

viewnext.adrformacion.com © ADR Infor SL
VICTOR TENA PALOMARES

Arrays

© ADR Infor SL

viewnext.adrformacion.com © ADR Infor SL
VICTOR TENA PALOMARES

viewnext.adrformacion.com © ADR Infor SL
VICTOR TENA PALOMARES

Indice

Competencias y Resultados de Aprendizaje desarrollados en esta unidad	3
Arrays.	4
Objetivos.	4
Concepto de array.	4
Ejemplos.	7
Ejemplo 1. Operaciones típicas con arrays.	7
Ejemplo 2. Error de tamaño.	9
Ejemplo 3. Inicializar elementos.	10
Ejercicios	11
Ejercicio 1. Ordenar un array.	11
Datos a mostrar por consola	11
Ejercicio 2. Calcular el factorial.	11
Pistas.	12
Datos a mostrar por consola.	12
Ejercicio 3. Rellenar el código que falta.	12
Lo necesario para comenzar	12
Datos a mostrar por consola.	13
Recursos	15
Enlaces de Interés	15

Competencias y Resultados de Aprendizaje desarrollados en esta unidad

Competencia:

Conocer el concepto de Array en un programa Java y aprender su manejo

Resultados de Aprendizaje:

- Conocer qué son los arrays.
- Entender cómo se define y utiliza un array.
- Utilizar los arrays en sus diferentes variantes.

Arrays.

Objetivos.

- Conocer qué son los arrays.
- Entender cómo se define y utiliza un array.
- Conocer los métodos más comunes sobre los arrays.
- Utilizar los arrays en sus diferentes variantes.

Concepto de array.

Vocabulario: Array

Un array es una estructura de almacenamiento que contiene elementos pertenecientes a un mismo tipo de variable primitiva o referenciada (asociada a objetos), pero nunca mezcla de las dos.

Un array, independientemente del tipo de dato que almacene, **es un objeto de la clase java.lang.Object.**

Puede haber arrays de números enteros almacenados en variables int, arrays de números decimales almacenados en variables double, arrays de String, arrays de objetos Integer, etc., pero no arrays de String e Integer o de int y double o de String y char.

Características de un array.

Es una estructura estática

Su número de elementos, tamaño, longitud o dimensión debe ser declarado al construirlo y no puede modificarse durante el transcurso de un programa.

Está indexado

Cada uno de sus elementos tiene asociado un índice: 0 el primero, 1 el segundo, etc.

Declaración.

Sirve para indicar que se va a emplear un array en el código de un programa.

Vamos a ver con dos ejemplos como podemos declarar un array:

Declaración de un array de nombre enteros que almacena números enteros de tipo int

```
int enteros[];
```

ó

```
int[] enteros;
```

Declaración de un array de nombre textos que almacena datos de tipo String

```
String texto[];
```

ó

```
String[] texto;
```

Creación de la instancia.

Para poder trabajar con una variable necesitamos crear una instancia de la misma. Para ello usamos la palabra reservada **new**.

La creación de una instancia de un objeto de tipo array permite definir el número de elementos que contendrá el array.

Instancia con tamaño

```
enteros = new int[5];  
textos = new String[3];
```

Declaración + Creación.

A la hora de crear una variable podemos realizar la declaración y la creación en una sola línea, esta es la forma más normal de uso en Java.

Declaración + creación

```
int enteros[] = new int[5];  
String[] textos = new String[3];
```

Inicialización.

De momento solamente hemos creado una instancia de un tipo concreto en Java pero todavía no le hemos asignado ningún valor.

El proceso de **asignar un valor inicial a una variable** o instancia en Java se llama **inicialización**.

En este caso, inicializar un array, significa dar un valor inicial a cada posición del array. Vamos a ver un ejemplo de como debe hacerse:

```
int enteros[] = new int[5];
enteros[0]=25;
enteros[1]=30;
enteros[2]=38;
enteros[3]=48;
enteros[4]=15;
```

```
String textos[] = new String[3];
textos[0] = "Hola ";
textos[1] = "que ";
textos[2] = "tal.";
```

Declaración + Creación + Inicialización.

El lenguaje Java nos permite realizar los tres pasos implicados en la creación de una variable en una sola línea de código:

Declaración + Creación + Inicialización

```
int enteros[] = {25,30,18,38,48,15};

ó

int enteros[] = new int[] {25,30,18,38,48,15};
```

Hay que tener cuidado a la hora de inicializar arrays y tener en cuenta que la siguiente sentencia genera un error de compilación:

```
int enteros[] = new int[5];
enteros = {25,30,18,38,48,15};
```

Tamaño.

Ya hemos visto que los arrays tienen un tamaño que podemos definir con un número entero. Este tamaño indica el número de elementos que contendrá el array.

Dado que los arrays son objetos indexados y que al primer elemento le corresponde el índice 0, podemos establecer que el último elemento del array tendrá como índice **tamañoArray - 1**.

Para obtener el tamaño de un array podemos utilizar la propiedad `length` de la siguiente forma:

```
int enteros[] = new int[] {25,30,18,38,48,15};  
int tamaño = enteros.length;
```

Donde `tamaño` tendrá como valor el entero 6



Creamos nuestro primer array

Ejemplos.

Todos los códigos de este tema se guardarán en `c:\cursojava\unidad5` o en el **workspace** correspondiente si se emplea Eclipse.

Ejemplo 1. Operaciones típicas con arrays.

En este ejemplo se mostrará un código que realiza operaciones básicas con arrays.

Se puede observar que para recorrer un array se utiliza un bucle `for`. En este caso no utilizamos las llaves del bloque `for` pues su código siempre tiene una sola línea.

```

public class EjemploArray1 {
    public static void main(String args[]) {
        //Declaración, construcción e inicialización de un array de int
        int enteros[]={5,8};

        //Declaración y construcción de otro array de int
        int enterosBis[]=new int[3];
        //Inicialización de enterosBis
        enterosBis[0]=1;
        enterosBis[1]=10;
        enterosBis[2]=100;

        //Declaración, construcción e inicialización de un array de Integer
        Integer integer[]={new Integer(5),new Integer(8)};
        //Declaración, construcción e inicialización de un array de String
        String nombres[]{"Jesus","Pepe"};

        //Obtener el número de elementos de cada array
        System.out.println("Numero de elementos de cada array:");
        System.out.println("De enteros: "+enteros.length);
        System.out.println("De enterosBis: "+enterosBis.length);
        System.out.println("De integer: "+integer.length);
        System.out.println("De nombres: "+nombres.length);

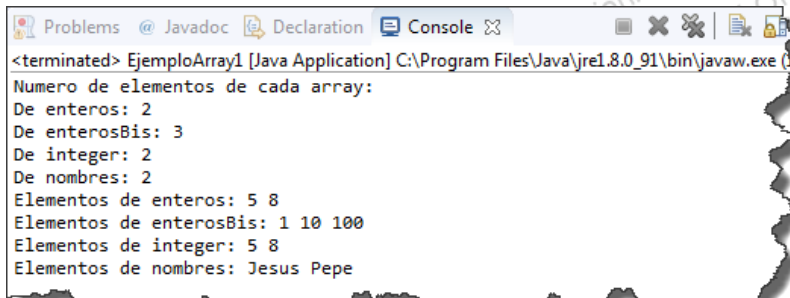
        //Mostrar por consola los elementos de cada array
        System.out.print("Elementos de enteros: ");
        for(int i=0;i<enteros.length;i++)
            System.out.print(enteros[i]+" ");

        System.out.print("\nElementos de enterosBis: ");
        for(int i=0;i<enterosBis.length;i++)
            System.out.print(enterosBis[i]+" ");

        System.out.print("\nElementos de integer: ");
        for(int i=0;i<integer.length;i++) System.out.print(integer[i]+" ");
        System.out.print("\nElementos de nombres: ");
        for(int i=0;i<nombres.length;i++)
            System.out.print(nombres[i]+" ");
    }
}

```

Si ejecutamos este código el resultado en la consola debe ser el siguiente:



```

<terminated> EjemploArray1 [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe
Numero de elementos de cada array:
De enteros: 2
De enterosBis: 3
De integer: 2
De nombres: 2
Elementos de enteros: 5 8
Elementos de enterosBis: 1 10 100
Elementos de integer: 5 8
Elementos de nombres: Jesus Pepe

```

Ejemplo 2. Error de tamaño.

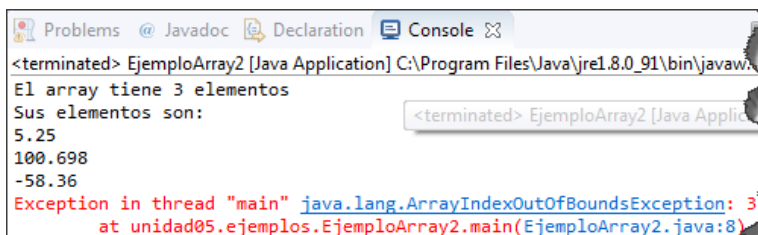
En este ejemplo se mostrará el resultado de intentar acceder a un elemento de un array mediante un índice superior a su tamaño.

```

public class EjemploArray2 {
    public static void main(String args[]) {
        double decimales[] = {5.25, 100.698, -58.36};
        System.out.println("El array tiene " + decimales.length + " elementos");
        System.out.println("Sus elementos son: ");
        for (int i = 0; i <= decimales.length; i++)
            System.out.println(decimales[i] + " ");
        System.out.println("FIN DE PROGRAMA");
    }
}

```

Al intentar ejecutar este código nos saldrá el siguiente texto en la consola:



```

<terminated> EjemploArray2 [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe
El array tiene 3 elementos
Sus elementos son:
5.25
100.698
-58.36
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3
    at unidad05.ejemplos.EjemploArray2.main(EjemploArray2.java:8)

```

En este caso el código genera una excepción de tipo `java.lang.ArrayIndexOutOfBoundsException`. Es una excepción de tipo `RuntimeException` (ver API) que no deberemos gestionar en nuestro código.

Para evitar este error se modificaría el bucle for de este modo:

```
for(int i=0;i<decimales.length;i++)
```

```
.....
```

Ejemplo 3. Inicializar elementos.

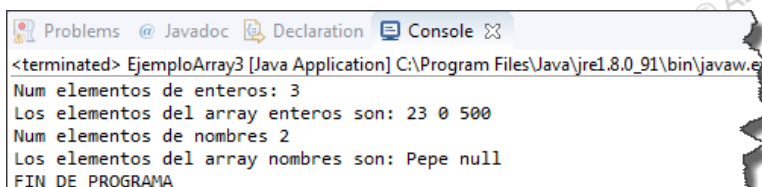
En este ejemplo vamos a ver un código que se encarga de comprobar si algún elemento de un array no se inicializa explícitamente, y cuando lo encuentra lo inicializa con el elemento nulo asociado al tipo de dato que almacene el array.

Así;

- si se tiene un array de enteros con alguno de sus elementos no inicializado va a inicializarse con 0
- si se tiene un array de números decimales con 0.0
- si se tiene un array de variables referenciadas con null

```
public class EjemploArray3 {
    public static void main(String args[]) {
        int enteros[] = new int[3];
        String nombres[] = new String[2];
        enteros[0] = 23; enteros[2] = 500;
        nombres[0] = "Pepe";
        System.out.println("Num elementos de enteros: " + enteros.length);
        System.out.print("Los elementos del array enteros son: ");
        for(int i=0; i<enteros.length; i++)
            System.out.print(enteros[i] + " ");
        System.out.println("\nNum elementos de nombres " + nombres.length);
        System.out.print("Los elementos del array nombres son: ");
        for(int i=0; i<nombres.length; i++)
            System.out.print(nombres[i] + " ");
        System.out.println("\nFIN DE PROGRAMA");
    }
}
```

En este caso, al ejecutarlo, tendremos el siguiente resultado en la consola:



```
<terminated> EjemploArray3 [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe
Num elementos de enteros: 3
Los elementos del array enteros son: 23 0 500
Num elementos de nombres 2
Los elementos del array nombres son: Pepe null
FIN DE PROGRAMA
```

Ejercicios

Todos los códigos en `c:\cursojava\unidad5` o en el **workspace** si se emplea Eclipse

Ejercicio 1. Ordenar un array.

20

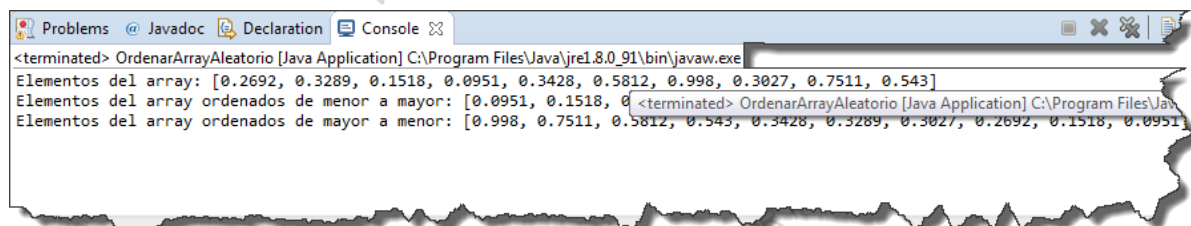
Crear una clase pública de nombre **OrdenarArrayAleatorio** que contenga sólo al método main y que haga lo siguiente:

Construir un **array** de tamaño **10** que contenga **números decimales** pertenecientes al intervalo $[0,1]$ de tipo double generados aleatoriamente y redondeados en base a la quinta cifra decimal.

A continuación:

1. Mostrar por consola sus elementos de este modo: `[ele1,ele2,...,ele10]`.
2. Mostrar por consola sus elementos ordenados de **menor a mayor** del mismo modo que antes, usando un método estático adecuado de la clase `java.util.Arrays`.
3. Igual que el anterior pero de **mayor a menor**.

Datos a mostrar por consola



```
<terminated> OrdenarArrayAleatorio [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe
Elementos del array: [0.2692, 0.3289, 0.1518, 0.0951, 0.3428, 0.5812, 0.998, 0.3027, 0.7511, 0.543]
Elementos del array ordenados de menor a mayor: [0.0951, 0.1518, 0.2692, 0.3027, 0.3289, 0.3428, 0.543, 0.5812, 0.7511, 0.998]
Elementos del array ordenados de mayor a menor: [0.998, 0.7511, 0.5812, 0.543, 0.3428, 0.3289, 0.3027, 0.2692, 0.1518, 0.0951]
```

Ejercicio 2. Calcular el factorial.

30

Crear una clase pública de nombre **Factorial** que contenga sólo al método main y que calcule el factorial de un número entero positivo mediante un array.

- Se comprobará que el **entero más grande** cuyo factorial puede calcularse utilizando variables primitivas **es 170**.
- Se pedirá al usuario que **introduzca el entero** por consola.
- Se **construirá el array** de tamaño igual al número introducido y **se inicializará** con los enteros que van desde el 1 hasta el número introducido.

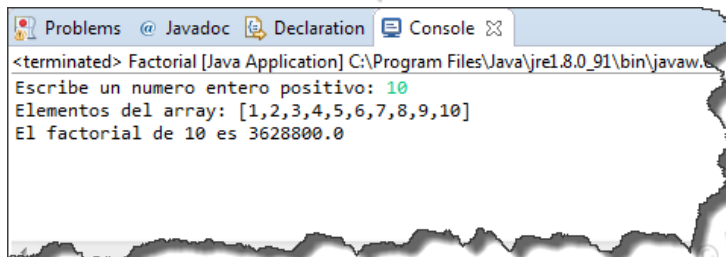
Pistas.

El factorial de **170** es **7.257415615307994E306**.

Si el usuario ha metido 5, eso implica que el array estará formado por 1,2,3,4,5.

Finalmente lo que se hará para obtener el factorial será el producto de todos los elementos que forman el array.

Datos a mostrar por consola.



```
<terminated> Factorial [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe
Escribe un numero entero positivo: 10
Elementos del array: [1,2,3,4,5,6,7,8,9,10]
El factorial de 10 es 3628800.0
```

Ejercicio 3. Rellenar el código que falta.

30

Tomando como punto de partida el esqueleto de abajo, llenar de código sus métodos en base a los comentarios.

Lo necesario para comenzar

```

public class ArrayFruticola{
    String frutas[]=new String[5];
    void inicializarArray(){
        /*
         * Inicializa el array frutas con los siguientes nombres de frutas:
         * pera, limón, melocotón, nectarina y uva.
         * Además llama al método mostrarArrayInicial pasando como parámetro un array que contiene el número de caracteres de
         * cada elemento del array frutas
         */
    }

    void mostrarArrayInicial(int frutasCaracteres[]){
        /*
         * Muestra por consola los elementos del array frutas y el producto
         * de los números de caracteres de sus elementos.
         */
    }

    String[] crearNuevoArray(){
        /*
         * Crea un nuevo array basándose en el array frutas pero sin
         * modificarlo, del siguiente modo:
         * Aquellas frutas que tengan más de 6 caracteres se cambiarán
         * por la fruta melón. El resto seguirá sin cambios.
         * El nuevo array se llamará frutillasNuevo y sus componentes serán:
         * pera, limón, melón, melón y uva.
         */
    }

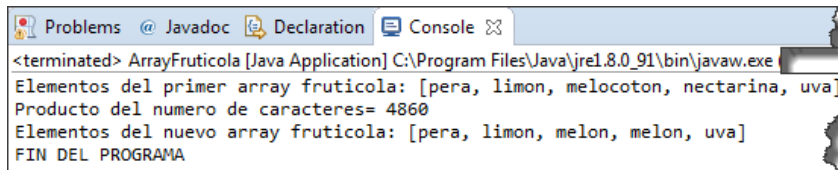
    void mostrarNuevoArray(String frutasNuevo[]){
        /*
         * Muestra por consola los elementos del array pasado por parámetro.
         */
    }

    public static void main(String args[]){
        /*
         * Se encargará del control de la ejecución.
         * En él se creará una instancia de la clase y se usará para realizar
         * las llamadas a los métodos que permitirán:
         * 1) Inicializar el array frutas.
         * 1) Crear un nuevo array y mostrar sus datos por consola.
         * 3) Mensaje de FIN DE PROGRAMA.
         */
    }
}

```

Datos a mostrar por consola.

Arrays



The screenshot shows a Java IDE console window with the following output:

```
<terminated> ArrayFruticola [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe  
Elementos del primer array fruticola: [pera, limon, melocoton, nectarina, uva]  
Producto del numero de caracteres= 4860  
Elementos del nuevo array fruticola: [pera, limon, melon, melon, uva]  
FIN DEL PROGRAMA
```

viewnext.adrformacion.com © ADR Infor SL
VICTOR TENA PALOMARES

viewnext.adrformacion.com © ADR Infor SL
VICTOR TENA PALOMARES

viewnext.adrformacion.com © ADR Infor SL
VICTOR TENA PALOMARES

Recursos

Enlaces de Interés



<http://docs.oracle.com/javase/7/docs/api/java/lang/ArrayIndexOutOfBoundsException.html>

API java.lang.ArrayIndexOutOfBoundsException