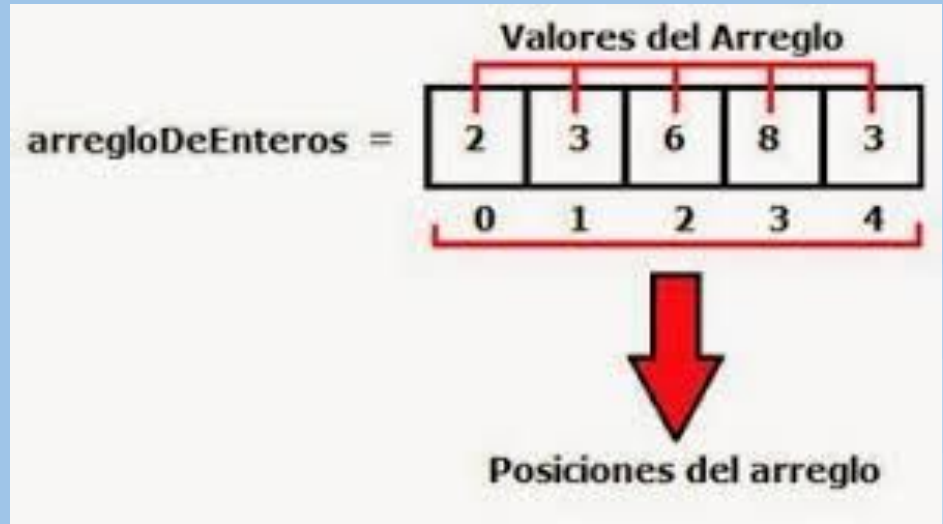


- Definición - *se define el tipo*
- Creación - *espacio en memoria*
- Inicialización - *valores iniciales*
- Uso
- Dimensión - *tamaño*

```
int[] arr
```

```
arr = new int[10]
```

```
int[] arr = {2, 3, 4}
```





Carga de datos...

- En la declaración

```
double[] medidas = {5.1, 3.02, 9.65};
```

-

- En tiempo de ejecución

```
int[] medidas = new int[10];  
for(int i = 0; i < medidas.length; i++)  
    medidas[i] = 0;
```





Seudocódigo y java

ENTERO [] arr

ENTERO i

arr ← CREAR ENTERO[10]

PARA i ← 0 HASTA **LONGITUD(arr)-1**

arr[i] ← 0

FIN PARA

.....

int[] arr

arr = **new int[10];**

int i;

for (i = 0; i < **arr.length**; i++) {

arr[i] = 0;

}



Preguntas

- Se puede dimensionar en ejecución?
- Cuando ocurren errores y de qué tipo?
- Se puede pasar un arreglo por parámetro?
- Se puede retornar un arreglo?



Dimensión de un arreglo

```
double[] lluviaMensual = new double[12];  
for (int i=0; i< 12; i++)  
    System.out.println(lluviaMensual[i]);
```

```
double[] lluviaMensual;  
int longitud;  
  
Longitud = TecladoIn.readLineInt();  
lluviaMensual = new double[longitud];  
  
for (int i=0; lluviaMensual.length; i++)  
    System.out.println(lluviaMensual[i]);
```

- *Qué imprime?*

```
int i;  
int[] a = new int[3];  
int[] b = new int[3];  
for(i=0; i < a.length; i++)  
    a[i] = i; // a = [0 1 10]  
b = a;  
System.out.println(a[2] + " " + b[2]); // 2 2  
b[2] = 1; a[2] = 10;  
System.out.println(a[2] + " " + b[2]); // 10 10
```

```
int i;  
int[] a = new int[3];  
int[] b = new int[3];  
  
for(i=0; i < a.length; i++)  
    a[i] = 0;  
for(i=0; i < b.length; i++)  
    b[i] = 0;  
  
System.out.println("a es igual a b" + a==b );
```



Arreglos llenados parcialmente

- *Los elementos no inicializados contienen datos basura (garbage)*
- *No existe un mecanismo automático para detectar cuántos elementos han sido cargados*
- *Se utiliza variable*
- *Se puede llenar todo el arreglo con un valor imposible (valor centinela)*



Usar una variable `cantEntradas` para determinar cantidad de elementos ingresados

`entrada[0]`

leche

`entrada[1]`

llamar a casa

`entrada[2]`

ir a la playa

`entrada[3]`

`entrada[4]`

`cantEntradas - 1`

Valores basura

`cantEntradas` tiene valor 3
`entrada.length` tiene valor 5



Arreglos multidimensionales

- Los arreglos pueden tener más de un índice
- Número de dimensiones = número de índices
- Los arreglos con más de dos dimensiones son una extensión de los arreglos bidimensionales (2D)
- Un arreglo 2D representa una tabla o grilla



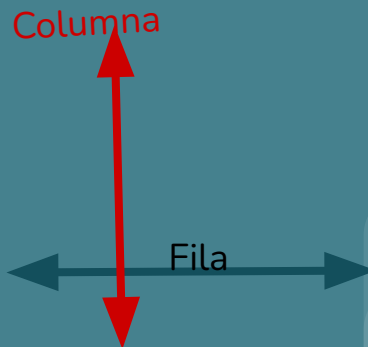


Bidimensionales

| | | | |
|------|---------|--------|--------|
| | Columns | | |
| Rows | (0, 0) | (0, 1) | (0, 2) |
| | (1, 0) | (1, 1) | (1, 2) |
| | (2, 0) | (2, 1) | (2, 2) |

Element position: (row, column)

- Una dimensión es la fila
- La otra dimensión es la columna
- Una celda es la intersección entre una fila y una columna
- Un elemento del arreglo es una celda de la tabla



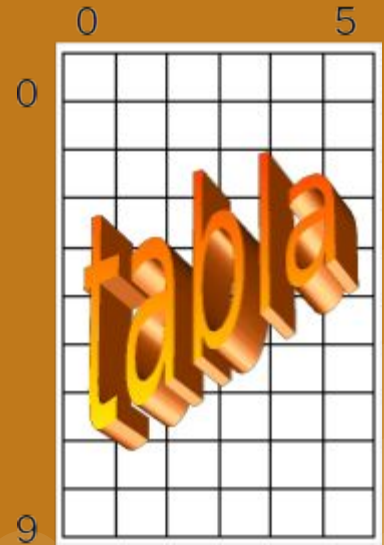
CREAR arreglos 2D

Sintaxis similar a los arreglos 1D

Declaración de un arreglo bidimensional tabla:

```
int[][] tabla = new int[10][6];
```

tabla tiene diez filas y seis columnas



- *Tabla como arreglo 2D*

| Indices | 0 | 1 | 2 | 3 | 4 | 5 |
|---------|--------|--------|--------|--------|--------|--------|
| 0 | \$1050 | \$1055 | \$1060 | \$1065 | \$1070 | \$1075 |
| 1 | \$1103 | \$1113 | \$1124 | \$1134 | \$1145 | \$1156 |
| 2 | \$1158 | \$1174 | \$1191 | \$1208 | \$1225 | \$1242 |
| 3 | \$1216 | \$1239 | \$1262 | \$1286 | \$1311 | \$1335 |
| 4 | \$1276 | \$1307 | \$1338 | \$1370 | \$1403 | \$1436 |
| ... | ... | ... | ... | ... | ... | ... |

Índice fila **3**
(4ta fila)

Índice
columna **4**
(5ta
columna)

- Dos índices: **[fila][columna]**
- Primera dimensión: índice **fila**
- Segunda dimensión: índice **columna**
- La tabla corresponde al arreglo 2D balance
- balance**[3][4]** = celda en la **4ta fila** y **5ta columna**
- balance**[3][4]** = \$1311 (en amarillo)



Bidimensionales

- Los arreglos Multidimensionales se implementan como arreglos de arreglos.

Ejemplo: `int[][] tabla = new int[3][4];`

- `tabla` es un arreglo unidimensional de longitud 3
- cada elemento en `tabla` es un arreglo de tipo base `int`.
- Se accede a cada fila por un subíndice: `tabla[0].length` es 4

`tabla.length` es 3
`tabla[0].length` es 4

| primer | fila del | arreglo | |
|--------|----------|---------|--|
| | | | |
| | | | |

`tabla[0]` se refiere a la primer fila del arreglo, la cual es un arreglo unidimensional.

Arreglos MULTIDIMENSIONALES

Java no provee arreglos multidimensionales

Pero se pueden crear arreglos de arreglos

Se pueden crear arreglos de arreglos de arreglosy así sucesivamente.

Ejemplo de un arreglo de dos dimensiones

```
int [][] dosDim = new int [2][];  
dosDim [0] = new int[5];  
dosDim [1] = new int[5];  
dosDim [2] = new int[5];
```

- *Cómo recorrer?*

```
public void mostrarMatriz (int[] [] mat,  
int cantFila, int cantCol) {  
    int fila, col;  
  
    for (fila = 0; fila < cantFila; fila++) {  
        for (col = 0; col < cantCol; col++)  
            System.out.println(mat[fila][col]);  
        }  
    }
```




Arreglos no uniformes (Ragged)

1. Los Arreglos No Uniformes tienen filas de longitudes distintas
2. Java permite usar Arreglos No Uniformes

- Ejemplo: arreglo 2D int con:
5 elementos en la primer fila,
7 en la segunda fila, y
4 en la tercera fila:

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

```
b[0] = new int[5];
```

```
b[1] = new int[7];
```

```
b[2] = new int[4];
```



=, == y parámetros en Arreglos

| | Tipo Primitivo | Arreglo Completo |
|----------------------|-------------------------------------|---------------------------------------|
| Asignación (=) | Copia contenido | Copia dirección |
| Igualdad (==) | Compara contenido | Compara dirección |
| Pasaje de Parámetros | <i>Pasa el valor</i> (contenido) | <i>Pasa la referencia</i> (dirección) |