

# Bloque II. Elementos del lenguaje de programación Java



- 1. Introducción a los lenguajes de programación
- 2. Estructura de un programa
- 3. Datos y expresiones simples
- 4. Instrucciones de control
- 5. Entrada/salida simple
- 6. Arrays, secuencias y tablas

## Construcción de tablas mediante arrays



Los arrays permiten guardar muchos datos del mismo tipo

- se agrupan bajo un nombre común
- se utiliza un índice numérico para referirse al dato individual
  - en Java el índice se restringe:  $[0..N-1]$
- o varios índices en el caso de arrays multidimensionales
- los datos pueden ser variables o referencias a objetos

El tamaño es fijo: no puede cambiar

El array en Java se usa como un objeto

- se usa a través de una referencia
- se crea con **new**

# Arrays unidimensionales

Declaración:

```
tipo nombre[];
```

O también:

```
tipo[] nombre;
```

*Pseudocódigo*

```
tipo[0..tamaño-1] nombre;
```

Para crear un objeto del tipo array

```
nombre=new tipo[tamaño];
```

- El tipo es el tipo base de los elementos del array
  - tipo elemental
  - o una clase
- El nombre representa una referencia a un objeto del tipo array, que aún no existe.

# Ejemplos de creación de arrays

Declaración y creación

```
float coord[];  
coord = new float[3];
```

*Pseudocódigo*

```
real[0..2] coord;
```

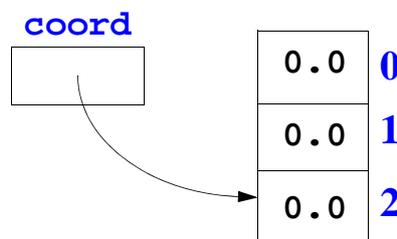
O todo junto:

```
float coord[] = new float[3];
```

O equivalentemente:

```
float[] coord = new float[3];
```

El array referenciado por **coord** tiene tres elementos enteros



- numerados **0,1, y 2.**
- al crear el array todos sus elementos se inicializan a cero

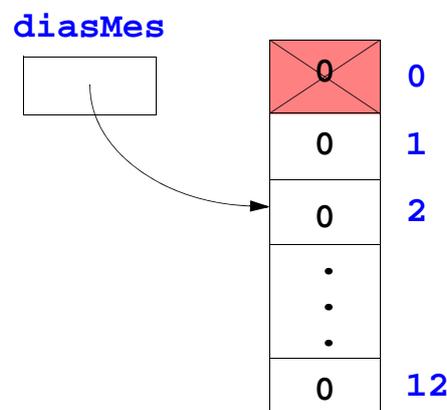
# Nota sobre el rango del índice

Si el que el array comience por cero representa un problema:

- se puede definir un elemento más, e ignorar la casilla cero

Por ejemplo

```
int[] diasMes = new int[13];
```



# Uso de arrays

Observar que un array no es una clase, sino un objeto

- no tiene operaciones
- pero sí el atributo **length**, que contiene el tamaño  
`diasMes.length // es 13`

Uso de un elemento de un array:

```
nombre[indice]
```

Ejemplos de uso

```
final int ENERO=1, FEBRERO=2; // etc
diasMes[2] = 28;
diasMes[ENERO] = 31;
System.out.println
    ("Febrero tiene "+diasMes[FEBRERO]+" dias");
```

## Uso de arrays (cont.)

Se puede escribir un literal de array en la inicialización, al crear la referencia al array:

```
int diasMes[] =
    {0,31,28,31,30,31,30,31,31,30,31,30,31};
```

Si se intenta acceder a una casilla que no existe, se produce un error

## Ejemplo de recorrido en tablas: media de una tabla de enteros

```
public class Media {
    public static void main(String[] args) {

        int nums[] = {3,4,7,8,4,5,6};
        int suma=0;
        double media;

        for (int i=0; i<nums.length; i++) {
            suma=suma+nums[i];
        }
        media=(double)suma / nums.length;
        System.out.println("La media es "+media);
        System.out.println("Longitud es "+nums.length);
    }
}
```

# Uso de arrays como parámetros

Se pueden pasar a un método parámetros del tipo array

Por ejemplo, aquí se pasa un array de enteros llamado lista

```
public void método (int[] lista) {...}
```

## Ejemplo de recorrido parcial: Máximo de unas edades de personas

```
public class Persona {
    private int edad;
    private String nombre;
    public Persona(int edad, String nombre) {
        this.edad=edad;
        this.nombre=nombre;
    }
    public int edad() {
        return edad;
    }
    public String nombre() {
        return nombre;
    }
}
```

## Ejemplo de recorrido parcial (cont.)

```
public class ListaPersonas {
    /**Calcula edad máxima de array de objetos de clase
     * Persona, recorriendo la tabla desde el segundo
     * elemento hasta el ultimo */
    public static int edadMaxima(Persona[] perso) {
        int max=perso[0].edad();
        int i=1;
        while (i<perso.length) {
            if (perso[i].edad(>max) {
                max=perso[i].edad();
            }
            i++;
        }
        return max;
    }
}
```

## Ejemplo de recorrido parcial (cont.)

```
/** programa de prueba; crea una tabla de personas
 * e invoca al metodo anterior
 */
public static void main (String[] args) {
    Persona[] lista=new Persona[5];
    lista[0]=new Persona(10,"juan");
    lista[1]=new Persona(14,"pedro");
    lista[2]=new Persona(18,"andres");
    lista[3]=new Persona(8,"ana");
    lista[4]=new Persona(13,"lucia");
    System.out.println("Edad máxima:"+
        edadMaxima(lista));
}
}
```

## Ejemplo de búsqueda

### Clase que guarda numero de un mes

- al constructor se le pasa el nombre del mes y lo busca en una tabla, guardando el número del mes (el índice)
- hay una operación para retornar el número de días del mes
- hay una operación para retornar el nombre del mes

## Ejemplo: mes

```
import fundamentos.*;
public class Mes {
    // atributos
    private int numMes;

    // constantes estaticas
    private static final int[] diasMes=
        {0,31,28,31,30,31,30,31,31,30,31,30,31};

    private static final String[] nombreMes =
        {"Error",
        "Enero", "Febrero", "Marzo",
        "Abril", "Mayo", "Junio",
        "Julio", "Agosto", "Septiembre",
        "Octubre", "Noviembre", "Diciembre"};
```

## Ejemplo (cont.)

```

/** Si encontrado, numMes= numero encontrado
 * pero si no encontrado, numMes=0 */
public Mes(String mes) {
    int m=1; boolean encontrado=false;
    numMes=0; // por si no lo encontramos
    while ((!encontrado) && m<13) {
        encontrado=
            mes.equalsIgnoreCase(nombreMes[m]);
        if (encontrado) {
            numMes=m;
        }
        m++;
    }
}

```

## Ejemplo (cont.)

```

/**
 * Retorna el numero de dias del mes
 */
public int dias() {
    return diasMes[numMes];
}

/**
 * Retorna el nombre del mes
 */
public String nombre() {
    return nombreMes[numMes];
}
}

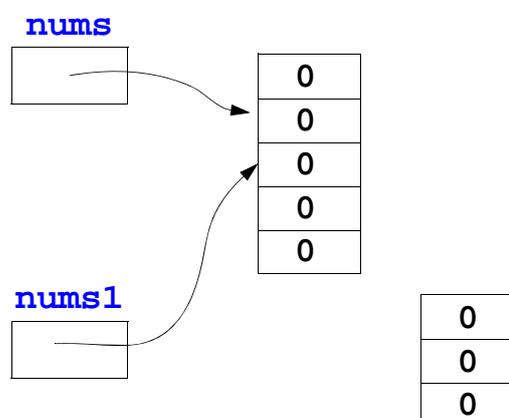
```

# Uso de un array completo

Se hace por su nombre

- Se puede cambiar la referencia al array.

```
int nums[] = new int[5];
int nums1[] = new int[3];
...
nums1 = nums;
```



Esto no copia el array

- sólo cambia la referencia

# Arrays multidimensionales

Los arrays multidimensionales son arrays de arrays, con algunas facilidades para hacer mas simple su uso.

Declaración y creación

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

El índice izquierdo representa la fila, y el derecho la columna

Los elementos se usan de la manera

```
matrizA[2][3]
```

Aunque los tamaños de cada fila pueden ser distintos, no es aconsejable (es fácil equivocarse)

# Tablas de tamaño variable

Los arrays son de tamaño fijo

- Se puede guardar una tabla de tamaño variable (pero limitado)
- Basta usar sólo la primera parte del array, dejando el resto sin usar
- Hay que llevar cuenta del número de casillas útiles

Existe una clase en Java que representa tablas de tamaño variable, cuyo tamaño puede crecer

- es la clase `ArrayList`
- está en el paquete `java.util`
- pertenece a las llamadas "*Java collections*"
- sólo se pueden almacenar objetos

# El uso de la clase `ArrayList`

Declaración y creación de una tabla cuyos elementos son objetos de la clase `Elemento`

```
ArrayList <Elemento> v= new ArrayList <Elemento> ();
```

Métodos para manejar la tabla

Cabecera	Descripción
<code>int size()</code>	Obtener el tamaño actual
<code>Elemento get(int índice)</code>	Obtener el elemento de la casilla indicada
<code>void set(int índice, Elemento e)</code>	Cambiar el elemento de la casilla indicada
<code>boolean add(Elemento e)</code>	Añadir el elemento al final de la lista

## Ejemplo con la clase ArrayList

Es el mismo ejemplo de la lista de personas

```
import java.util.ArrayList;
public class OpPersonas1{
    public static int edadMaxima
        (ArrayList<Persona> perso)
    {
        int max=perso.get(0).edad(); int i=1;
        while (i<perso.size()) {
            if (perso.get(i).edad(>max) {
                max=perso.get(i).edad();
            }
            i++;
        }
        return max;
    }
}
```

## Ejemplo con la clase ArrayList (cont.)

```
public static void main (String[] args) {
    ArrayList<Persona> lista=
        new ArrayList<Persona>();
    lista.add(new Persona(10, "juan"));
    lista.add(new Persona(14, "pedro"));
    lista.add(new Persona(18, "andres"));
    lista.add(new Persona(8, "ana"));
    lista.add(new Persona(13, "lucia"));

    System.out.println("La edad máxima es "+
        edadMaxima(lista));
}
}
```