¿Qué es un Array en Java?

Un array es una estructura de datos que nos permite almacenar una gran cantidad 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[tamaño];
```

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

```
char arrayCaracteres[];


arrayCaracteres = new char[10];
```

Los arrays 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 especifico 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:


         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.

Tamaño del array: .length

Esta variable nos devuelve el número de elementos que posee el array. Hay que tener en cuenta que es una variable de solo lectura, es por ello que no podremos realizar una asignación a dicha variable.

Por ejemplo esto nos serviría a la hora de mostrar el contenido de los elementos de un array:

```
char array[];

    array = new char[10];

    for (int x=0;x
```

Matrices en Java

Podremos declarar arrays de varios subíndices, pudiendo tener arrays de dos niveles, que serían similares a las matrices, arrays de tres niveles, que serían como cubos y así sucesivamente, si bien a partir del tercer nivel se pierde la perspectiva geométrica. Para declarar e inicializar un array de varios subíndices lo haremos de la siguiente manera:

```
tipo_dato nombre_array[][];


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

De esta forma podemos declarar una matriz de 2x2 de la siguiente forma:

```
int matriz[][];


    matriz = new int[2][2];
```

El acceso se realiza de la misma forma que antes:

```
int x = matriz[1][1]; // Para leer el contenido de un elemento


    matriz[1][1] = x; // Para asignar un valor.
```

Hay que tener en cuenta que para mostrar su contenido tendremos que utilizar dos bucles. Para saber el número de columnas lo haremos igual que antes mediante la variable length, pero para saber el numero de filas que contiene cada columna lo tendremos que realizar de la siguiente manera:

matriz[numero_elemento].lenght

Nuestra lectura de los elementos de una matriz quedaría de la siguiente forma:

```
int matriz[][];
        matriz = new int[4][4];


        for (int x=0; x < matrix.length; x++) {
                for (int y=0; y < matriz[x].length; y++) {


          System.out.println (matriz[x][y]);
                }
          }
```

Inicializacion de Arrays

Existe una forma de inicializar un array con el contenido, amoldándose su tamaño al número de elementos a los que le inicialicemos. Para inicializar un array utilizaremos las llaves de la siguiente forma:

```
    tipo_dato array[] = {elemento1,elemento2,...,elementoN};
```

Así, por ejemplo, podríamos inicializar un array o una matriz:

```
   // Tenemos un array de 5 elementos.
          char array[] = {'a','b','c','d','e'};



          // Tenemos un array de 4x4 elementos.
```

```
int array[][] = { {1,2,3,4}, {5,6,7,8}};
```