¿Cómo se declara un arreglo en Java?

Los arreglos en java son unas estructuras de datos que permiten almacenar múltiples valores del mismo tipo en una sola variable. Estas son de un tamaño fijo y se pueden acceder a sus elementos mediante índices que comienzan en 0.

Ejemplos:

```java
// Forma 1: declarar  y asignar tamaño
int[] numeros = new int[5];
numeros[0] = 10;
numeros[1] = 20;

 // Forma 2: declarar e inicializar directamente
int[] valores = {5, 3, 9, 1, 7};
```

Métodos principales de java.util.Arrays

La clase Arrays del paquete java.util proporciona métodos estáticos que facilitan el manejo de arreglos. Estos métodos permiten ordenar, buscar, copiar, llenar y comparar arreglos de forma sencilla y eficiente, evitando que el programador tenga que implementar estas operaciones manualmente.

Arrays.sort(): Ordena los elementos del arreglo en orden ascendente.

Arrays.binarySearch(): Busca un elemento dentro de un arreglo previamente ordenado.

Arrays.copyOf(): Crea una copia del arreglo con el tamaño indicado.

Arrays.fill(): Rellena todas las posiciones del arreglo con un mismo valor.

Arrays.equals(): Compara dos arreglos y verifica si son iguales.

Arrays.toString(): Convierte el arreglo en una cadena de texto legible.

```java
// Ordenar arreglo
   Arrays.sort(valores);

   // Buscar un valor (el arreglo debe estar ordenado)
   int posicion = Arrays.binarySearch(valores, 9);
   System.out.println("Posición del 9: " + posicion);

   // Copiar arreglo
   int[] copia = Arrays.copyOf(valores, valores.length);

   // Llenar arreglo
   Arrays.fill(numeros, 100);

   // Comparar arreglos
   boolean iguales = Arrays.equals(valores, copia);
   System.out.println("¿Son iguales? " + iguales);

   // Mostrar arreglo como texto
   System.out.println("Valores: " + Arrays.toString(valores));
```

¿Cómo se recorren los arreglos en Java?

Recorrer un arreglo significa acceder a cada uno de sus elementos. En Java existen diferentes formas de recorrer arreglos, cada una adecuada para distintas necesidades. El for tradicional permite trabajar con índices, el for-each simplifica la lectura del código y los Streams ofrecen una forma moderna y funcional de procesar los datos.

For tradicional: Permite recorrer el arreglo usando índices.

For-each: Recorre directamente los valores del arreglo.

Streams: Forma moderna de recorrer arreglos usando programación funcional.

```java
   // 4.3 RECORRER ARREGLOS
// 1. For tradicional
   for (int i = 0; i < valores.length; i++) {
       System.out.println("For tradicional: " + valores[i]);
   }

   // 2. For-each
   for (int v : valores) {
       System.out.println("For-each: " + v);
   }

   // 3. Streams (Java 8+)
   Arrays.stream(valores).forEach(v -> {
       System.out.println("Stream: " + v);
   });
```

Diferencias entre arreglos y ArrayList en Java

Los arreglos tienen tamaño fijo y permiten usar tipos primitivos, mientras que ArrayList es dinámico y utiliza clases envolventes. ArrayList ofrece métodos que facilitan la manipulación de datos..

```java
// ARRAY: tamaño fijo, tipo primitivo
int[] edadesArreglo = new int[3];
edadesArreglo[0] = 18;
edadesArreglo[1] = 20;
edadesArreglo[2] = 25;

System.out.println("Tamaño del arreglo: " + edadesArreglo.length);

// ARRAYLIST: tamaño dinámico, clase envolvente
ArrayList<Integer> edadesLista = new ArrayList<>();
edadesLista.add(18);
edadesLista.add(20);
edadesLista.add(25);
edadesLista.add(30); // crece dinámicamente

System.out.println("ArrayList: " + edadesLista);
System.out.println("Tamaño del ArrayList: " + edadesLista.size());

// Eliminar elemento
edadesLista.remove(1);

// Acceso a elementos
System.out.println("Primer elemento del arreglo: " + edadesArreglo[0]);
System.out.println("Primer elemento del ArrayList: " + edadesLista.get(0));
```