

Tema 5. Estructuras de datos – 2 Arrays multidimensionales

1. ARRAYS MULTIDIMENSIONALES

Un array puede tener más de una dimensión. El caso más general son los arrays de dos dimensiones (bidimensionales) también llamados **matrices** o **tablas**.

La dimensión de un array la determina el número de índices necesarios para acceder a sus elementos.

Los arrays unidimensionales solo utilizan un índice para acceder a cada elemento.

Un array bidimensional necesita dos índices para acceder a sus elementos. Gráficamente podemos representar un array bidimensional como una tabla de n filas y m columnas cuyos elementos son todos del mismo tipo.

La siguiente figura representa un array bidimensional llamado M de 3 filas y 5 columnas:

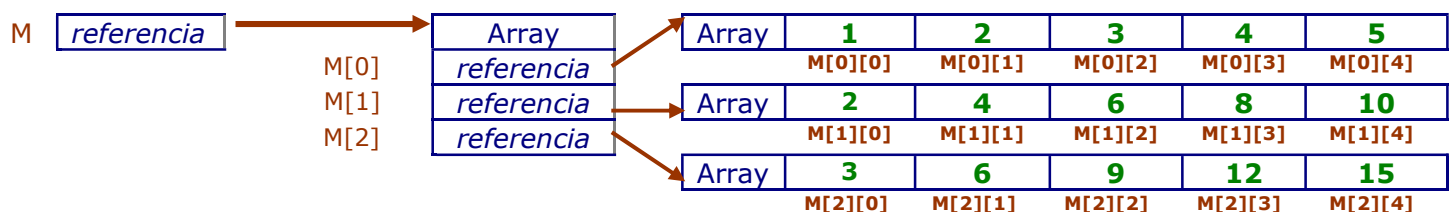
| M | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 4 | 6 | 8 | 10 |
| 2 | 3 | 6 | 9 | 12 | 15 |

A los elementos del array se accede mediante la fila y columna donde están situados.

A efectos prácticos, cuando trabajamos con arrays bidimensionales podemos pensar en una tabla como la que se muestra en la imagen anterior donde los elementos están distribuidos en filas y columnas.

Pero en realidad **un array bidimensional en Java está formado por arrays unidimensionales. Es un array de arrays.**

La disposición real en memoria del array anterior la podemos representar gráficamente así:



M es el nombre del array.

M contiene la dirección de memoria de un array unidimensional de 3 elementos.

Cada elemento de este primer array unidimensional contiene la dirección de memoria de otro array unidimensional.

Cada uno de estos últimos arrays unidimensionales contiene los valores de cada fila.



M.length indica el número de filas del array.

M[i].length indica el número de columnas de la fila i.

En este ejemplo el número de filas del array M (M.length) es 3.

La longitud de cada fila del array (M[i].length) es 5.

Para acceder a cada elemento de un array bidimensional se utilizan dos índices. El primero indica la fila y el segundo la columna.

`M[0][2] = 9;` //asigna el valor 9 al elemento situado en la primera fila [0] y tercera columna [2]

2. CREAR ARRAYS MULTIDIMENSIONALES

Se crean de forma similar a los unidimensionales, poniendo tantos pares de corchetes como dimensiones tenga:

Por ejemplo:

array bidimensional de datos de tipo `int` llamado *ventas* de 4 filas y 6 columnas:

```
int [][] ventas = new int[4][6];
```

array tridimensional de datos `double` llamado *temperaturas* de 3 filas, 4 columnas y 2 capas:

```
double [][][] temperaturas = new double[3][4][2];
```

➡ En Java se pueden crear **arrays irregulares** en los que el número de elementos de la segunda y posteriores dimensiones sea variable. Solo es obligatorio indicar el tamaño de la primera dimensión.

Por ejemplo, para crear un array de enteros de tres filas en el que cada fila podrá tener un número de columnas distinto al de las demás, se escribe la instrucción:

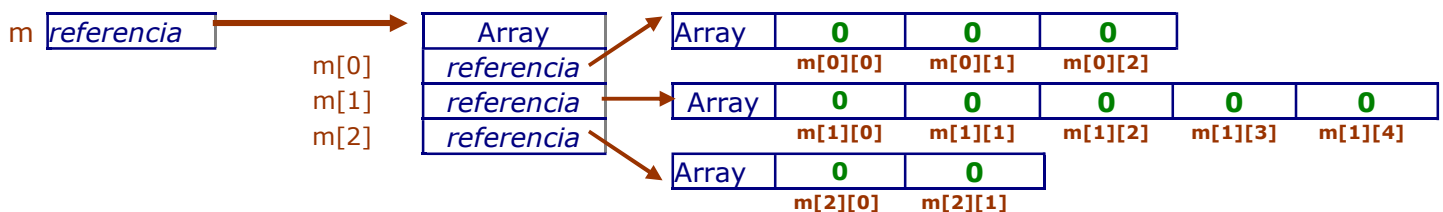
```
int [][] m = new int[3][];
```

Esto crea un array bidimensional *m* de 3 filas. La segunda dimensión (número de columnas) se deja en blanco.

A continuación a cada fila se le asigna su tamaño:

```
m[0] = new int[3];
m[1] = new int[5];
m[2] = new int[2];
```

Gráficamente podemos representar la disposición real en memoria del array anterior así:



3. INICIALIZAR ARRAYS MULTIDIMENSIONALES

Cuando se crea un array multidimensional, sus elementos toman por defecto los valores iniciales que ya vimos para los arrays unidimensionales.

También podemos dar otros valores iniciales al array cuando se crea.

Los valores iniciales se escriben entre llaves separados por comas.

Los valores que se le asignen a cada fila aparecerán a su vez entre llaves separados por comas.

El número de valores determina el tamaño del array.

Por ejemplo, la instrucción:

```
int [][] numeros = {{6,7,5}, {3, 8, 4}, {1,0,2}, {9,5,2}};
```

crea el array *numeros* de tipo `int`, de 4 filas y 3 columnas y se le asignan esos valores iniciales.

Asignando valores iniciales se pueden crear también arrays irregulares.

Por ejemplo, la instrucción:

```
int [][] a = {{6,7,5,0,4}, {3, 8, 4}, {1,0,2,7}, {9,5}};
```

crea un array *a* de 4 filas. La primera fila de 5 columnas, la segunda de 3, la tercera de 4 y la cuarta de 2.

4. RECORRER UN ARRAY MULTIDIMENSIONAL

Para recorrer un array multidimensional **se anidan tantas instrucciones for como dimensiones tenga el array**.

Ejemplo: Programa que lee por teclado números enteros y los guarda en un array bidimensional. A continuación muestra los valores leídos, el mayor y el menor y las posiciones que ocupan. El número de filas y columnas se piden por teclado. El valor mínimo para las filas y columnas es 2.

```
import java.util.Scanner;
public class Bidimensional2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int filas, columnas, mayor, menor, filaMayor, filaMenor, colMayor, colMenor;
        do{ //lectura de las filas
            System.out.print("Introduce número de filas del array: ");
            filas = sc.nextInt();
            if(filas < 2){
                System.out.println("Valor no válido");
            }
        }while(filas < 2);
        do{ //lectura de las columnas
            System.out.print("Introduce número de columnas del array: ");
            columnas = sc.nextInt();
            if(columnas < 2){
                System.out.println("Valor no válido");
            }
        }while(columnas < 2);
        //se crea el array
        int[][] A = new int[filas][columnas];

        //se introducen por teclado los elementos al array
        System.out.println("Lectura de elementos del array: ");
        for (int i = 0; i < A.length; i++) {
            for (int j = 0; j < A[i].length; j++) {
                System.out.print("A[" + i + "][" + j + "]= ");
                A[i][j] = sc.nextInt();
            }
        }

        //se muestra el contenido del array
        System.out.println("valores introducidos:");
        for (int i = 0; i < A.length; i++) {
            for (int j = 0; j < A[i].length; j++) {
                System.out.print(A[i][j] + " ");
            }
            System.out.println();
        }

        //se calcula el mayor y el menor elemento y sus posiciones (fila, columna)
        mayor = menor = A[0][0]; //se toma el primero como mayor y menor
        filaMayor = filaMenor = colMayor = colMenor = 0;
        for (int i = 0; i < A.length; i++) {
            for (int j = 0; j < A[i].length; j++) {
                if (A[i][j] > mayor) {
                    mayor = A[i][j];
                    filaMayor = i;
                    colMayor = j;
                } else if (A[i][j] < menor) {
                    menor = A[i][j];
                    filaMenor = i;
                    colMenor = j;
                }
            }
        }
    }
}
```

```
//se muestra el mayor y el menor elemento y sus posiciones (fila, columna)
System.out.print("Elemento mayor: " + mayor);
System.out.println(" Fila: " + filaMayor + " Columna: " + colMayor);
System.out.print("Elemento menor: " + menor);
System.out.println(" Fila: " + filaMenor + " Columna: " + colMenor);
}
}
```

Ejemplo: Array bidimensional irregular.

Programa que crea una matriz irregular de enteros. El número de filas se pide por teclado. Para cada fila se pedirá el número de columnas que tiene. El número mínimo de filas debe ser 2 y el número mínimo de columnas debe ser 1. A continuación asigna a cada elemento de la matriz un número aleatorio del 1 al 5. Finalmente muestra por pantalla el contenido de la matriz.

```
import java.util.Random;
import java.util.Scanner;
public class MatrizIrregular {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Random rnd = new Random();
        int filas, columnas;
        do { //lectura de las filas
            System.out.print("Introduce número de filas: ");
            filas = sc.nextInt();
            if (filas < 2) {
                System.out.println("Valor no válido");
            }
        } while (filas < 2);

        //se crea el array solo con el número de filas
        //el número de columnas se deja vacío
        //el número de columnas para cada fila lo indicaremos después

        int[][] A = new int[filas][];

        //para cada fila pedimos el número de columnas que tendrá
        for (int i = 0; i < A.length; i++) {
            do {
                System.out.print("Número de columnas para la fila " + i + ": ");
                columnas = sc.nextInt();
                if (columnas < 1) {
                    System.out.println("Valor no válido");
                }
            } while (columnas < 1);

            //Se crea un array del tamaño indicado y se le asigna a la fila
            A[i] = new int[columnas];

            //A cada elemento de la matriz se le asigna un valor aleatorio del 1 al 5
            for (int j = 0; j < A[i].length; j++) {
                A[i][j] = rnd.nextInt(5) + 1;
            }
        }

        //Se muestra el contenido de la matriz
        System.out.println("Contenido del array:");
        for (int i = 0; i < A.length; i++) {
            for (int j = 0; j < A[i].length; j++) {
                System.out.print(A[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Ejemplo de ejecución:

```
Introduce número de filas del array: 4
Introduce número de columnas para la fila 0: 3
Introduce número de columnas para la fila 1: 9
Introduce número de columnas para la fila 2: 4
Introduce número de columnas para la fila 3: 2
Contenido del array:
2 1 4
5 1 1 3 1 1 3 2 4
4 2 5 4
3 2
```

5. ARRAYS MULTIDIMENSIONALES Y MÉTODOS

Como vimos en los arrays unidimensionales, cuando se pasa un array a un método **el método recibe una copia de la referencia** (dirección de memoria) del array. Dentro del método tenemos acceso al array original y podemos modificar su contenido.

Ejemplo:

Programa que crea una matriz irregular de enteros. El número de filas se pide por teclado. Para cada fila se pedirá el número de columnas que tiene. El número mínimo de filas y columnas debe ser 2. A continuación asigna a cada elemento de la matriz un número entero que se introduce por teclado. El programa mostrará por pantalla el contenido de la matriz, el mayor elemento de cada fila y el mayor elemento de la matriz.

Se usarán los siguientes métodos:

- Método para crear la matriz. Se encargará de pedir el número de filas y columnas. Creará la matriz y la devolverá.
- Método para introducir los valores de la matriz por teclado.
- Método para mostrar por pantalla el contenido de la matriz. Si el valor a mostrar es positivo se mostrará con el signo + delante.
- Método para obtener el mayor elemento de cada fila de la matriz. El método recibirá la matriz y devolverá un array con los mayores valores de cada fila.
- Método para mostrar los mayores valores de cada fila. Este método recibirá el array obtenido en el método anterior y lo mostrará por pantalla.
- Método para obtener el mayor elemento de la matriz. El método recibirá el array con los mayores elementos de cada fila y devolverá el mayor.

```
import java.util.Scanner;
public class MatrizIrregular2 {
    //método main
    public static void main(String[] args) {
        int[][] M = crearMatriz();
        leerMatriz(M);
        mostrarMatriz(M);
        int [] N = obtenerMayoresDeCadaFila(M);
        mostrarMayoresDeCadaFila(N);
        System.out.println("Mayor elemento de la matriz: " + obtenerMayorElemento(N));
    }
}
```

```
//método para crear la matriz irregular
//se pide por teclado el número de filas
//para cada fila se pide el número de columnas
//el método devuelve la matriz creada

public static int[][] crearMatriz() {
    Scanner sc = new Scanner(System.in);
    int filas, columnas;
    do {
        System.out.print("Introduce número de filas del array: ");
        filas = sc.nextInt();
        if (filas < 2) {
            System.out.println("Valor no válido");
        }
    } while (filas < 2);

    int[][] A = new int[filas][]; //Se crea la matriz solo con las filas

    //Para cada fila se pide el columna el número de columnas
    for (int i = 0; i < A.length; i++) {
        do {
            System.out.print("Introduce número de columnas para la fila " + i + ": ");
            columnas = sc.nextInt();
            if (columnas < 2) {
                System.out.println("Valor no válido");
            }
        } while (columnas < 2);
        A[i] = new int[columnas]; //se asigna a la fila sus columnas
    }
    return A; //se devuelve la matriz creada
}

//método que recibe la matriz y asigna a cada elemento
//un valor introducido por teclado

public static void leerMatriz(int[][] A) {
    Scanner sc = new Scanner(System.in);
    System.out.println("\nIntroduce los elementos de la matriz:");
    for (int i = 0; i < A.length; i++) {
        for (int j = 0; j < A[i].length; j++) {
            System.out.print("A[" + i + "][" + j + "]= ");
            A[i][j] = sc.nextInt();
        }
    }
}

//Método que muestra el contenido de la matriz

public static void mostrarMatriz(int[][] A) {
    System.out.println("\nContenido del array:");
    for (int i = 0; i < A.length; i++) {
        for (int j = 0; j < A[i].length; j++) {
            System.out.printf("%+5d", A[i][j]);
        }
        System.out.println();
    }
}
```

```
//método que recibe una matriz y devuelve un array unidimensional
//cada elemento del array contiene el mayor valor de cada fila de la matriz
//el primer elemento del array contiene el mayor de la primera fila de la matriz
//el segundo elemento contiene el mayor de la segunda fila, etc.

public static int[] obtenerMayoresDeCadaFila(int[][] A) {

    //se crea un array con tantos elementos como filas tenga la matriz
    int[] mayores = new int[A.length];

    int mayor;

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

        //en cada fila se toma el primer elemento como el mayor
        mayor = A[i][0];

        //después se sigue buscando el mayor en el resto de la fila
        for (int j = 1; j < A[i].length; j++) {

            if (A[i][j] > mayor) {
                mayor = A[i][j];
            }

        }

        //se guarda el mayor de la fila en el array
        mayores[i] = mayor;

    }

    return mayores; //se devuelve el array
}

//método para mostrar los mayores de cada fila
public static void mostrarMayoresDeCadaFila(int [] A){
    System.out.println("\nMayores valores de cada fila:");
    for(int i = 0; i < A.length; i++){
        System.out.println("Fila " + i + ": " + A[i]);
    }
}

//método que recibe el array con los mayores elementos de cada fila
//y calcula y devuelve el mayor elemento de la matriz
public static int obtenerMayorElemento(int[] A) {
    int mayor = A[0];
    for (int i = 0; i < A.length; i++) {
        if (A[i] > mayor) {
            mayor = A[i];
        }
    }
    return mayor;
}
}
```

Ejemplo de ejecución:

```
Introduce número de filas del array: 4
Introduce número de columnas para la fila 0: 3
Introduce número de columnas para la fila 1: 5
Introduce número de columnas para la fila 2: 2
Introduce número de columnas para la fila 3: 4

Introduce los elementos de la matriz:
```

```
A[0][0]= 19
A[0][1]= 12
A[0][2]= 30
A[1][0]= 1
A[1][1]= 5
A[1][2]= 10
A[1][3]= 6
A[1][4]= 9
A[2][0]= -7
A[2][1]= -9
A[3][0]= 4
A[3][1]= 6
A[3][2]= -2
A[3][3]= 3
```

Contenido del array:

```
+19 +12 +30
+1  +5 +10  +6  +9
-7  -9
+4  +6  -2  +3
```

Mayores valores de cada fila:

Fila 0: 30

Fila 1: 10

Fila 2: -7

Fila 3: 6

Mayor elemento de la matriz: 30