

Tema 5: Matrices

Antonio J. Sierra

Índice

- Introducción
- Matrices unidimensionales
- Matrices multidimensionales
- Cadenas

Introducción

- Grupo de variables del mismo tipo a las que se hace referencia con el mismo nombre.
- Ofrecen un medio de agrupar información relacionada.
- Una o más dimensiones.
- Cualquier tipo
- Acceso mediante índice.

Matrices unidimensionales

- Colección de variables del mismo tipo accesible mediante un índice.
- Dos pasos:
 - Declaración: **Tipo NombreMatriz [];**
(**NombreMatriz** es asignado a **null**).
 - Reserva memoria:
NombreMatriz = new Tipo [tamaño];
operador **new**.

Ejemplo

- Declaración:

```
int tabla[];
```

- Reserva

```
tabla = new int[10];
```

- Inicialización a 0.

- Uso

```
tabla[0] = 10;
```

En un solo paso

- Declaración y reserva

```
int tabla[] = new int [10];
```

Inicialización en la declaración

- Declaración y reserva

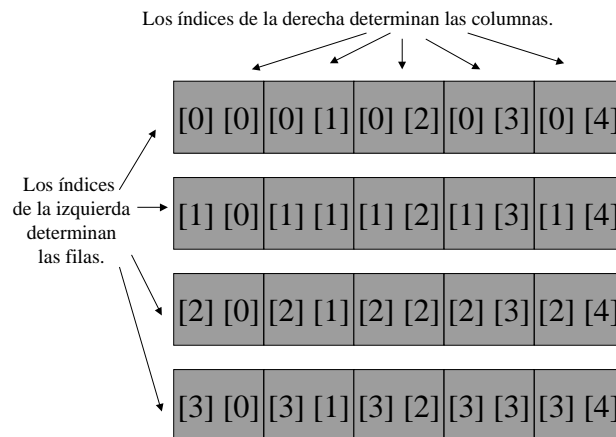
```
int tabla[] = {0, 1, 2,  
3, 4, 5, 6, 7, 8, 9};
```

Matrices multidimensionales

- Son matrices de matrices
- Diferencias:
 - La declaración de variables del tipo matriz multidimensional necesita índices adicionales
- Declaración y reserva

```
int tablaBi[] = new int [4][5];
```

```
int tablaBi [][] = new int[4][5];
```

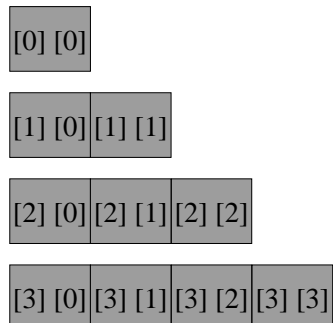


Otra forma

```
int tablaBi[][] = new int[4][];  
tablaBi[0] = new int [5];  
tablaBi[1] = new int [5];  
tablaBi[2] = new int [5];  
tablaBi[3] = new int [5];
```

Matrices multidimensionales irregulares

```
int tablaBi[][] = new int[4][];  
tablaBi[0] = new int [1];  
tablaBi[1] = new int [2];  
tablaBi[2] = new int [3];  
tablaBi[3] = new int [4];
```



Inicialización de matrices multidimensionales

```
int m[][] ={{ 0*0, 1*0, 2*0, 3*0},
            { 0*1, 1*1, 2*1, 3*1},
            { 0*2, 1*2, 2*2, 3*2},
            { 0*3, 1*3, 2*3, 3*3}};
```

Sintaxis alternativa

- Para declarar una matriz se puede utilizar una segunda forma.
`Tipo[] NombreMatriz;`
- Son equivalentes:
`int tabla[] = new int[3];`
`int[] tabla = new int[3];`
- Son equivalentes:
`char DosDim[][] = new char[3][4];`
`char[][] DosDim = new char[3][4];`

Cadenas

- Secuencias de caracteres
- La clase **String** (paquete **java.lang**).
- El método **main** usa **String** en su declaración de **args: String args[]**.
- Los **String** son inmutables (no se pueden cambiar una vez que se han creado).
- **StringBuffer**, que se usa para crear y manipular caracteres de datos que se cambian.
- Concatenación **String** con el operador **+**.

Ejemplo

```
String str = "Esto es un cadena";  
System.out.println(str);
```

```
String s = "Hola";
```

```
String s = new String("Hola");
```


StringBuffer

```
StringBuffer sb = new StringBuffer ("Hola");  
String s = new String(sb);
```

Ejemplo: Concatenación

- Para concatenar cadenas se puede utilizar el signo

+:

```
String s3 = s2 + s1;
```

```
String s;
```

```
s = "Hola";
```

```
s = " Adios ";
```

```
String ss;
```

```
ss = "Hola";
```

```
ss += "Adios";
```