

ARREGLOS JAVA

Conceptos Fundamentales de Arreglos en Java

Los arreglos son objetos contenedores que almacenan múltiples variables del mismo tipo. En Java, los arreglos tienen características esenciales:

- **Tamaño Fijo:** Una vez que se crea un arreglo, su tamaño es inmutable. No se pueden agregar ni eliminar elementos dinámicamente.
- **Mismo Tipo de Elementos:** Todos los elementos dentro de un arreglo deben ser del mismo tipo de dato (por ejemplo, todos int, todos String, etc.).
- **Acceso por Índice:** Los elementos se acceden y modifican utilizando un índice (posición), que comienza en 0 y va hasta la longitud del arreglo menos uno (`length - 1`).

Código de Ejemplo

El siguiente código muestra diferentes maneras de trabajar con arreglos en Java.

```
import java.util.Arrays;

import java.util.Iterator;

import java.util.Scanner;

public class App {

    public static void main(String[] args) {
        //declaracion

        int[] edades;

        //inicializacion del arreglo con 5 elementos

        edades = new int[5];

        //este arreglo se inicia con el valor por defecto

        //para cada elemento int, que es 0
    }
}
```

```
//PARA VER TODO EL ARREGLO PODEMOS USAR LA CLASE Arrays

//con su método toString

System.out.println(Arrays.toString(edades));

//modificando elementos, lo hacemos por su índice

edades[0] = 5;

edades[3] = 77;

System.out.println(Arrays.toString(edades));

//longitud del arreglo

System.out.println(edades.length);

//recorrido con for each

for (int edad : edades) {

    System.out.println("EDAD " + edad);

}

//RECORRIDO CON FOR

for (int i = 0; i < edades.length; i++) {

    System.out.println("EDAD " + edades[i]);

}

//recorrido con while

int contador = 0;

while(contador < edades.length) {

    System.out.println("EADADES ---" + edades[contador]);
```

```

        contador++;

    }

    //otra manera de crear arreglos

    //declaracion e inicializacion

    float[] estatura = new float[5];

    estatura[0] = 45.6f;

    System.out.println(Arrays.toString(estatura));

    String[] nombres = {"ADSO", "SENA", "GALAN"};

    //actualizar un valor de un arreglo, ingresado por el
    usuario

    Scanner entrada = new Scanner(System.in);

    System.out.println("INGRESE NOMBRE");

    String nombreNuevo = entrada.nextLine();

    nombres[2] = nombreNuevo;

    System.out.println(Arrays.toString(nombres));

    }

}

```

1. Declaración e Inicialización

Concepto	Código de Ejemplo	Descripción
Declaración	<code>int[] edades;</code>	Declara una variable de referencia edades capaz de

Concepto	Código de Ejemplo	Descripción
		apuntar a un arreglo de enteros.
Inicialización por Tamaño	<code>edades = new int[5];</code>	Instancia el arreglo, asignándole un espacio de memoria para 5 elementos enteros. Los valores se inicializan a los valores por defecto (0 para int).
Inicialización Directa	<code>String[] nombres = {"ADSO", "SENA", "GALAN"};</code>	Declara e inicializa el arreglo con los valores especificados. El tamaño del arreglo se fija automáticamente a 3.

2. Clase Arrays y Representación

Concepto	Código de Ejemplo	Descripción
Arrays.toString()	<code>System.out.println(Arrays.toString(edades));</code>	Método estático de la clase <code>java.util.Arrays</code> que permite obtener una representación <code>String</code> del contenido del arreglo, útil para la depuración.
Importación	<code>import java.util.Arrays;</code>	Necesaria para usar la clase <code>Arrays</code> .
Importación Scanner	<code>import java.util.Scanner;</code>	Necesaria para la interacción con el usuario (entrada de datos).

3. Acceso y Modificación de Elementos

Los elementos se modifican y acceden por su índice, que siempre empieza en 0.

Concepto	Código de Ejemplo	Descripción
Modificación	<code>edades[0] = 5;</code>	Asigna el valor 5 al primer elemento (índice 0) del arreglo edades.
Acceso a longitud	<code>edades.length</code>	La propiedad <code>length</code> devuelve el número de elementos en el arreglo (en este caso, 5). Es una propiedad final, no un método.
Actualización por Usuario	<code>nombres[2] = nombreNuevo;</code>	El valor del elemento en el índice 2 (tercer elemento) es reemplazado por la cadena de texto ingresada por el usuario a través de Scanner.

4. Recorridos del Arreglo

Hay varias formas de iterar sobre los elementos de un arreglo:

Tipo de Recorrido	Código de Ejemplo	Descripción
for tradicional	<pre>for (int i = 0; i < edades.length; i++) { ... }</pre>	Utiliza un contador (i) para iterar desde el índice 0 hasta menor que la longitud del arreglo. Permite acceder al índice y al valor.
for-each (Bucle Mejorado)	<pre>for (int edad : edades) { ... }</pre>	Recorre todos los elementos del arreglo edades, asignando el valor de cada elemento secuencialmente a la variable edad. Es más conciso, pero no permite modificar el arreglo ni acceder al índice directamente.

Tipo de Recorrido	Código de Ejemplo	Descripción
while	<pre>while(contador < edades.length) { ... }</pre>	Utiliza un contador y una condición explícita para iterar. Es más verboso que el for para este propósito, pero demuestra el control manual de la iteración.