Arrays multidimensionales en Java

ÍNDICE

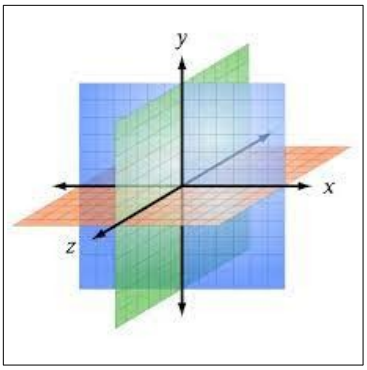
1. Introducción

Un **array multidimensional** es una estructura de datos que nos permite almacenar un conjunto de datos de un **mismo tipo** en un sistema de múltiples dimensiones (más de dos). Como ya sabemos, necesitaremos un valor de índice por cada dimensión del array. Por ejemplo, si el array tiene 3 dimensiones necesitaremos 3 índices para acceder a cada posición del array.

Durante el presente tema nos vamos a centrar en el estudio de los mismos.

2. Arrays multidimensionales

El uso de dimensiones en los arrays puede llegar hasta las dimensiones que deseemos y tengan sentido para nuestra lógica de negocio. Por ejemplo, si tenemos que situar un punto en el espacio, podemos hacer uso de sus coordenadas (x, y, z). Dicha situación, la podemos modelar con un array de 3 dimensiones (una por cada coordenada).



2.1 Declaración e inicialización

Como ya podemos intuir, necesitaremos el uso de 3 subíndices de la siguiente forma:

```
nombreArray = new TipoDato[tamaño1][tamaño2][tamaño3]
```

Imaginad una situación en la que debemos almacenar cuántas personas viven en un barrio residencial compuesto por 4 portales, cada uno de ellos con 7 pisos/plantas y 2 viviendas por piso. Podemos modelarlo de la siguiente forma:

```
byte[][][] residentes = new byte[4][7][2];
```

2.2 Asignación de valores

¿Cómo podríamos indicar que en el portal 3 (posi índice ción 2), piso 5 (índice 4) y vivienda 1 (índice 0) viven 5 personas? De la siguiente forma:

```
residentes[2][4][0] = 5;
```

2.3 Lectura de valores

¿Cómo podría saber cuántas personas viven en el portal 1 (índice 0), piso 4 (índice 3) y vivienda 2 (índice 1)? De la siguiente forma:

```
byte numResidentes = residentes[0][3][1];
```

2.4 Resumen

De esta manera, un código completo, resumiendo las operaciones vistas, podría ser el siguiente:

```
byte[][][] residentes = new byte[4][7][2];
residentes[2][4][0] = 5;
byte numResidentes = residentes[0][3][1];
```

3. Recorrer arrays multidimensionales

Para recorrer arrays multidimensionales debemos anidar un `bucle n` por cada dimensión. Por ejemplo, si el array tiene 3 dimensiones, deberemos utilizar 3 bucles `for` (o `foreach`) uno dentro de otro.

```
for(int i=0; i < miArray.length; i++) {
	for(int j=0; j < miArray[i].length; j++) {
		for(int k=0; k < miArray[i][j].length; k++) {
			System.out.println(miArray[i][j][k]);
		}
	}
}
```