1️⃣

Arrays en Java

Un **array** o matriz es como **una variable que puede contener valores múltiples**, a diferencia de una variable regular que solo puede contener un único valor.

- En Java, todas las matrices se asignan estáticamente.
- Como las matrices/arrays son objetos en Java, cada array tiene asociado una variable de instancia de longitud (length) que contiene la cantidad de elementos que la matriz puede contener. (En otras palabras, **length** contiene el tamaño de la matriz.)
- Una variable array en Java se declara como otras variables con corchetes **[]** después del tipo de datos.
- Las variables en el array están ordenadas y cada una tiene un índice que **comienza desde 0**.
- El array Java también se puede usar como un campo estático, una variable local o un parámetro de método.
- El tamaño de un array debe especificarse mediante un **valor int** y no, long o short.
- El array puede contener tipos de datos primitivos, así como también objetos de una clase según la definición del array.

▼ Declaración de un array

La declaración de un array tiene dos componentes: **el tipo y el nombre**.

Tipo declara el tipo de elemento del array. El tipo de elemento determina el tipo de datos de cada elemento que comprende la matriz. Al igual que la matriz de tipo int, también podemos crear una matriz de otros tipos de datos primitivos como char, float, double..etc o tipo de datos definido por el usuario (objetos de una clase). Por lo tanto, el tipo de elemento para la matriz determina el tipo de datos que la matriz contendrá.

```
//Forma general de declarar un arreglo
tipo nombreArray[];
tipo[] nombreArray;
```

```
//Ejemplo seudocodigo
public void main(String args[]) {
  //Ambas son declaraciones validas
  int intArray[];
  int[] intArray2;

  //Tipo de datos primitivos
  byte byteArray[];
  short shortArray[];
  boolean booleanArray[];
  long longArray[];
  float floatArray[];
  double doubleArray[];
  char charArray[];

  //Tipos de datos definidos por el usuario
  //Una serie de referencias a objetos de la clase MiClase(creada por el usuario)
  MiClase miClaseArray[];
}
```

▼ Instancia de un array

Cuando un array se declara, solo se crea una referencia del array. Para realmente crear o dar memoria al array (a partir de aquí solo mencionaré a array, y no matriz o arreglo), puede crear un array de la siguiente manera:

```
tipo[] nombreArray=new tipo[tamaño];
```

- tipo → especifica el tipo de datos que se asignara
- tamaño → específica el número de elementos en el array
- nombreArray → nombre de la variable del array vinculado al mismo
- Es decir, para usar **new** para asignar un array, **debe especificar el tipo y la cantidad de elementos a asignar**.

```
int intArray[]; //declaracion de un array
intArray=new int[20]; //asignando memoria al array
int[] intArray=new int[20]; //combinando ambas declaraciones en una
```

En una situación en la que ya se conoce el tamaño y los elementos del array se pueden usar literales del array.

```
//Declarando un array literal
int[] intArray=new int[]{1,2,3,4,5,6,7,8,9,10};
```

- La longitud de este array determina la longitud del array creado.
- No es necesario escribir **new int[]** en las últimas versiones de Java.

▼ Accediendo a los elementos del array usando el bucle for

A cada elemento del array se accede a través de su índice. El índice comienza con 0 y termina en (tamaño total del array -1). Se puede acceder a todos los elementos de la matriz usando el bucle for en Java.

```
//Ejemplo seudocodigo
public void main(String args[]) {
  int array[]=new int[]{1,2,3,4,5,6,7,8,9,10};
  //acceder a los elementos del array
  for(int i=0;i<array.length;i++) {
    System.out.println("Elemento en el indice "+i+" : "+array[i]);
  }
}
```