

Arreglos en Java

1. Introducción a los Arreglos

Los **arreglos** (arrays) son estructuras de datos fundamentales en Java que permiten almacenar múltiples elementos del mismo tipo bajo un nombre común. Cada elemento se accede mediante un índice numérico que comienza en 0.

Características principales:

- Colección de objetos del mismo tipo
- Acceso mediante índices
- Tamaño fijo una vez creados
- Los índices van de 0 a (longitud - 1)

2. Declaración y Creación de Arreglos

Sintaxis de Declaración

```
```java
// Forma 1: Recomendada
tipo_dato[] identificador;

// Forma 2: Alternativa
tipo_dato identificador[];
```
```

Ejemplos de Declaración

```
```java
int[] numeros; // Arreglo de enteros
String[] nombres; // Arreglo de cadenas
double[] promedios; // Arreglo de decimales
```
```

Creación de Arreglos

```
```java
// Creación separada
int[] numeros;
numeros = new int[5]; // Crea arreglo de 5 enteros

// Declaración y creación juntas
int[] numeros = new int[5];
```
```

3. Inicialización de Arreglos

Inicialización con Valores por Defecto

Cuando se crea un arreglo, sus elementos reciben valores por defecto:

- **int, byte, short, long:** 0
- **float, double:** 0.0
- **boolean:** false
- **Referencias de objetos:** null

Inicialización Explícita

```
```java
// Inicialización directa
int[] numeros = {10, 20, 30, 40, 50};

// Arreglos de String
String[] ciudades = {"México", "Buenos Aires", "Lima", "Bogotá"};

// Con variables
String nombre = "Juan";
String[] saludos = {"Hola " + nombre, "Adiós " + nombre};
```
```

4. Acceso y Manipulación de Elementos

Acceso a Elementos

```

```java
int[] calificaciones = {85, 92, 78, 96, 88};

// Acceder al primer elemento
int primera = calificaciones[0]; // 85

// Modificar un elemento
calificaciones[2] = 80; // Cambia 78 por 80

// Obtener la longitud
int longitud = calificaciones.length; // 5
```

```

Ejemplo Práctico: Sistema de Calificaciones

```

```java
public class SistemaCalificaciones {
 public static void main(String[] args) {
 String[] estudiantes = {"Ana", "Carlos", "María", "Pedro"};
 int[] calificaciones = {85, 92, 78, 96};

 System.out.println("Reporte de Calificaciones:");
 for (int i = 0; i < estudiantes.length; i++) {
 System.out.println(estudiantes[i] + ": " + calificaciones[i]);
 }
 }
}
```

```

5. Arreglos Multidimensionales

Matrices (Arreglos Bidimensionales)

```

```java
// Declaración y creación
int[][] matriz = new int[3][4]; // 3 filas, 4 columnas

// Inicialización directa
int[][] tabla = {
 {1, 2, 3},
 {4, 5, 6},
}
```

```

```
{7, 8, 9}  
};  
...
```

Arreglos Irregulares (Jagged Arrays)

```
```java  
int[][] irregular = new int[3][];
irregular[0] = new int[2]; // Primera fila: 2 elementos
irregular[1] = new int[4]; // Segunda fila: 4 elementos
irregular[2] = new int[3]; // Tercera fila: 3 elementos
```
```

Ejemplo: Tabla de Multiplicar

```
```java  
public class TablaMultiplicar {
 public static void main(String[] args) {
 int[][] tabla = new int[10][10];

 // Llenar la tabla
 for (int i = 0; i < 10; i++) {
 for (int j = 0; j < 10; j++) {
 tabla[i][j] = (i + 1) * (j + 1);
 }
 }

 // Mostrar la tabla del 5
 System.out.println("Tabla del 5:");
 for (int j = 0; j < 10; j++) {
 System.out.println("5 × " + (j + 1) + " = " + tabla[4][j]);
 }
 }
}
```
```

6. Recorrido de Arreglos

Ciclo For Tradicional

```
```java
```

```
int[] numeros = {10, 20, 30, 40, 50};

for (int i = 0; i < numeros.length; i++) {
 System.out.println("Elemento " + i + ": " + numeros[i]);
}
...
```

## For-Each (Enhanced For Loop)

```
```java
int[] numeros = {10, 20, 30, 40, 50};

for (int numero : numeros) {
    System.out.println("Número: " + numero);
}
...
```

Ejemplo: Análisis de Temperaturas

```
```java
public class AnalisisTemperaturas {
 public static void main(String[] args) {
 double[] temperaturas = {22.5, 25.0, 23.8, 26.2, 24.1, 21.9, 23.3};
 String[] dias = {"Lunes", "Martes", "Miércoles", "Jueves",
 "Viernes", "Sábado", "Domingo"};

 double suma = 0;
 double maxTemp = temperaturas[0];
 double minTemp = temperaturas[0];

 // Calcular estadísticas
 for (double temp : temperaturas) {
 suma += temp;
 if (temp > maxTemp) maxTemp = temp;
 if (temp < minTemp) minTemp = temp;
 }

 double promedio = suma / temperaturas.length;

 System.out.println("Análisis Semanal de Temperaturas:");
 System.out.println("Temperatura promedio: " + promedio + "°C");
 System.out.println("Temperatura máxima: " + maxTemp + "°C");
 System.out.println("Temperatura mínima: " + minTemp + "°C");
 }
}
```

```
}
}
...
```

## 7. Arreglos de Objetos

### Creación de Arreglos de Objetos

```
```java  
// Clase Estudiante  
class Estudiante {  
    private String nombre;  
    private int edad;  
  
    public Estudiante(String nombre, int edad) {  
        this.nombre = nombre;  
        this.edad = edad;  
    }  
  
    public String toString() {  
        return nombre + " (" + edad + " años)";  
    }  
}  
  
// Uso del arreglo de objetos  
public class UniversidadDemo {  
    public static void main(String[] args) {  
        // Crear arreglo de estudiantes  
        Estudiante[] estudiantes = new Estudiante[3];  
  
        // Inicializar cada elemento  
        estudiantes[0] = new Estudiante("Ana García", 20);  
        estudiantes[1] = new Estudiante("Carlos López", 22);  
        estudiantes[2] = new Estudiante("María Rodríguez", 21);  
  
        // Mostrar información  
        for (Estudiante est : estudiantes) {  
            System.out.println(est);  
        }  
    }  
}  
...
```

8. Paso de Parámetros con Arreglos

Los Arreglos se Pasan por Referencia

```
```java
public class ModificacionArreglos {

 public static void modificarArreglo(int[] arr) {
 arr[0] = 100; // Modifica el arreglo original
 }

 public static void main(String[] args) {
 int[] numeros = {1, 2, 3, 4, 5};

 System.out.println("Antes: " + numeros[0]); // 1
 modificarArreglo(numeros);
 System.out.println("Después: " + numeros[0]); // 100
 }
}
```
```

Ejemplo: Calculadora de Estadísticas

```
```java
public class CalculadoraEstadisticas {

 public static double calcularPromedio(double[] valores) {
 double suma = 0;
 for (double valor : valores) {
 suma += valor;
 }
 return suma / valores.length;
 }

 public static double encontrarMaximo(double[] valores) {
 double max = valores[0];
 for (double valor : valores) {
 if (valor > max) {
 max = valor;
 }
 }
 }
}
```

```

 }
 return max;
}

public static void main(String[] args) {
 double[] ventas = {1500.50, 2300.75, 1800.25, 2100.00, 1950.80};

 System.out.println("Ventas promedio: $" + calcularPromedio(ventas));
 System.out.println("Venta máxima: $" + encontrarMaximo(ventas));
}
}
...

```

## 9. Métodos Útiles para Arreglos

### Clonación de Arreglos

```

```java
int[] original = {1, 2, 3, 4, 5};
int[] copia = original.clone(); // Copia superficial
...

```

System.arraycopy()

```

```java
int[] origen = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int[] destino = new int[5];

// Copiar 5 elementos desde la posición 2 del origen
// a la posición 0 del destino
System.arraycopy(origen, 2, destino, 0, 5);
// destino ahora contiene: {3, 4, 5, 6, 7}
...

```

## 10. El Arreglo args[] en el Método main

### Argumentos de Línea de Comandos

```

```java

```



```

public class ArgumentosDemo {
    public static void main(String[] args) {
        System.out.println("Número de argumentos: " + args.length);

        for (int i = 0; i < args.length; i++) {
            System.out.println("Argumento " + i + ": " + args[i]);
        }
    }
}
...

```

Si ejecutas: `java ArgumentosDemo Hola Mundo 123`

Salida:

...

Número de argumentos: 3

Argumento 0: Hola

Argumento 1: Mundo

Argumento 2: 123

...

11. Ejercicios Prácticos

Ejercicio 1: Inventario de Productos

```

```java
public class InventarioTienda {
 public static void main(String[] args) {
 String[] productos = {"Laptop", "Mouse", "Teclado", "Monitor"};
 int[] cantidades = {15, 50, 30, 8};
 double[] precios = {15000.00, 250.00, 800.00, 5500.00};

 System.out.println("INVENTARIO DE TIENDA");
 System.out.println("=====");

 double valorTotal = 0;
 for (int i = 0; i < productos.length; i++) {
 double valorProducto = cantidades[i] * precios[i];
 valorTotal += valorProducto;

 System.out.printf("%-10s | Cant: %3d | Precio: $%8.2f | Valor: $%10.2f%n",
 productos[i], cantidades[i], precios[i], valorProducto);
 }
 }
}

```

```

 System.out.println("=====");
 System.out.printf("VALOR TOTAL DEL INVENTARIO: $%.2f%n", valorTotal);
 }
}
...

```

## Ejercicio 2: Matriz de Calificaciones por Materia

```

```java
public class CalificacionesMateria {
    public static void main(String[] args) {
        String[] materias = {"Matemáticas", "Física", "Química", "Historia"};
        String[] estudiantes = {"Ana", "Carlos", "María"};

        // Matriz: [estudiante][materia]
        double[][] calificaciones = {
            {85.5, 78.0, 92.5, 88.0}, // Ana
            {90.0, 85.5, 87.0, 82.5}, // Carlos
            {88.5, 91.0, 89.5, 94.0}  // María
        };

        // Mostrar calificaciones
        System.out.println("REPORTE DE CALIFICACIONES");
        System.out.println("=====");

        for (int i = 0; i < estudiantes.length; i++) {
            System.out.println("\n" + estudiantes[i] + ":");
            double suma = 0;

            for (int j = 0; j < materias.length; j++) {
                System.out.printf(" %-12s: %.1f%n",
                    materias[j], calificaciones[i][j]);
                suma += calificaciones[i][j];
            }

            double promedio = suma / materias.length;
            System.out.printf(" Promedio: %.1f%n", promedio);
        }
    }
}
...

```

Resumen

Los arreglos en Java son estructuras de datos esenciales que permiten:

1. **Almacenar múltiples elementos** del mismo tipo
2. **Acceso eficiente** mediante índices
3. **Manipulación de datos** de forma organizada
4. **Creación de estructuras multidimensionales** para datos complejos
5. **Paso eficiente de parámetros** a métodos

Puntos Clave para Recordar:

- Los índices comienzan en 0
- El atributo `length` proporciona el tamaño del arreglo
- Los arreglos se pasan por referencia
- Se pueden crear arreglos de cualquier tipo de dato
- Las matrices pueden ser irregulares (diferentes tamaños por fila)

Los arreglos son fundamentales en la programación y forman la base para estructuras de datos más complejas como listas, pilas y colas.

