



DEFINICION DE ARREGLOS



Qué es un Array en Java

Una array o arreglo es una colección de variables del mismo tipo, a la que se hace referencia por un nombre común. En Java, los arrays pueden tener una o más dimensiones, aunque el array unidimensional es el más común.

Los arrays se usan para una variedad de propósitos porque ofrecen un medio conveniente de agrupar variables relacionadas. Por ejemplo, puede usar una matriz para mantener un registro de la temperatura alta diaria durante un mes, una lista de promedios de precios de acciones o una lista de tu colección de libros de programación.

La ventaja principal de un array es que organiza los datos de tal manera que puede ser manipulado fácilmente. Por ejemplo, si tiene un array que contiene los ingresos de un grupo seleccionado de hogares, es fácil calcular el ingreso promedio haciendo un ciclo a través del array. Además, los arrays organizan los datos de tal manera que se pueden ordenar fácilmente.



Aunque los arrays en Java se pueden usar como matrices en otros lenguajes de programación, tienen un atributo especial: se implementan como objetos. Este hecho es una de las razones por las que la discusión de los arrays se pospuso hasta que se introdujeron los objetos. Al implementar arrays como objetos, se obtienen varias ventajas importantes, una de las cuales es que los arrays no utilizados pueden ser recolectados.

Instanciando un array en Java

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:

nombre-array = new tipo ;

tipo especifica el tipo de datos que se asignará

tamaño especifica el número de elementos en el array

nombre-array es el 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.

Ejemplo:

```
int intArray[]; //declarando un array
```

```
intArray = new int; // asignando memoria al array
```

o

```
int[] intArray = new int; // combinando ambas declaraciones en una
```

Nota:

- Los elementos en la matriz asignada por se inicializarán automáticamente a cero (para tipos numéricos), false (para booleano) o null (para tipos de referencia).
- Obtener un array es un proceso de dos pasos. Primero, debe declarar una variable del tipo de array deseado. En segundo lugar, debe asignar la memoria que mantendrá el array, usar y asignarla a la variable del array. Por lo tanto, en Java, todos los arrays se asignan dinámicamente.



Un array Java es una estructura de datos que nos permite almacenar una ristra de datos de un mismo tipo. El tamaño de los arrays se declara en un primer momento y no puede cambiar en tiempo de ejecución como puede producirse en otros lenguajes. La declaración de un array en Java y su inicialización se realiza de la siguiente manera:

```
tipo_dato nombre_array[];  
  
nombre_array = new tipo_dato[tamano];
```

Por ejemplo, podríamos declarar un array de caracteres e inicializarlo de la siguiente manera:

```
char arrayCaracteres[];  
  
arrayCaracteres = new char[10];
```

Los arrays Java se numeran desde el elemento cero, que sería el primer elemento, hasta el tamaño-1 que sería el último elemento. Es decir, si tenemos un array de diez elementos, el primer elemento sería el cero y el último elemento sería el nueve. Para acceder a un elemento específico utilizaremos los corchetes de la siguiente forma. Entendemos por acceso, tanto el intentar leer el elemento, como asignarle un valor.

```
arrayCaracteres[numero_elemento];
```

Por ejemplo, para acceder al tercer elemento lo haríamos de la siguiente forma:

```
// Lectura de su valor.  
  
char x = arrayCaracteres[2];
```

// Asignación de un valor. Como se puede comprobar se pone el número dos, que coincide con el tercer elemento. Ya que como dijimos anteriormente el primer elemento es el cero.

```
arrayCaracteres[2] = 'b';
```

El objeto array, aunque podríamos decir que no existe como tal, posee una variable, la cual podremos utilizar para facilitar su manejo.