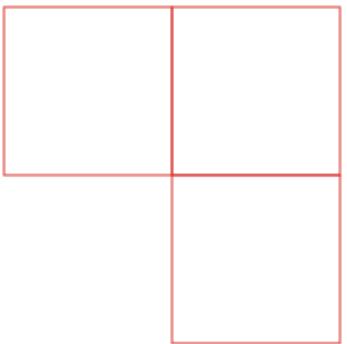


3

## Foreach y envoltorio 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

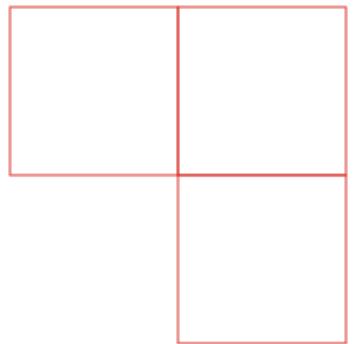


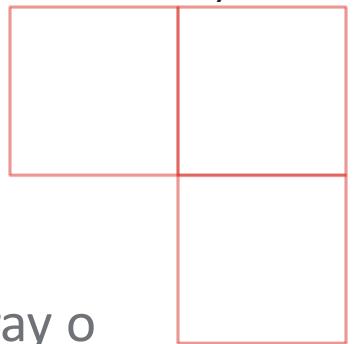


# INDICE

## ■ Foreach y envoltorio Arrays

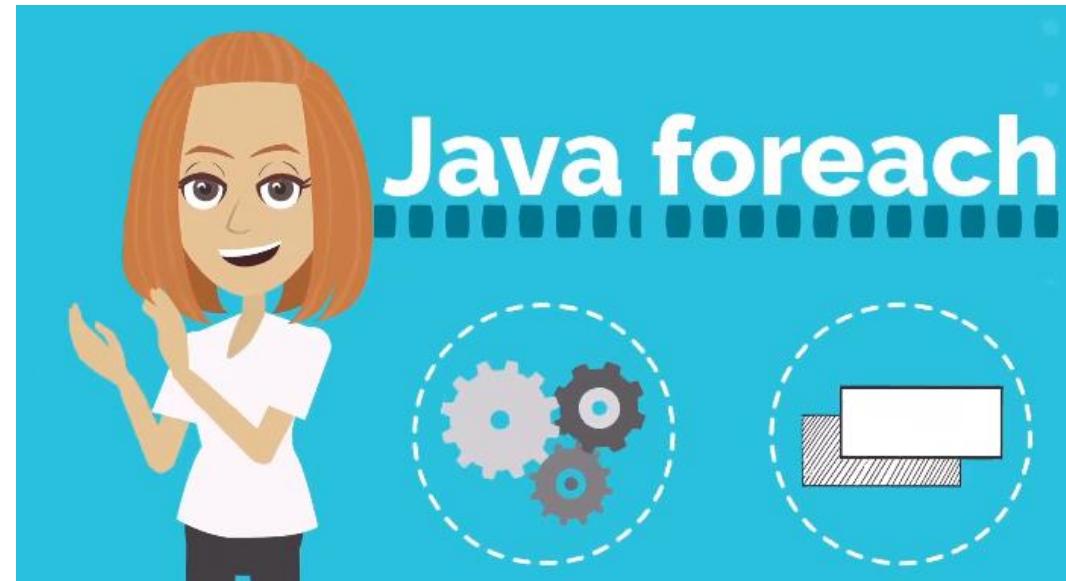
1. Introducción
2. Foreach
3. Envoltorio Arrays
4. Ejercicio

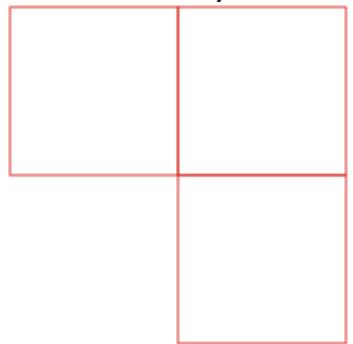




# Introducción

- Hay una versión alternativa del for que es especialmente útil cuando queremos recorrer un array o una colección de datos, cuando no nos importa la posición de los elementos. Este es el foreach o for extendido que interesa conocer.
- Además, Java nos da un envoltorio lleno de herramientas para manejar los arrays, es el envoltorio Arrays.



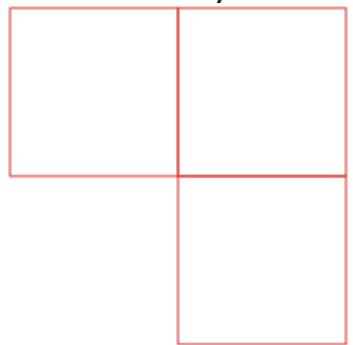


# Foreach

- También se conoce como for extendido.
- En otros lenguajes se existe un bucle cuya sintaxis empieza con la palabra clave foreach, pero en Java se utiliza la instrucción for con una sintaxis alternativa.
- Sirve para recorrer los elementos de una colección de datos sin utilizar índices auxiliares.
- Su sintaxis es la siguiente:

```
for (tipoElemento variableTemporal : Colección) {  
    bloqueInstrucciones  
}
```

- Se leería de la siguiente manera: Para cada elemento de tipo *tipoElemento* que se encuentre dentro de la colección *Colección* ejecuta el bloque *bloqueInstrucciones*.



# Foreach

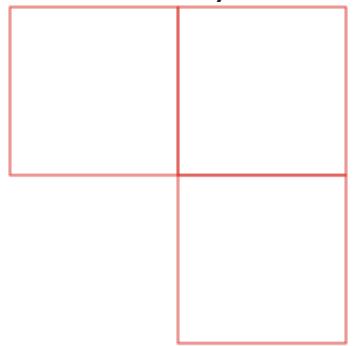
- Por ejemplo, para saber si el siguiente array está ordenado:

Array

2	5	1	3	4
[0]	[1]	[2]	[3]	[4]

- Se podría utilizar el siguiente código:

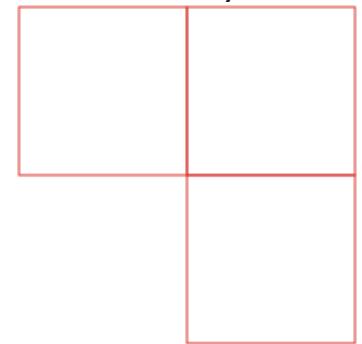
```
1 public class ForEachJava {
2
3     public static void main(String[] args) {
4
5         int [] arrayEjemplo = {1,2,3,7,5};
6         boolean estaOrdenado = true;
7
8         int anterior = Integer.MIN_VALUE;
9         for (int i : arrayEjemplo) {
10             if (anterior>i)
11                 estaOrdenado=false;
12             anterior = i;
13         }
14
15         if (estaOrdenado)
16             System.out.println("Esta ordenado");
17         else
18             System.out.println("NO esta ordenado");
19
20 }
```



# Ejercicio

```
// Complete this method to return the number of zeros in the int
array
// parameter "values"

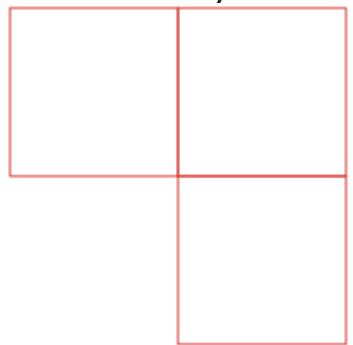
public class Util
{
    public int getCount(int[] values)
    {
        int count = 0;
        for (int current : values)
        {
            // TODO: count the number of zeros in values
        }
        return count;
    }
}
```



# Envoltorio Arrays

- El envoltorio Arrays tiene bastante utilidades que te conviene conocer:
  - `toString`: para poder imprimir directamente un vector.
  - `sort`: para ordenar
  - `binarySearch`: para buscar
- Veamos el siguiente código para comprenderlas mejor:

```
1 import java.util.*;
2 public class ConArrays{
3     public static void main(String a[]) throws Exception {
4         // Declaro un vector inicial
5         int [] miVector={11,80,66,8,9};
6
7         // Imprimir un vector
8         System.out.println(Arrays.toString(miVector));
9
10        // Ordenacion del vector
11        Arrays.sort(miVector);
12
13        // Busco un elemento en concreto en un vector ordenado
14        System.out.println(Arrays.binarySearch(miVector, 11));
15
16        // Imprimir un vector
17        System.out.println(Arrays.toString(miVector));
18    }
19 }
```

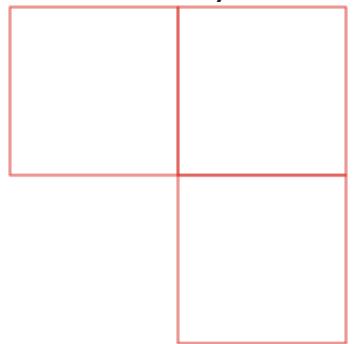


# Envoltorio Arrays

- ¿Qué tipo de ordenación hace el método sort?
  - a) Ascendente
  - b) Descendente
  - c) Se pregunta antes de ejecutarse
  - d) Alfabética
- Es necesario que el vector esté ordenado antes de utilizar el método de búsqueda, pero ¿qué devuelve binarySearch si no encuentra el elemento?
  - a) Error
  - b) 0
  - c) Not Found
  - d) -1
- ¿En qué formato se imprime el array en la línea 17?

Un elemento por línea

- a) Elementos entre llaves
- b) Elementos entre corchetes
- c) Elementos entre paréntesis



# Ejercicio

- El método sort del envoltorio Collections ordena los elementos de forma ascendente, pero ¿se podría cambiar la ordenación para que fuera de forma descendente?
- La respuesta es afirmativa, si bien, el array tiene que ser de elementos de tipo Integer en lugar de int.
- Veamos el siguiente código:

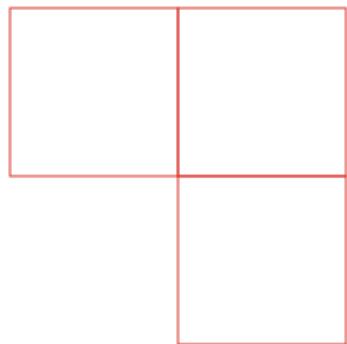
```
1 import java.util.*;
2 public class ConArrays{
3     public static void main(String a[]) throws Exception {
4         Integer [] miVector={11,80,66,8,9};
5         Arrays.sort(miVector, Collections.metodo);
6         System.out.println(Arrays.toString(miVector));
7     }
8 }
```

- ¿Qué método de “Collections” hay que poner en la línea 5 para que se imprima en orden descendente?
  - Collections.reverse
  - Collections.reverseOrder
  - Collections.falling
  - Collections.decreasing

**CURSO INICIACIÓN JAVA**  
**Tema 3 – Arrays y Strings**  
**Parte 3.4 – Arrays – foreach**  
**y envoltorio Arrays**



# APUNTE...



**BricoGeek®**

[Tienda](#)

## Cómo construir un Macro Pad casero con RP2040

Publicado por Oscar Gonzalez en Raspberry Pi el 15/10/2022

madro pad teclas cherry rp2040 raspberry pi



Un Madro Pad sirve para automatizar tareas mundanas y sobre es muy práctico cuando utilizas programas de CAD o edición de imágenes. Por supuesto existen soluciones comerciales de todo tipo que no suelen ser baratas. Pero si estás leyendo esto aquí es que estás interesado en montar un MadroPad casero uno tú mismo, que tiene mucha más gracia!



**Universidad  
Europea**

# GRACIAS

Pedro J. Camacho

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

Ve más allá