

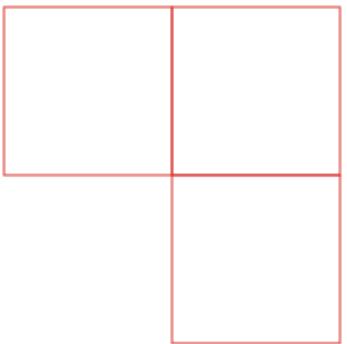
**2**

## Operaciones con Arrays



Ve más allá





# CONTENIDOS

## ■ Unidad 3: Arrays lineales

1.Arrays en Java

## 2.Operaciones con Arrays

3.Foreach y envoltorio Arrays

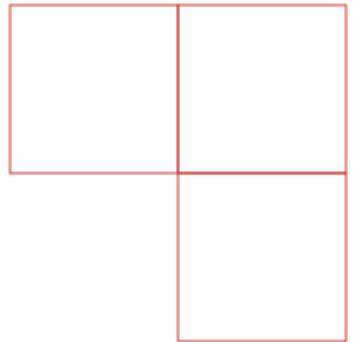
4.Objetos, atributos y métodos

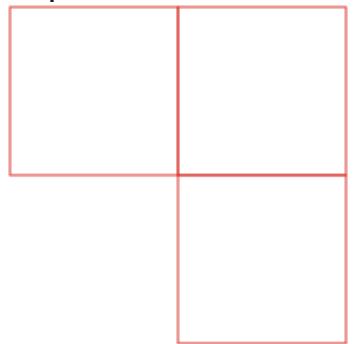




# INDICE

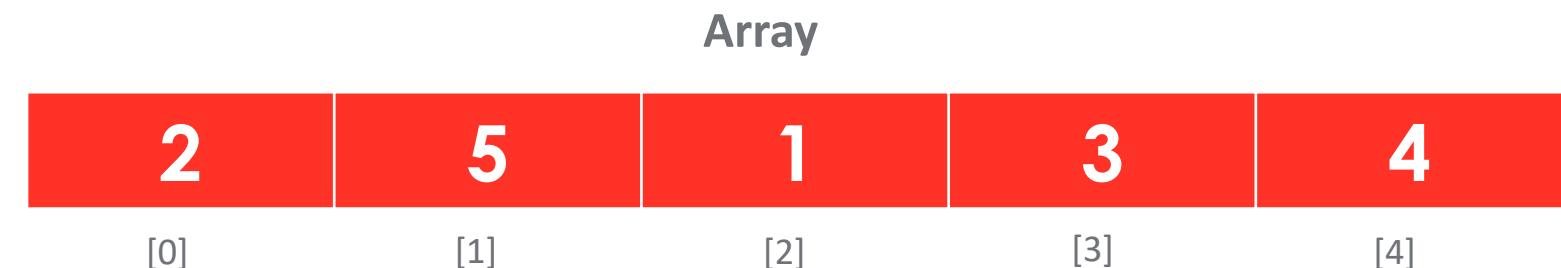
- Operaciones con arrays
  - 1.Imprimir un Array
  - 2.Sumar un Array
  - 3.Buscar en un Array
  - 4.Ejercicio

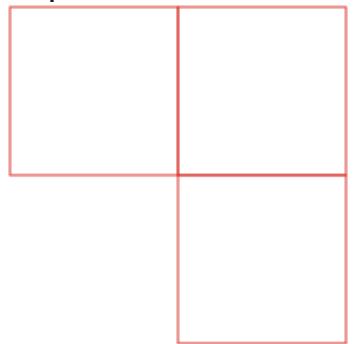




# Imprimir un Array

- Cuando queremos imprimir el contenido de un Array, lo habitual es recorrerlo con un bucle for.
- También es posible utilizar otras sentencias de iteración como while o do-while, pero el for es más cómodo.
- A la hora de imprimir un Array, hay que acceder al contenido de cada uno de sus elementos utilizando los corchetes.
- Por ejemplo, para imprimir el siguiente array:

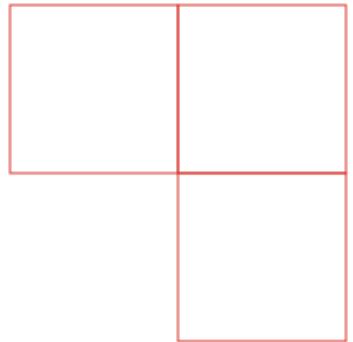




# Imprimir un Array

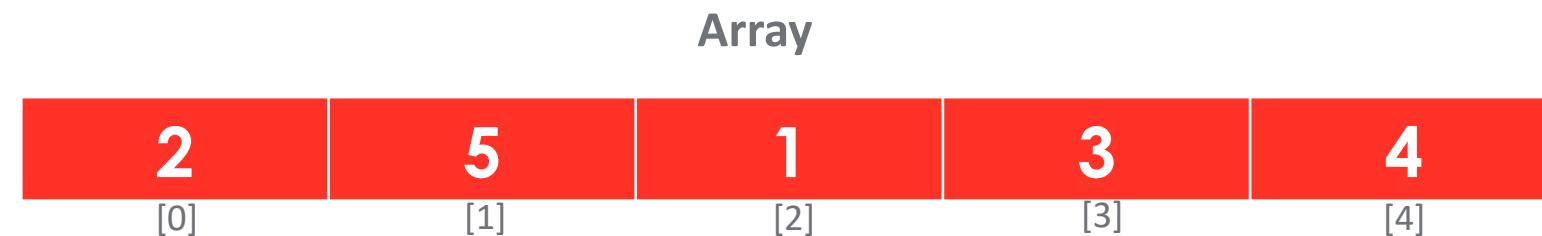
- Se podría utilizar el siguiente código

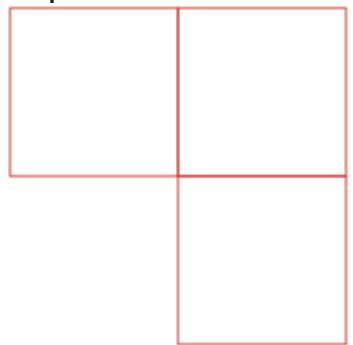
```
1 public class Ejemplos {  
2  
3     private void imprimir(int[] arrayEjemplo) {  
4         for (int i = 0; i < arrayEjemplo.length; i++) {  
5             System.out.print(arrayEjemplo[i] + " ");  
6         }  
7         System.out.println("");  
8     }  
9  
10    public static void main(String[] args) {  
11        int [] arrayEjemplo = {2,5,1,3,4};  
12        Ejemplos programa = new Ejemplos();  
13        programa.imprimir(arrayEjemplo);  
14    }  
15 }
```



# Sumar un Array

- Es muy habitual que haya que utilizar los elementos de un array para sumarlos, obtener la media o realizar otro tipo de cálculos.
- Como en el caso anterior, habrá que recorrer el array con una sentencia for y utilizar una variable de tipo acumulador para ir almacenando los datos según se recorren.
- Por ejemplo, para sumar el siguiente array:

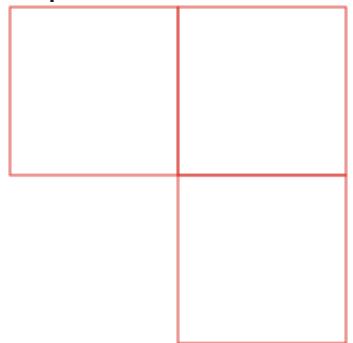




# Imprimir un Array

- Se podría utilizar el siguiente código

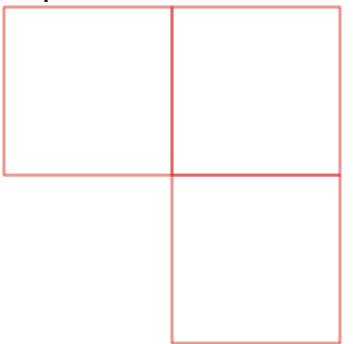
```
1 public class Ejemplos {  
2  
3     private int sumaTotal(int[] arrayEjemplo) {  
4         int suma = 0;  
5         for (int i = 0; i < arrayEjemplo.length; i++) {  
6             suma = suma + arrayEjemplo[i];  
7         }  
8         return suma;  
9     }  
10  
11    public static void main(String[] args) {  
12        int [] arrayEjemplo = {2,5,1,3,4};  
13        Ejemplos programa = new Ejemplos();  
14        int suma=0;  
15        suma = programa.sumaTotal(arrayEjemplo);  
16        System.out.println("La suma de los numeros es: " + suma);  
17    }  
18 }
```



# Buscar en un Array

- Una operación que se realiza con bastante frecuencia es la de recorrer un array en busca de algún dato. Se puede también buscar el mayor, el menor o algún otro dato que cumpla alguna condición.
- Por ejemplo, para buscar el mayor elemento en el siguiente array:





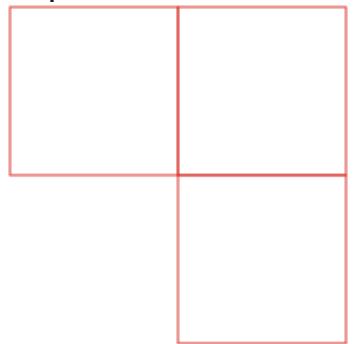
# Buscar en un Array

- Se podría utilizar el siguiente código:

```
1 public class Ejemplos {  
2  
3     private int mayor(int[] arrayEjemplo) {  
4         int mayor = Integer.MIN_VALUE;  
5         for (int i = 0; i < arrayEjemplo.length; i++)  
6             if (mayor < arrayEjemplo[i])  
7                 mayor = arrayEjemplo[i];  
8         return mayor;  
9     }  
10  
11    public static void main(String[] args) {  
12        int [] arrayEjemplo = {2,5,1,3,4};  
13        Ejemplos programa = new Ejemplos();  
14        int mayor=0;  
15        mayor = programa.mayor(arrayEjemplo);  
16        System.out.println("El mayor de los numeros es: " + mayor);  
17    }  
18 }
```

**CURSO INICIACIÓN JAVA**  
**Tema 3 – Arrays y Strings**  
Parte  **Arrays –**  
**Imprimir, recorrer, buscar**



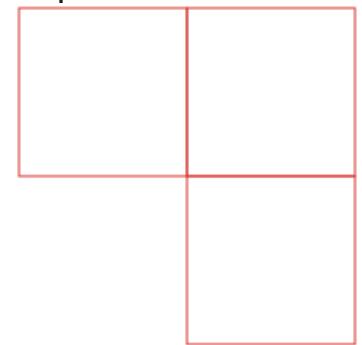


# Ejercicio 1

- Queremos realizar un programa que recorra el contenido de un array para saber si todos sus elementos son pares o no.
- Partimos del siguiente código:

```
1 public class Ejemplos {  
2  
3     private boolean sonPares (int[] arrayEjemplo) {  
4         boolean par = true;  
5         for (int i = 0; i < arrayEjemplo.length; i++)  
6             if(condicion)  
7                 par=false;  
8         return par;  
9     }  
10  
11    public static void main(String[] args) {  
12        int [] arrayEjemplo = {2,5,1,3,4};  
13        Ejemplos programa = new Ejemplos();  
14        System.out.println(programa.sonPares(arrayEjemplo)?  
15                                "Todos pares":"Hay impares");  
16    }  
17 }
```

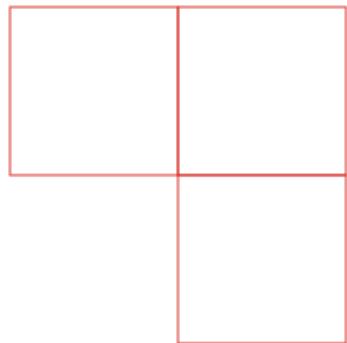
- ¿Qué condición hay que poner en la línea 6 para que funcione correctamente
  - a) arrayEjemplo%2==0
  - b) arrayEjemplo/2==0
  - c) arrayEjemplo[i]%2!=0
  - d) arrayEjemplo[i]%2==0



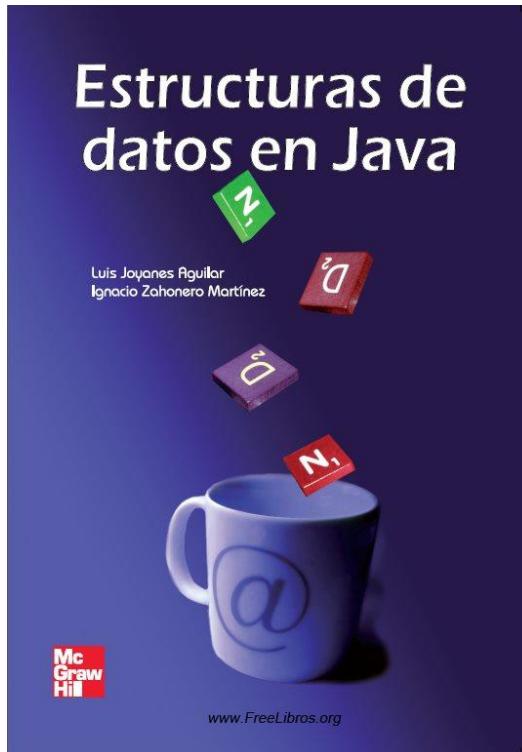
# Ejercicio 2

```
// Complete the method in this class. It returns a string with the elements of the array
// in reverse order separated by a space. If the array looks like this [1, 2 ,3 ,4], your
// output will look like this:
// 4 3 2 1
```

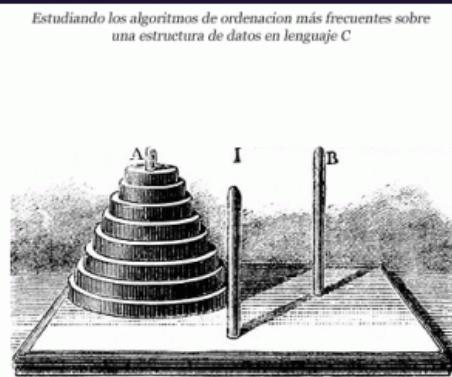
```
public class ArrayUtil
{
    /**
     * Returns a string with the elements of the array in reverse order separated by a space
     * @param values the array to reverse
     * @return the string containing the elements in reverse order
     */
    public String inReverse(int[] values)
    {
        String message = "";
        for (...) //finish the loop header
        {
            // TODO: concatenate the elements to message in reverse order
        }
        return message;
    }
}
```



# APUNTE...



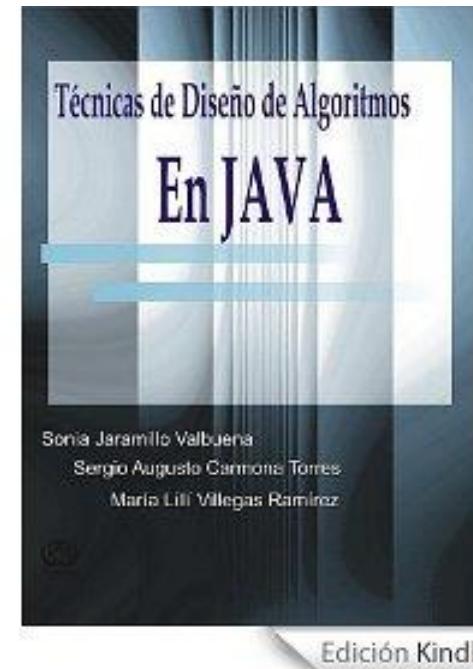
Pulsa aquí->



## Algoritmos de Ordenación

OPENLIBRA

Sebastián Gurin



Edición Kindle



**WIKIPEDIA**  
*La enciclopedia libre*



**Universidad  
Europea**

# GRACIAS

Pedro J. Camacho

[Universidadeuropea.com](http://Universidadeuropea.com)

Ve más allá