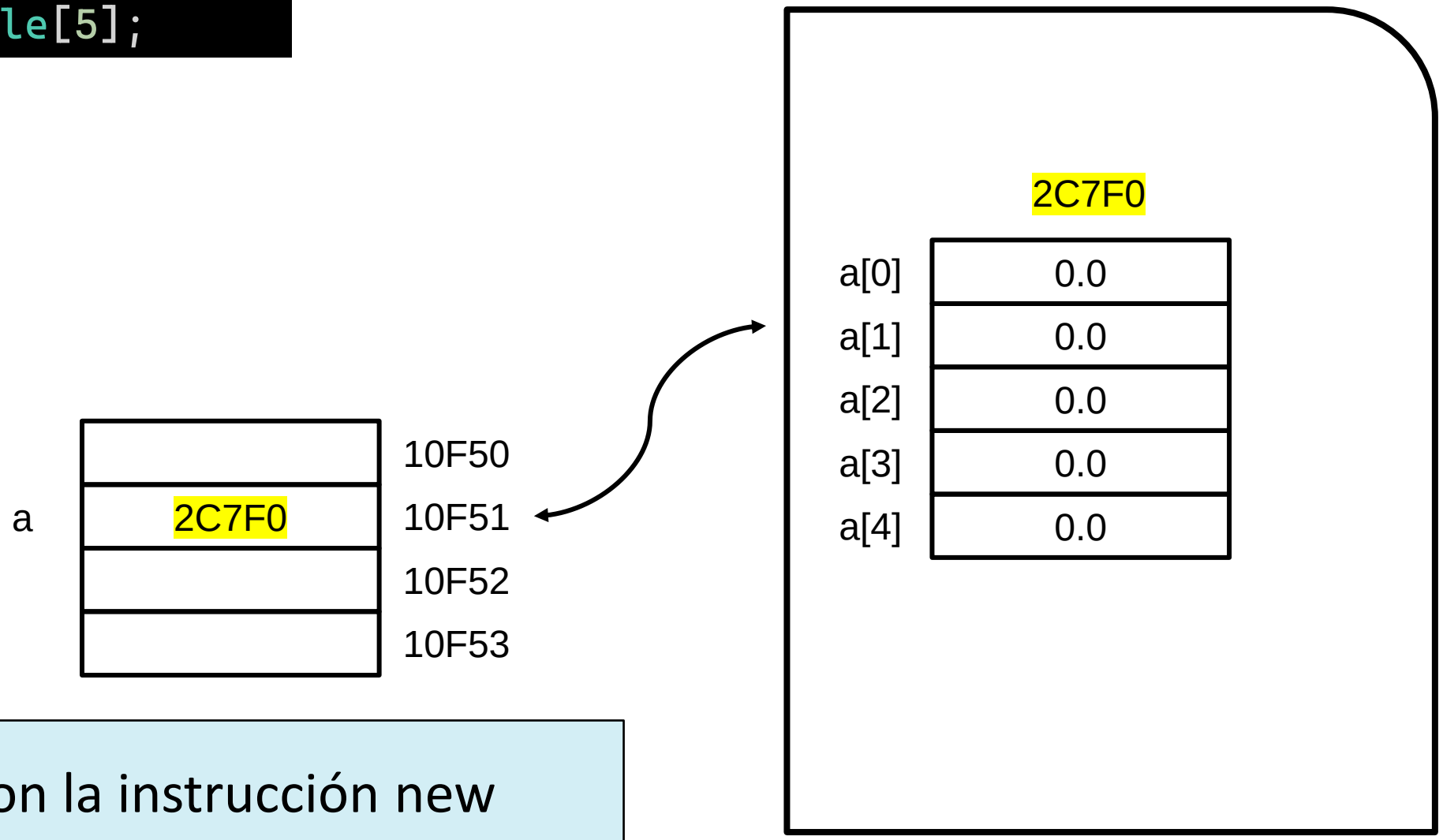
Un arreglo es un objeto, por lo que cuando hacemos referencia a él en realidad estamos accediendo a una dirección de memoria que apunta a la lista de elementos.

Veamos el siguiente ejemplo:

```
double[] a;
```

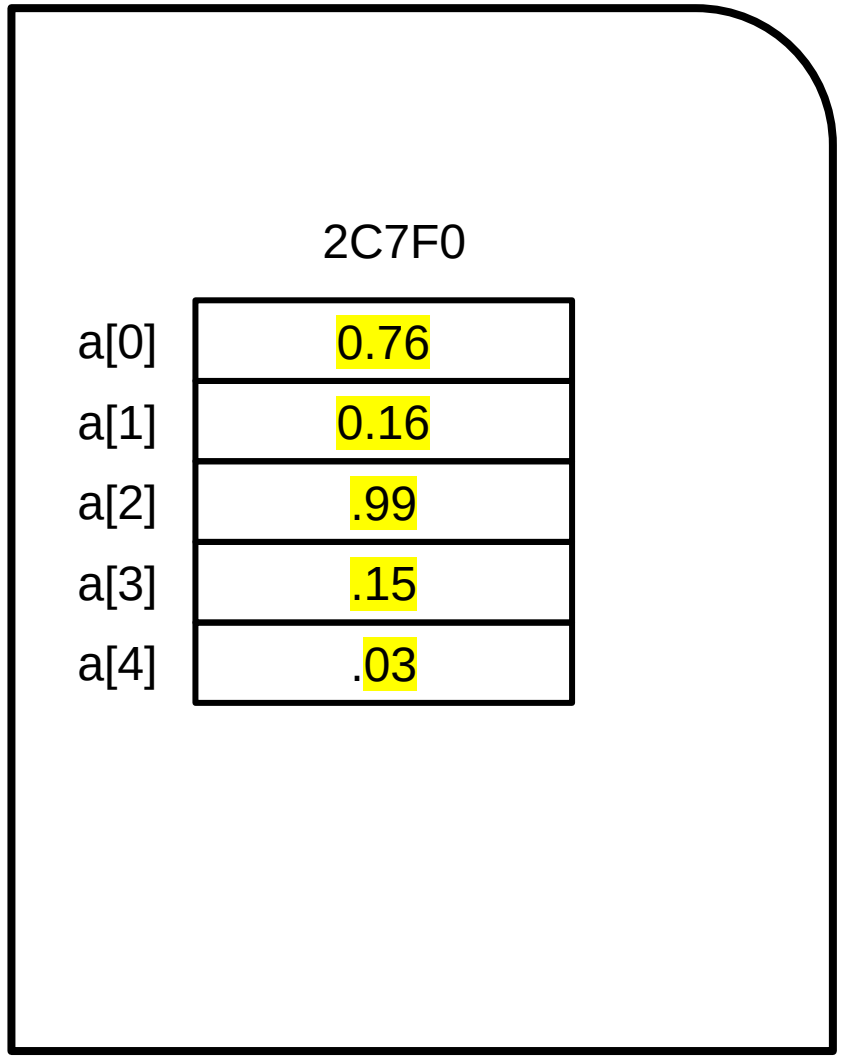
Al inicializar la variable a, se reserva una espacio para almacenar una referencia a un arreglo.

```
double[] a;
a = new double[5];
```



Con la instrucción new instanciamos un arreglo de 5 posiciones, en la dirección 2C7F0.

```java
double[] a;
a = new double[5];
a[0] = Math.random();
a[1] = Math.random();
a[2] = Math.random();
a[3] = Math.random();
a[4] = Math.random();
```



Generamos valores para cada elemento del arreglo.

¿Y EN QUÉ NOS AFECTA?

Métodos y arreglos

Al crear métodos que tengan arreglos como parámetros de entrada, estamos transfiriendo la dirección de memoria.

```java
public static void main(String[] args){

        int a[] = {1,5,7};
         System.out.println("Before:");
         for(int i1: a){
             System.out.println(i1);
         }

         doSomething(a);

         System.out.println("After:");
         for(int i1: a){
             System.out.println(i1);
         }
    }

    public static void doSomething(int[] arr1){
         for(int i = 0; i<arr1.length; i++){
             arr1[i]++;
         }
    }
```

```
Before:
1
5
7
After:
2
6
8
```

Métodos y arreglos

Al hacer la llamada del método:

```
doSomething(a);
```

Estamos transfiriendo la dirección de memoria del arreglo a, por lo que el método modifica el contenido del arreglo también!

```
public static void doSomething(int[] arr1){
    for(int i = 0; i<arr1.length; i++){
        arr1[i]++;
  }
}
```