

# 8

## Creación y uso de matrices

edacion@proydesa.org) has a  
this Student Guide.

## Introducción a las matrices

- Una matriz es un objeto contenedor que incluye un grupo de valores de un único tipo.
- Un valor de la matriz puede ser un tipo primitivo o un tipo de objeto.
- La longitud de una matriz se establece al crearla.
- Tras la creación, la longitud de una matriz no se puede cambiar.
- Cada elemento de una matriz se denomina *elemento*.
- A cada elemento se accede mediante un índice numérico.
- El índice del primer elemento es 0 (cero).

# Matrices unidimensionales

Ejemplo:

```
int ageOne = 27;  
int ageTwo = 12;  
int ageThree = 82;  
int ageFour = 70;  
int ageFive = 54;  
int ageSix = 6;  
int ageSeven = 1;  
int ageEight = 30;  
int ageNine = 34;  
int ageTen = 42;
```

# Creación de matrices unidimensionales

Matriz de tipos `int`

27	12	82	70	54	1	30	34
----	----	----	----	----	---	----	----

Matriz de tipos `Shirt`

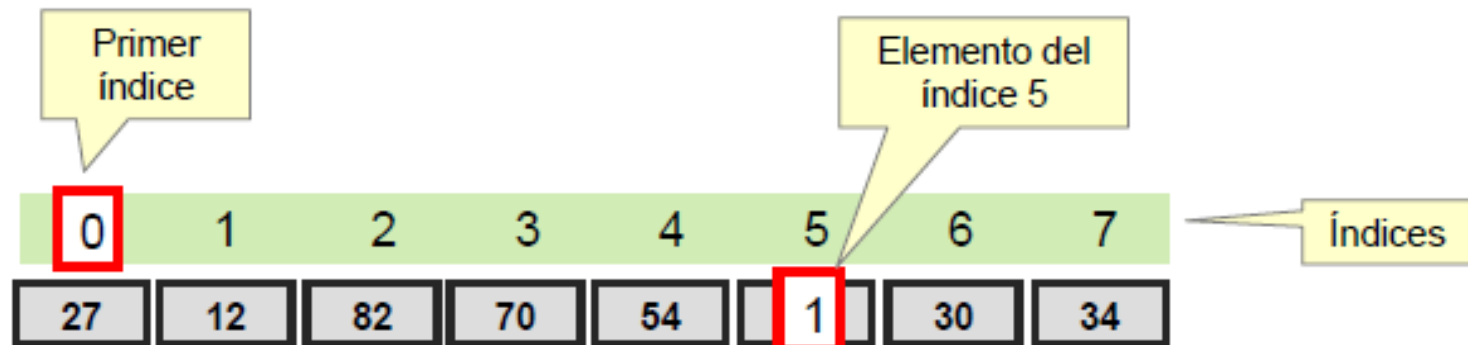


Matriz de tipos `String`

Hugh Mongus  
Aaron Datires  
Stan Ding  
Albert Kerkie  
Carrie DeKeys  
Walter Mellon  
Hugh Morris  
Moe DeLawn

# Índices y longitud de matriz

Matriz ages de ocho elementos



La longitud de matriz es 8  
(`ages.length`)

# Declaración de una matriz unidimensional

- Sintaxis:

```
type [] array_identifier;
```

- Declarar matrices de tipos `char` e `int`:

```
char [] status;  
int [] ages;
```

- Declarar matrices de referencias de objetos de tipo `Shirt` y `String`:

```
Shirt [] shirts;  
String [] names;
```

# Instanciación de una matriz unidimensional

- Sintaxis:

```
array_identifier = new type [length];
```

- Ejemplos:

```
status = new char [20];  
ages = new int [5];  
  
names = new String [7];  
shirts = new Shirt [3];
```

# Inicialización de una matriz unidimensional

- Sintaxis:

```
array_identifier[index] = value;
```

- Definir valores en la matriz `ages`:

```
ages[0] = 19;  
ages[1] = 42;  
ages[2] = 92;  
ages[3] = 33;
```

- Definir referencias en los objetos `Shirt` de la matriz `shirts`:

```
shirts[0] = new Shirt();  
shirts[1] = new Shirt();  
shirts[2] = new Shirt();
```



# Declaración, instanciación e inicialización de matrices unidimensionales

- Sintaxis:

```
type [] array_identifier = {comma-separated list of values  
or expressions};
```

- Ejemplos:

```
int [] ages = {19, 42, 92, 33, 46};  
Shirt [] shirts = {new Shirt(), new Shirt(), new Shirt()};
```

- No permitida (NetBeans mostrará un error):

```
int [] ages;  
ages = {19, 42, 92, 33, 46};
```

## Acceso a un valor de una matriz

- Definición de un valor:

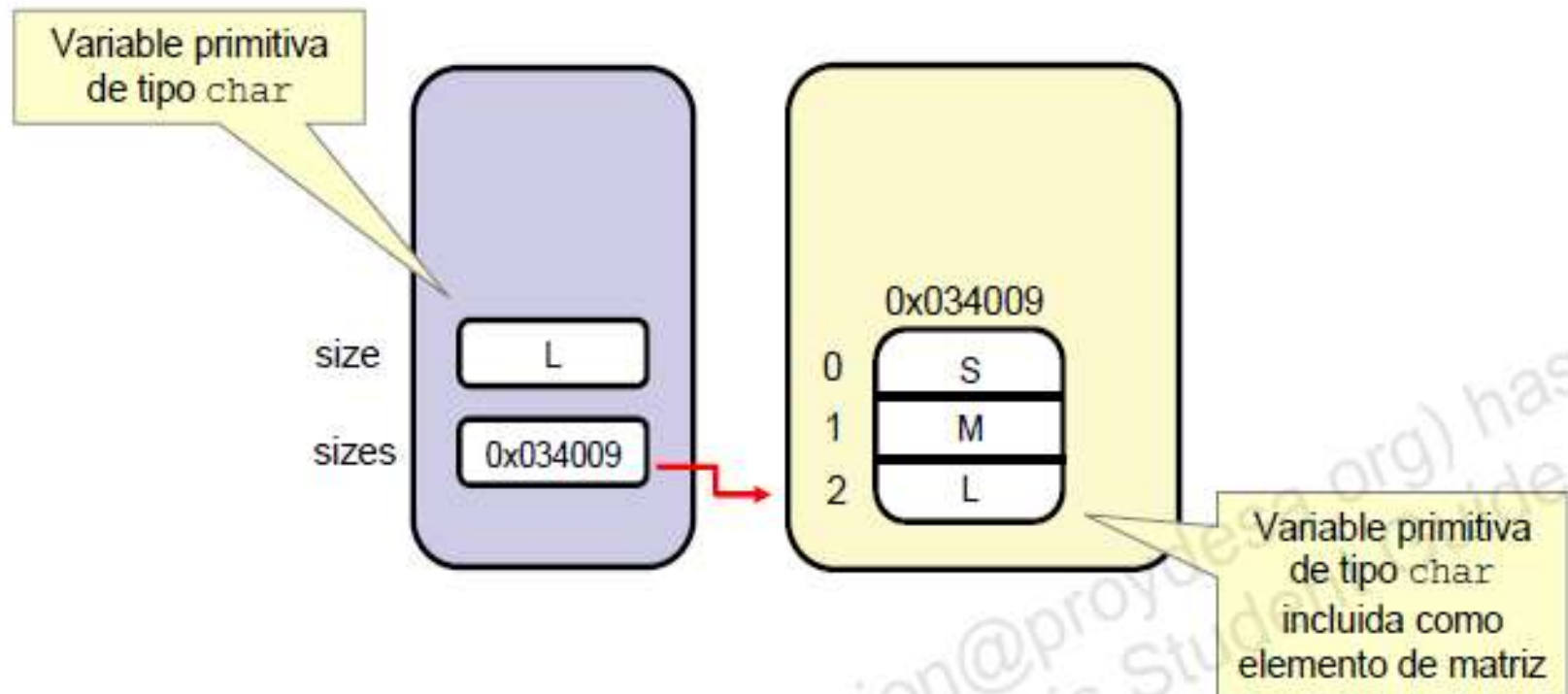
```
status[0] = '3';  
names[1] = "Fred Smith";  
ages[1] = 19;  
prices[2] = 9.99F;
```

- Obtención de un valor:

```
char s = status[0];  
String name = names [1];  
int age = ages[1];  
double price = prices[2];
```

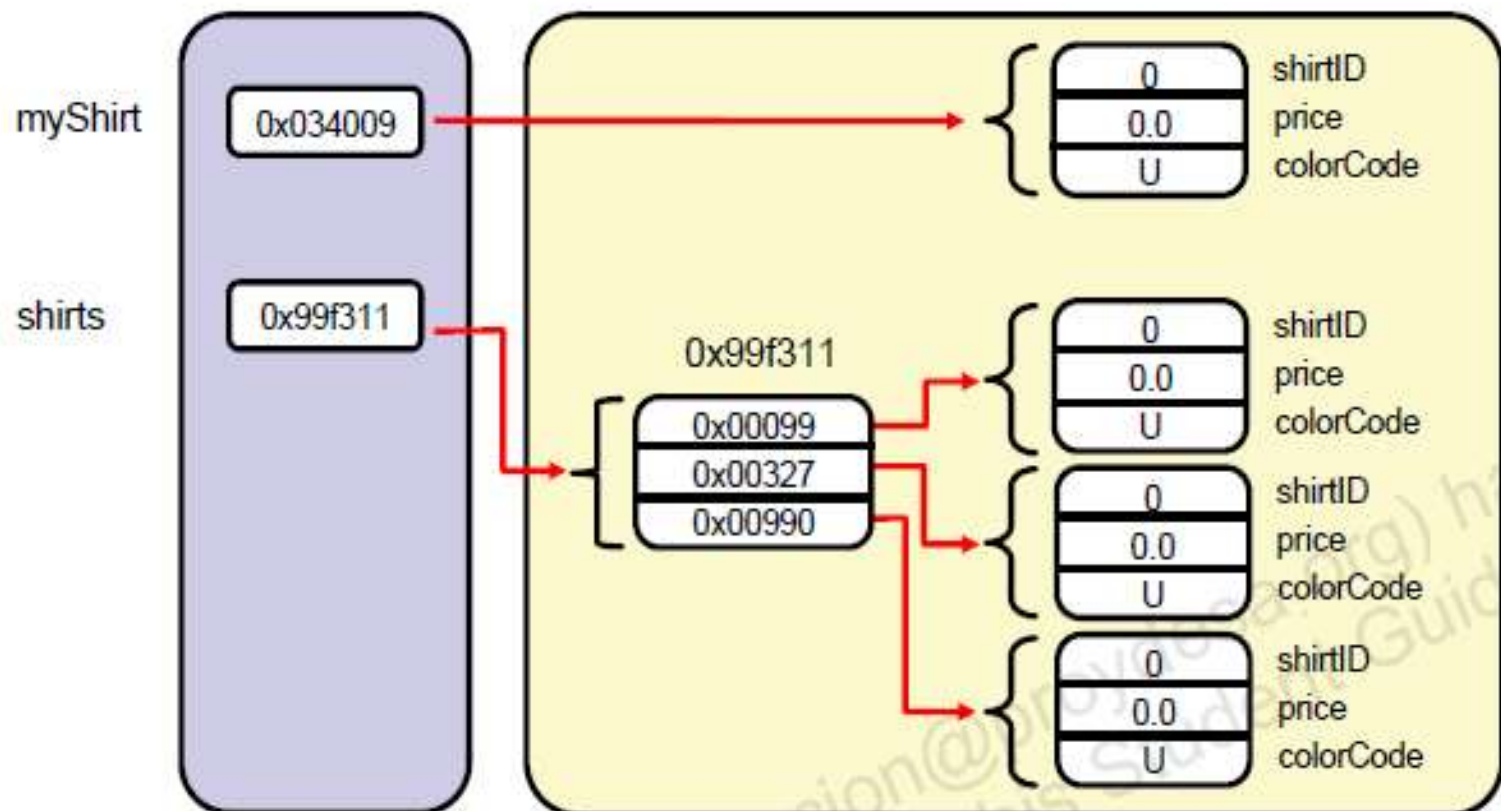
# Almacenamiento de matrices en memoria

```
char size = 'L'  
char[] sizes = {'S', 'M', 'L'};
```



# Almacenamiento de matrices de referencias en memoria

```
Shirt myShirt = new Shirt();  
Shirt[] shirts = { new Shirt(), new Shirt(), new Shirt() };
```



# Uso de la matriz args en el método main

- Los parámetros se pueden escribir en la línea de

```
> java ArgsTest Hello World!  
args[0] is Hello  
args[1] is World!
```

El primer parámetro va en args[0].

El segundo parámetro va en args[1], etc.

- Código para recuperar los parámetros:

```
public class ArgsTest {  
    public static void main (String[] args) {  
        System.out.println("args[0] is " + args[0]);  
        System.out.println("args[1] is " + args[1]);  
    }  
}
```



# Conversión de argumentos String en otros tipos

- Los números se pueden introducir como parámetros:

```
> java Argstest 2 3
```

```
Total is: 23
```

Concatenación, no adición.

```
Total is: 5
```

- Conversión de String en int:

```
public class Argstest {  
    public static void main (String[] args) {  
        System.out.println("Total is: " + (args[0] + args[1]));  
  
        int arg1 = Integer.parseInt(args[0]);  
        int arg2 = Integer.parseInt(args[1]);  
        System.out.println("Total is: " + (arg1 + arg2));  
    }  
}
```

Son argumentos String.

Integer.parseInt  
Int() se  
convierte en int.

Observe los paréntesis.

# Descripción de matrices bidimensionales

	Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Semana 1							
Semana 2							
Semana 3							
Semana 4							

## Declaración de una matriz bidimensional

- Sintaxis:

```
type [][] array_identifier;
```

- Ejemplo:

```
int [][] yearlySales;
```

(fundacion@proydesa.org) has a  
to use this Student Guide.



# Instanciación de una matriz bidimensional

- Sintaxis:

```
array_identifier = new type [number_of_arrays] [length];
```

- Ejemplo:

```
// Instantiates a 2D array: 5 arrays of 4 elements each  
yearlySales = new int[5][4];
```

	Trimestre 1	Trimestre 2	Trimestre 3	Trimestre 4
Año 1				
Año 2				
Año 3				
Año 4				
Año 5				

# Inicialización de una matriz bidimensional

Ejemplo:

```
yearlySales[0][0] = 1000;  
yearlySales[0][1] = 1500;  
yearlySales[0][2] = 1800;  
yearlySales[1][0] = 1000;  
yearlySales[3][3] = 2000;
```

	Trimestre 1	Trimestre 2	Trimestre 3	Trimestre 4
Año 1	1000	1500	1800	
Año 2	1000			
Año 3				
Año 4				2000
Año 5				

## Clase ArrayList

Las matrices no son la única forma de almacenar listas de datos relacionados:

- ArrayList es una de una serie de clases de lista.
- Tiene un juego de métodos útiles para gestionar sus elementos:
  - `add()`, `get()`, `remove()`, `indexOf()` y muchos otros
- No necesita especificar el tamaño al instanciar una ArrayList:
  - A medida que agregue más elementos, la ArrayList aumentará según sea necesario.
  - Puede especificar una capacidad inicial, pero no es obligatorio hacerlo.
- Una ArrayList solo puede almacenar objetos, no primitivos.

# Nombres de clases y sentencia de importación

- ArrayList está en el paquete `java.util`.
- Para hacer referencia a la ArrayList de su código, puede cualificar totalmente

*`java.util.ArrayList myList;`*

o bien puede agregar la sentencia de importación en la parte superior de la clase.

```
import java.util.ArrayList;
public class ArrayListExample {
    public static void main (String[] args) {
        ArrayList myList;
    }
}
```

# Trabajar con una ArrayList

```
ArrayList myList;
```

Declarar una referencia.

```
myList = new ArrayList();
```

Instanciar la ArrayList.

```
myList.add("John");  
myList.add("Ming");  
myList.add("Mary");  
myList.add("Prashant");  
myList.add("Desmond");
```

Inicializar la ArrayList.

```
myList.remove(0);  
myList.remove(myList.size()-1);  
myList.remove("Mary");
```

Modificar la ArrayList.

```
System.out.println(myList);
```