

Arrays Multidimensionales y HashSet

1. Arrays Multidimensionales

Un array multidimensional es una estructura de datos que permite almacenar elementos de manera organizada en más de una dimensión. En otras palabras, es un arreglo de arreglos.

Conceptos Clave:

- Dimensiones: La dimensión de un array indica cuántos índices necesitamos para acceder a un elemento.
 - 1D: Array simple [a, b, c]
 - 2D: Matriz [[a1, a2], [b1, b2]]
 - 3D o más: Array de matrices [[[a]], [[b]]]
- Acceso a elementos: Se utiliza una serie de índices para acceder a cada elemento.

Ejemplo en Java:

```
int[][] matriz = { {1, 2}, {3, 4} };  
int valor = matriz[0][1]; // valor = 2
```

- Uso común: Representación de tablas, imágenes, tableros de juego, etc.

Ventajas:

- Organiza datos en estructuras lógicas.
- Permite iteraciones sencillas con bucles anidados.

Limitaciones:

- El tamaño de cada dimensión generalmente debe definirse al crear el array.
- Pueden consumir mucha memoria si son de gran tamaño.

Referencias Arrays Multidimensionales:

- <https://www.geeksforgeeks.org/multidimensional-arrays-in-java/>
- <https://www.javatpoint.com/multidimensional-array>

2. HashSet

Un HashSet es una colección que almacena elementos únicos sin un orden específico, basado en una estructura de hashing. Forma parte del paquete java.util.

Conceptos Clave:

Arrays Multidimensionales y HashSet

- Elementos únicos: No permite duplicados.

Ejemplo en Java:

```
HashSet<String> conjunto = new HashSet<>();  
conjunto.add("Manzana");  
conjunto.add("Banana");  
conjunto.add("Manzana"); // No se agregará de nuevo
```

- Acceso rápido: Gracias al hashing, las operaciones de agregar, eliminar o buscar elementos son muy eficientes.
- Sin orden: No mantiene un orden específico de los elementos.
- Métodos importantes:
 - add(E e) -> agrega un elemento.
 - remove(Object o) -> elimina un elemento.
 - contains(Object o) -> verifica si existe un elemento.
 - size() -> devuelve la cantidad de elementos.

Ventajas:

- Previene duplicados automáticamente.
- Operaciones rápidas de inserción y búsqueda.

Limitaciones:

- No mantiene el orden de los elementos.
- Puede consumir más memoria que un array simple.

Referencias HashSet:

- <https://www.geeksforgeeks.org/hashset-in-java/>
- <https://www.javatpoint.com/hashset-in-java>