

# Java: Arreglos

Dr. Mario Marcelo Berón

# Índice

- 1 Arreglos
  - Algunos Métodos de Arreglos Útiles
- 2 Arreglos Multidimensionales

# Arreglos

- Tipo de dato que funciona como un contenedor para almacenar colecciones de elementos.
- Todos los elementos son del mismo tipo aunque no hay restricción sobre el tipo de los elementos.
- Creación de un arreglo:
  - ▶ Declarar la variable arreglo.
  - ▶ Distribuir memoria para los elementos del arreglo.
  - ▶ Una vez que se define el tamaño no se puede cambiar.

## Ejemplo - Creación de Arreglo

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

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

# Arreglos



- Las ubicaciones de memoria distribuidas son continuas.
- Cada variable se inicializa con un valor. Si son de tipo int o float se inicializan en 0. Si son char con un caracter unicode que representa el caracter vacío, si son boolean en falso, etc.
- Cuando los valores del arreglo son conocidos de antemano el arreglo se puede crear sin usar el operador new.

## Ejemplo - Creación de Arreglo sin new

```
double[] temperatura = {9, 11.5, 11, 8.5, 7, 9, 8.5} ;
```

# Arreglos



- Lectura: `nombreDelArreglo [índice]`. Donde índice tiene que ser un índice válido en el arreglo. Los índices comienzan en 0.
- Escritura: `nombreDelArreglo [índice] = expresión`. Donde índice tiene que ser un índice válido en el arreglo. Los índices comienzan en 0.
- Índice: el índice de un arreglo puede ser cualquier expresión que produzca como resultado un entero.
- Todos los arreglos tienen una primitiva (`length`) que retorna la longitud. Así si `a` es un arreglo `a.length` retorna la longitud de `a`. Esta característica es útil para usar en loop *for*.

# Arreglos

- Los arreglos pueden ser usados como parámetros y como valores de retorno.
- Para especificar que un método recibe como parámetro un arreglo se especifica el tipo y se informa que la variable es un arreglo. Ejemplo:

## Arreglo como Parámetro Formal

```
tipoRetornoMetodo nombreMetodo(tipo [] nombreArreglo ,...)  
    {...}
```

- Cuando se envía el arreglo como parámetro real solo se necesita colocar el nombre del mismo.
- Cuando se pasa un arreglo como parámetro lo que en realidad se pasa es la dirección base.

# Arreglos

- La característica *varargs* (variable arguments) permite enviar un número variable de datos del mismo tipo a un método.
- El acceso a los elementos del *varargs* se lleva a cabo igual que un arreglo.
- El *vararg* da más flexibilidad a la hora de invocar dado que no solo se puede invocar con arreglos sino también con diferentes variables.
- Los *varargs* deben ser el último elemento en la lista de parámetros.

## Ejemplo de *varargs*

```
static void mostrarTemps(double ... temperatura)
{
    // código
}
```

# Arreglos

- Para indicar que un método retorna un arreglo como parámetro el tipo de retorno del método tiene que ser un arreglo. En este caso lo que se retorna es una dirección a la base del arreglo.

## Retorno de un Arreglo

```
tipoDeRetorno [] nombreMetodo (...) {...}
```

- Java provee un loop for aumentado que permite recorrer arreglos sin utilizar un índice.

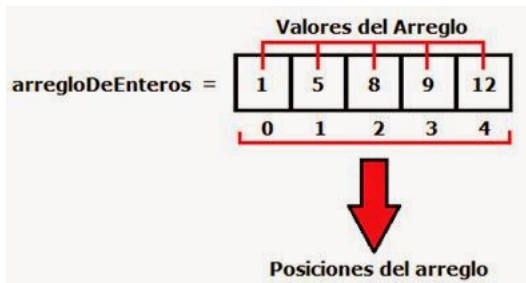
## Recorrido de un Arreglo

```
for(tipoBaseDelArreglo variable: Arreglo) {...}
```

- Este tipo de loops no se utilizan para modificar el arreglo dado a que, si bien no causan un error de compilación, puede ocasionar que el programa no se comporte adecuadamente.



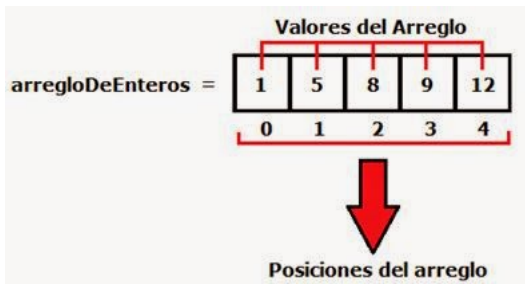
# Arreglos



Se utiliza un for aumentado cuando:

- Se desea acceder al arreglo completo.
- Se desea leer los elementos del arreglo y no modificarlos.
- No se requiere un índice para el procesamiento.

# Arreglos



Se utiliza un for aumentado cuando:

- Encontrar el máximo valor en un arreglo.
- Sumar un arreglo.
- Determinar si un elemento pertenece o no a un arreglo.
- Determinar la posición de un elemento en el arreglo.
- Etc.

# Arreglos Multidimensionales

Un arreglo multidimensional es un arreglo que tiene más de un índice. Para crear un arreglo multidimensional simplemente se proveen los índices.

## Ejemplo

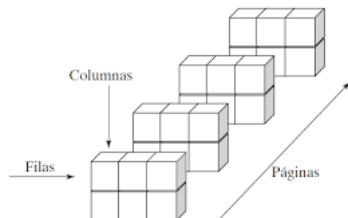
```
double [] [] temperatura = new double [4][7];
```

Para acceder a un elemento del arreglo se requiere especificar los índices del elemento.

## Ejemplo

```
temperatura [1][2] //es el elemento que está en la fila 1 columna 2.
```

# Arreglos Multidimensionales

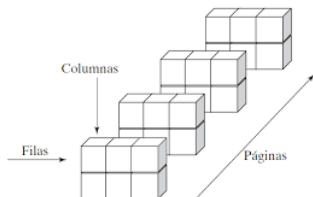


Es posible declarar e inicializar un arreglo multidimensional con una colección de valores.

## Ejemplo

```
int [] [] matriz = {{1,2,3},{4,5,6}};
```

# Arreglos Multidimensionales



Para procesar un arreglo bidimensional se utilizan loops anidados. Un loop para cada índice del arreglo.

## Ejemplo

```
for(i=0;i<arreglo.length;i++){  
    for(j=0;j<arreglo[i].length;j++){  
        Sentencias para procesar el arreglo  
    }  
}
```

# Arreglos Multidimensionales

arr [0]	10	9	8		
arr [1]	7	5	6	88	
arr [2]	30	15			
arr [3]	90				
arr [4]	10	20	30	40	50

Existe la posibilidad de definir arreglos irregulares es decir arreglos en donde la cantidad de columnas pueden diferir.

## Ejemplo

```
char [][] animales={{ 'M', 'O', 'N', 'O' },  
                    { 'G', 'A', 'T', 'O' },  
                    { 'P', 'A', 'J', 'A', 'R', 'O' }  
}
```

# Arreglos Multidimensionales

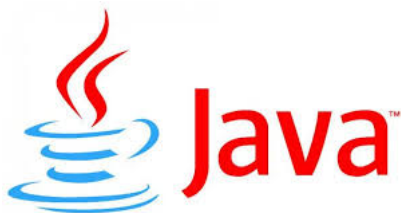
## Ejemplo

```
char [][] animales={{ 'M', 'O', 'N', 'O' },  
                    { 'G', 'A', 'T', 'O' },  
                    { 'P', 'A', 'J', 'A', 'R', 'O' }  
                    }
```

...

```
for(int i=0;i<animales.length;i++){  
    for(int j=0;j<animales[i].length;j++){  
        procesar elemento  
    }  
}
```

# Arreglos Multidimensionales



## Ejemplo

```
char [][] animales = new char [3][];
```

```
animales[0] = new char [6];
```

```
animales[1] = new char [3];
```

```
animales[2] = new char [4];
```