

UD3

3.1 – Arrays en Java



CONTENIDOS

- **Unidad 3: Arrays lineales**
 1. **Arrays en Java**
 2. **Operaciones con Arrays**
 3. **Foreach y envoltorio Arrays**
 4. **Objetos, atributos y métodos**

ÍNDICE

- **Arrays en Java**
 1. Introducción
 2. Estructura
 3. Declaración
 4. Características
 5. Ejercicios

Introducción

- Un tema que aparece frecuentemente en la programación es el de organizar una colección de datos en memoria.
- Este tema también es conocido con el título Estructuras de datos, y se relaciona también a distintos algoritmos para agilizar el acceso.



Introducción

- Hasta ahora, hemos utilizado variables (atributos, parámetros o variables locales) para almacenar los datos que nos hacían falta. Imagina que en un programa nos piden almacenar la nota de 100 alumnos que van a hacer un examen y además nos piden que obtengamos la nota media. No parece muy operativo tener 100 variables en nuestro programa y si además el número de alumnos aumentase hasta 200 o 1000... ¿qué harías?



- La solución a esto se llama arrays o vectores (son sinónimos).

Estructura

- Un **array** se compone de una serie de posiciones consecutivas en memoria.
- A los vectores se accede mediante un índice: **mi_vector [índice]**

array										
índice	0	1	2	3	4	5	6	7	8	9
contenido	13	21	34	92	0	6	76	4	0	84

matriz									
índice	0	1	2	3	4	5	6	7	8
índice	76	4	0	84	21	34	92	0	6
0	1	2	3	4	5	6	7	8	9
1	21	34	92	0	6	76	4	0	84

- Los **arrays** pueden ser de varias dimensiones. En la anterior imagen podemos ver uno de 1 dimensión y otro de dos dimensiones. Los primeros suelen denominarse vectores y los segundos matrices.

Declaración

- En Java se pueden declarar vectores de dos formas diferentes:

```
<tipo> [] <nombre>;  
<tipo> <nombre> [];
```

- Por ejemplo, para declarar un array de números enteros utilizaremos la siguiente sentencia:

```
int [] sueldos;
```

- El array aún no ha sido creado, sino meramente declarado. Java trata los vectores como si fueran objetos, por lo tanto, para crear el Array (reservar su memoria e inicializarlo) deberemos recurrir al operador new:

```
sueldos = new int [5];
```

- Por ejemplo, si queremos tener los siguientes valores:

sueldos				
1200	750	820	550	490
Sueldos[0]	Sueldos[1]	Sueldos[2]	Sueldos[3]	Sueldos[4]

Declaración

- Por ejemplo, si queremos tener los siguientes valores:

```
1 public class Declaracion {  
2  
3     public static void main(String[] args) {  
4         //Declaración  
5         int [] sueldos;  
6  
7         //Dimensionamiento  
8         sueldos = new int [5];  
9  
10        //Inicialización  
11        sueldos [0] = 1200;  
12        sueldos [1] = 750;  
13        sueldos [2] = 820;  
14        sueldos [3] = 550;  
15        sueldos [4] = 490;  
16    }  
17 }
```

CURSO INICIACIÓN JAVA
Tema 3 – Arrays y Strings
Parte 1 – Arrays -
Introducción



- Pero también hay una forma alternativa y cómoda de hacerlo cuando ya conocemos los valores del array:

```
1 public class Declaracion {  
2  
3     public static void main(String[] args) {  
4         //Todo en uno  
5         int [] otrosSueldos = {1200, 750, 820, 550, 490};  
6     }  
7 }
```

- De esta forma se declara e inicializa en una única instrucción, donde no hace falta utilizar new y se dan todos los valores juntos.

Declaración

Declare an Array Containing the First Five Primes

in one Line

(The first five primes are: 2, 3, 5, 7, 11)



Características

- Se puede acceder al número de elementos de un array gracias al atributo **length**. Por ejemplo, `vect.length` o en el caso del array de sueldos, `sueldos.length`.
- Es recomendable utilizar siempre este atributo en lugar de un número, para acceder al límite superior del índice.
- El índice de un array empieza siempre en 0 y termina en `length-1`
- Todos los elementos del array son del mismo tipo, por lo que no se pueden mezclar datos de diferentes tipos en el mismo array.
- Una vez dimensionado el array, no se puede modificar la cantidad de elementos que contiene.
- Si tratamos de acceder a una posición que no existe, el programa dará el famoso error: `ArrayIndexOutOfBoundsException`, que significa: Índice fuera de rango.

Ejercicio 1/6

Arrays of Strings

How would you declare an array of strings with space for 10 values?

How would you declare an array of strings that contains two values: "yes" and "no"?

Ejercicio 2/6

Put 10 in the First and Last Spaces

```
int[] values = new int[10];
```

Write statements that put the value 10
into the spots with the lowest index and
the highest index.



Ejercicio 3/6

What is printed by this segment of code?

```
int[] temperatures = {65, 71, 68, 85, 87, 89, 78};  
int[] dailyTemps = temperatures;  
dailyTemps[6] = 101;  
System.out.println(temperatures[6]);
```



Ejercicio 4/6

```
int m = 5;  
int [] a = new int[5];
```

0	0	0	0	0
a[0]	a[1]	a[2]	a[3]	a[4]

```
a[1] = 2;
```

0	2	0	0	0
a[0]	a[1]	a[2]	a[3]	a[4]

```
a[2] = a[1];
```

0	2	2	0	0
a[0]	a[1]	a[2]	a[3]	a[4]

```
a[0] = a[1] + a[2] + 2;
```

6	2	2	0	0
a[0]	a[1]	a[2]	a[3]	a[4]

```
a[0]++;
```

7	2	2	0	0
a[0]	a[1]	a[2]	a[3]	a[4]

```
int m = 5;  
a[3] = m + 10;
```

7	2	2	15	0
a[0]	a[1]	a[2]	a[3]	a[4]

Ejercicio 5/6

- Dado el siguiente programa:

```
1 public class Caracteristicas {  
2  
3     public static void main(String[] args) {  
4         int m = 5;  
5         int[] vector = new int[m];  
6         vector[1] = 2;  
7         vector[2] = vector[1];  
8         vector[0] = vector[1] + vector[2] + 2;  
9         vector[0]++;  
10        vector[3] = m + 10;  
11        vector[vector.length-1] = vector[0];  
12    }  
13 }
```

- Tras ejecutarse todas las instrucciones, ¿qué hay en la posición 4 del vector?
 - a) 2
 - b) 7
 - c) 15
 - d) Esa posición no existe en el vector

Ejercicio 6/6

- Dado el siguiente programa:

```
1 public class Caracteristicas {  
2  
3     public static void main(String[] args) {  
4         String[] laborables =  
5             { "lunes", "martes", "miercoles", "jueves", "viernes" };  
6         int actual = 3;  
7         int vacaciones = 22;  
8         int finalVacaciones = (actual + vacaciones) % 5;  
9         System.out.println(laborables[finalVacaciones]);  
10    }  
11 }
```

- ¿Qué imprimirá el siguiente programa?

- a) lunes
- b) martes
- c) miercoles
- d) jueves

CURSO INICIACIÓN JAVA
Tema 3 – Arrays y Strings
Parte 2 – Arrays -
Características



APUNTE...

The screenshot shows the homepage of the KeepCoding Tech School website. At the top, there is a navigation bar with links: Nosotros, Alumnos, Bootcamps, Career Services, Recursos, and Contáctanos. A search icon is also present. The main title "KEEPCODING" is displayed with a small logo of a person coding. Below the title, a large banner features the text "Blog de Programación y Tecnología" and "Noticias e información techie en un solo lugar". A search bar with the placeholder "¿Buscas algo en específico?" and a magnifying glass icon is located below the banner. At the bottom, there are several orange buttons with white text, each representing a different technology or field: Fundamentos de Programación, Inteligencia Artificial & Machine Learning, Desarrollo de Apps Móviles, Desarrollo Web, DevOps, Ciberseguridad, Marketing Digital, Cloud Computing, and Big Data.

KEEPCODING
Tech School

Nosotros ▾ Alumnos ▾ Bootcamps ▾ Career Services ▾ Recursos ▾ Contáctanos

Blog de Programación y Tecnología

Noticias e información techie en un solo lugar

¿Buscas algo en específico?

</> Fundamentos de Programación

</> Inteligencia Artificial & Machine Learning

</> Desarrollo de Apps Móviles

</> Desarrollo Web

</> DevOps

</> Ciberseguridad

</> Marketing Digital

</> Cloud Computing

</> Big Data