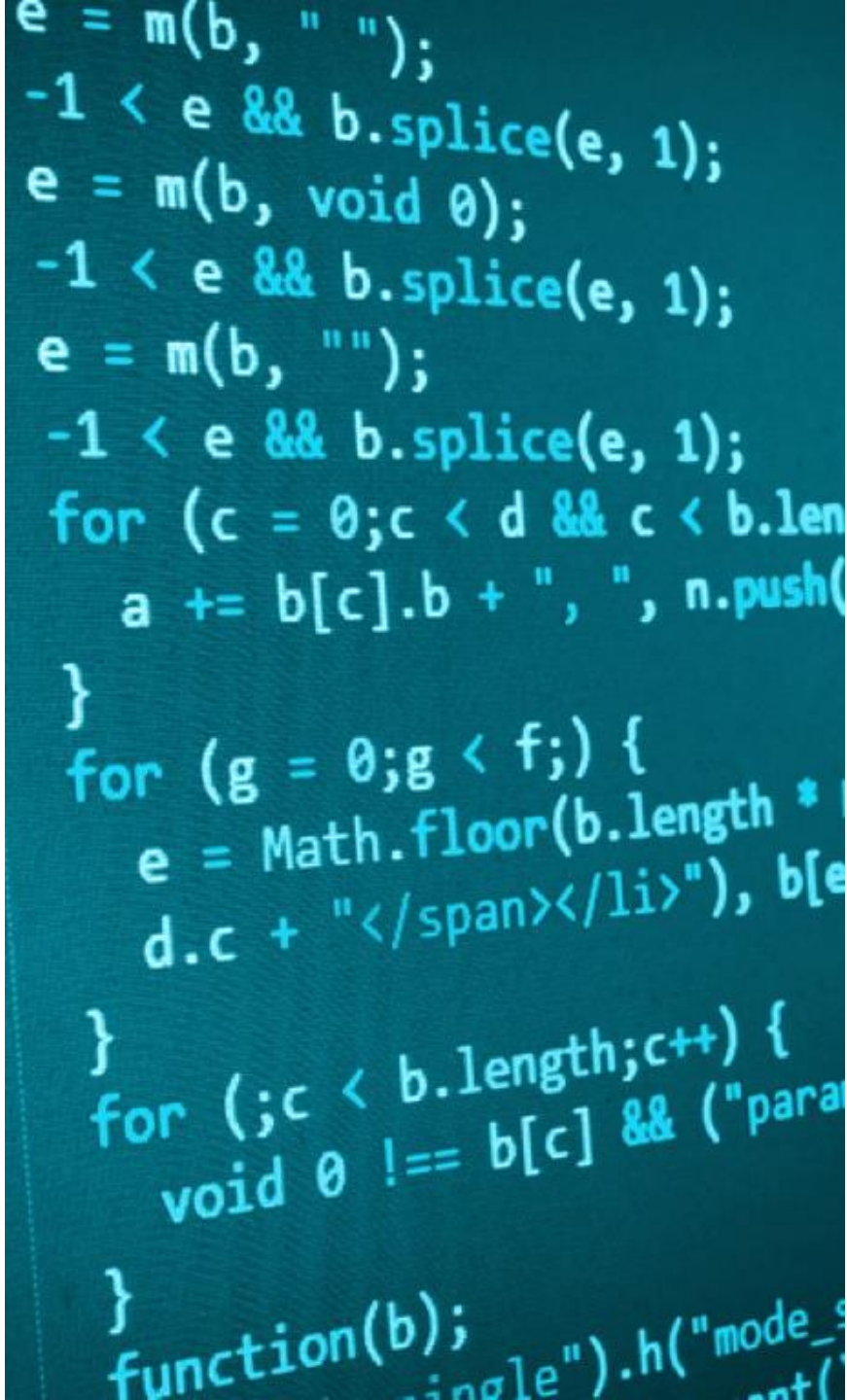
# UD3. Arrays bidimensionals

Verónica Mascarós

**Curs 23-24**



# Arrays bidimensionals en Java

- Els arrays poden tenir més d'una dimensió.
- Un array multidimensional és aquell que per accedir a una posició concreta, en comptes d'utilitzar un únic valor com a índex, s'utilitza una seqüència de diversos índexs.
  - Cada índex serveix com a coordenada per a una dimensió diferent.
- Els més utilitzats són els arrays de **2 dimensions**, més coneguts com a **matrius**.

# Creació de matrius en Java

- Un array bidimensional (matriu) es pot interpretar com una taula on la primera dimensió serien les **files** i la segona dimensió serien les **columnes**.
- Si per exemple, necessitem emmagatzemar les vendes per ciutat, podríem representar 10 ciutats, i de cada ciutat, tenir 5 vendes:
  - int[][] vendes = new int[10][5];

# Creació de matrius en Java

- La sintaxi és com la dels arrays unidimensionals però se li afegeixen uns claudàtors [] addicionals.

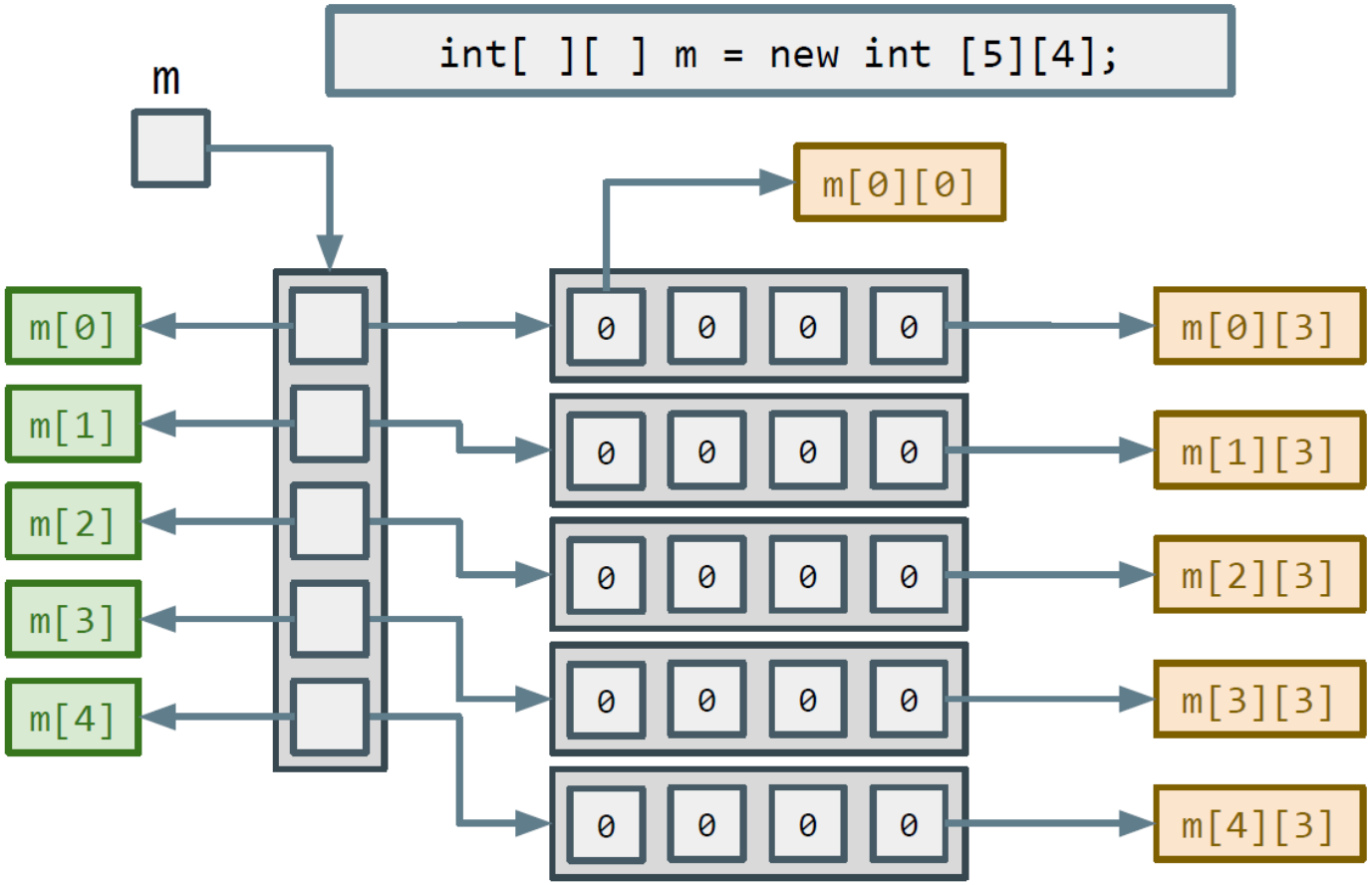
```
Tipo[][] identificadorVariable = new Tipo[numeroFilas][numeroColumnas];
```

- Exemples creació matriu:

```
char letras[][];

int precios[][] = new int[5][6];

double[][] notas = new double[10][10];

String ciudades[][] = new String[10][5];
```

# Creació de matrius en Java

- Exemple creació matriu: int **[][]** mat = new int[5][4];

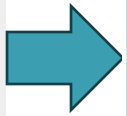



5 files x 4 columnes

# Creació de matrius en Java

- També podem inicialitzar els valors d'una matriu posant els elements de cada Array intern entre claus. Exemple

```
int[][] matriz = {
   {1,2,3,4,5},
   {6,7,8,9,10},
  {11,12,13,14,15}
         };
```



| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 |

# Ús de matrius en Java

- L'accés a les posicions d'una matriu o array bidimensional seria:

```
identificadorArray[índiceFila][índiceColumna]
```

A continuació es mostren les posicions o índex de cada element d'una matriu de 4 files x 5 columnes

int[][] mat = new int[4][5];

| | | | | |
|---|---|---|---|---|
| a[0][0] | a[0][1] | a[0][2] | a[0][3] | a[0][4] |
| a[1][0] | a[1][1] | a[1][2] | a[1][3] | a[1][4] |
| a[2][0] | a[2][1] | a[2][2] | a[2][3] | a[2][4] |
| a[3][0] | a[3][1] | a[3][2] | a[3][3] | a[3][4] |

# Ús de matrius en Java

- Coneixent  les posicions, podem assignar valors a cada element. Exemple:
- int[][] matriz = {{1,4,5},{6,7,2},{8,3,8}};
- o bé
- int[] matriz = new int[3][3]; //9 elements
- //emplenar "manualment"
- matriz[0][0]=1;  matriz[0][1]=4; matriz[0][2]=5;

# Ús de matrius en Java

- **Comportament de length**
- Podem obtenir tant la quantitat de files com la quantitat d'elements de cada fila. Exemple:
- int[][] mat = new int[4][5];
- mat.length --> 4 (nº de files)
- mat[index].length --> mat[0].length = 5 (nº de columnes, nº d'elements per fila)

# Recórrer matrius

- Per recórrer un array de diverses dimensions ho podem fer amb dos bucles **for** imbricats ("niats").
- Un bucle imbricat utilitzarà dos índexs, cada índex s'utilitzarà per a una dimensió diferent.
  - Si tinguerem un Array de 3 dimensions, es necessiten 3 bucles niats amb 3 índexs, encara que no és una estructura gens habitual.

# Recórrer matrius

- En matrius (arrays bidimensionals), la condició del bucle extern ha de fer referència a la quantitat de **files**.
- Mentre que la condició del bucle intern ha de fer referència a la quantitat d'elements de la fila actual (**columnes**).

```
int[][] matriz = {{1,4,5},{6,7,2},{8,3,8}};
int filas = matriz.length; //3
int elementosFila0 = matriz[0].length; //3
int elementosFila1 = matriz[1].length; //3
int elementosFila2 = matriz[2].length; //3
```

# Recórrer matrius

- Estructura exemple:

```
public class Ejemplo_Recorrer {
    public static void main(String[] args) {
        //declaración y inicialización
        int[][] matriz = {{1,4,5},{6,7,2},{8,3,8}};
        //recorrido de la matriz
        for (int fila = 0; fila < matriz.length; fila++) {
            for (int columna = 0; columna < matriz[fila].length; columna++) {
                System.out.print(matriz[fila][columna]+" ");
            }
            System.out.println();
        }
    }
}
```

# Recórrer matrius

- També podem utilitzar l'estructura **for-each.**
- Per Arrays de 2 dimensions podem fer-ho de forma abreujada de la següent forma. Exemple:

```
int[][] array3 = {{1,4,5},{6,7,8},{3,8}};
for(int[] fila : array3) {
    for(int columna : fila) {
        System.out.print(columna);
    }
} //14567838
```

# Recórrer matrius

- Si el que volem és mostrar (imprimir per pantalla) el contingut de cada fila:

```
int[][] array4 = {{3,8,5},{4,1,8,4},{5,2}};
for(int[] fila : array4) {
    System.out.println(Arrays.toString(fila));
}
```

- Mostraria: [3, 8, 5]

  [4, 1, 8, 4]

  [5, 2]