



Universidad
Zaragoza

Fundamentos de Informática

Lección 5. Arreglos (vectores)



Universidad
Zaragoza

Curso 2010-2011

Pedro Álvarez y José Ángel Bañares

2/11/2010. Dpto. Informática e Ingeniería de Sistemas.

Índice de contenidos

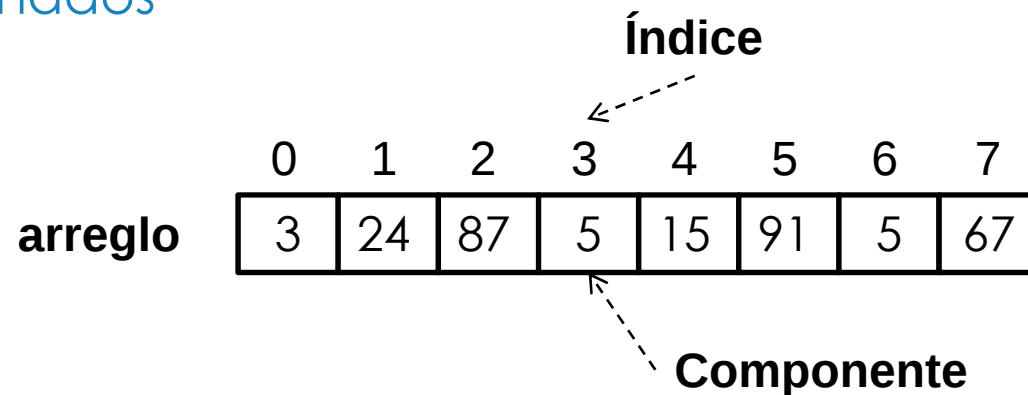
- Arreglos:
 - Concepto
 - Declaración y creación
 - Inicialización
 - Acceso
 - Asignación
- Arreglos multidimensionales
 - Concepto
 - Declaración y creación
 - Inicialización y acceso

Índice de contenidos

- **Arreglos:**
 - **Concepto**
 - **Declaración y creación**
 - **Inicialización**
 - **Acceso**
 - **Asignación**
- **Arreglos multidimensionales**
 - **Concepto**
 - **Declaración y creación**
 - **Inicialización y acceso**

Concepto de **arreglo**

- Un **arreglo** es una estructura de datos que consiste en una colección de valores del mismo tipo y semánticamente relacionados



- Características relevantes:
 - **Tamaño fijo** especificado en tiempo de compilación
 - Todos los valores de las componentes del **mismo tipo de dato**
 - **Acceso indexado** a los valores de las componentes

Declaración/creación de un arreglo

- La **declaración** consiste de:
 - Tipo de dato de las componentes
 - Número de componentes (int, byte o short positivos o char)

Sintaxis:

```
tipoDato identificador[] = new tipoDato [longitud];  
tipoDato[] identificador = new tipoDato [longitud];
```

Ejemplos:

```
int arreglo1[] = new int[12]; // Arreglo de 12 enteros  
double arreglo2[] = new double[10]; // Arreglo de 10 reales
```

- En Java, los arreglos son **tipos de datos de Referencia**
 - La declaración/creación devuelve una referencia al arreglo
 - Esto es importante y tiene implicaciones interesantes

Declaración/creación de un arreglo

- ¿Qué tipo de arreglo es declarado?

```
int arreglo[] = new int['a']; // Indexado por char
```

Declaración/creación de un arreglo

- ¿Qué tipo de arreglo es declarado?

```
int arreglo[] = new int['a']; // Indexado por char
```

- CUIDADO!!, sintácticamente correcto, pero:
 - Índice es interpretado como int (ordinal del carácter)
 - Por tanto, 97 de componentes (0..96)

```
int arreglo[] = new int[97]; // Equivalente UNICODE
```

Inicialización de un arreglo

- Inicialmente, cada componente del arreglo es inicializada con un **valor predefinido inicial**
 - **0**, para T.D. primitivos numéricos
 - **false**, para T.D. booleano
 - **primer carácter del dominio**, para T.D. char (cuidado!)
 - **null**, para T.D. de referencia
- Alternativas para inicializar un arreglo:
 - **Lista inicializadora**
 - **Instrucción FOR a nivel de componente**

Inicialización de un arreglo

■ Inicialización usando una **lista inicializadora**

Ejemplo:

```
int arreglo[] = {7, 15, -4, 12, 3, 89, 0, -31};
```

// No es necesario el operador **new** para la creación

// Implícitamente se deduce el número de componentes (8)

// Todos los **valores del mismo tipo** y del **tipo declarado** (int)

// El índice es de tipo **int** (0..length-1). Ejemplo: 0..7

■ Inicialización usando una **instrucción FOR**

Ejemplo:

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

```
for (int contador = 0; contador < arreglo.length; contador++)  
    arreglo[contador] = contador*2 + 2;
```

// Cada componente se inicializa con un entero par del 2 al 20

Acceso a los elementos de un arreglo

- Para **acceder a una componente** concreta:
 - Nombre de la referencia al arreglo
 - Índice a la componente a la que se quiere acceder ([..])

Ejemplo:

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

```
arreglo[5] = arreglo[4]*2;
```

```
arreglo[i] = (arreglo[i] % 10)*arreglo[j];
```

- Consideraciones:
 - Primer elemento siempre indexado por el **índice 0**
 - Longitud definida por la **constante length** (tamaño estático)

Acceso a los elementos de un arreglo

```
public class ExcepcionArreglos
{
    public static void main(String[] args)
    {
        int arreglo[] = new int[10];

        for (int contador=0; contador <= arreglo.length; contador++)
            arreglo[contador] = contador;
    }
}
```

¿Qué sucede al ejecutar este programa?

Acceso a los elementos de un arreglo

```
public class ExcepcionArreglos
{
    public static void main(String[] args)
    {
        int arreglo[] = new int[10];

        for (int contador=0; contador <= arreglo.length; contador++)
            arreglo[contador] = contador;
    }
}
```

¿Qué sucede al ejecutar este programa?

Exception in thread "main" **java.lang.ArrayIndexOutOfBoundsException: 10**

Asignación de arreglos

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

        int arreglo10[] = new int[10];
        int arreglo20[] = new int[20];

        for (int contador=0; contador < arreglo10.length; contador++)
            arreglo10[contador] = contador;

        arreglo20 = arreglo10;

        for (int contador=0; contador < arreglo20.length; contador++)
            System.out.println(contador + " -- " + arreglo20[contador]);

        System.out.println("Longitud = " + arreglo20.length);

    }
}
```

¿Qué sucede al ejecutar este programa?

Asignación de arreglos

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

        int arreglo10[] = new int[10];
        int arreglo20[] = new int[20];

        for (int contador=0; contador < arreglo10.length; contador++)
            arreglo10[contador] = contador;

        arreglo20 = arreglo10;

        for (int contador=0; contador < arreglo20.length; contador++)
            System.out.println(contador + " -- " + arreglo20[contador]);

        System.out.println("Longitud = " + arreglo20.length);

    }
}
```

0 -- 0
1 -- 1
2 -- 2
3 -- 3
4 -- 4
5 -- 5
6 -- 6
7 -- 7
8 -- 8
9 -- 9
Longitud = 10

¿Por qué?

Asignación de arreglos

```
public class AsignacionArreglos2
{
    public static void main(String[] args)
    {

        int arreglo10[] = new int[10];
        int arreglo20[] = new int[20];

        for (int contador=0; contador < arreglo20.length; contador++)
            arreglo20[contador] = contador;

        arreglo10 = arreglo20;

        for (int contador=0; contador < arreglo10.length; contador++)
            System.out.println(contador + " -- " + arreglo10[contador]);

        System.out.println("Longitud = " + arreglo10.length);

    }
}
```

¿Qué sucede al ejecutar este programa?

Asignación de arreglos

```
public class AsignacionArreglos2
{
    public static void main(String[] args)
    {

        int arreglo10[] = new int[10];
        int arreglo20[] = new int[20];

        for (int contador=0; contador < arreglo20.length; contador++)
            arreglo20[contador] = contador;

        arreglo10 = arreglo20;

        for (int contador=0; contador < arreglo10.length; contador++)
            System.out.println(contador + " -- " + arreglo10[contador]);

        System.out.println("Longitud = " + arreglo10.length);

    }
}
```

```
0 -- 0
1 -- 1
2 -- 2
3 -- 3
4 -- 4
5 -- 5
6 -- 6
7 -- 7
8 -- 8
9 -- 9
10 -- 10
11 -- 11
12 -- 12
13 -- 13
14 -- 14
15 -- 15
16 -- 16
17 -- 17
18 -- 18
19 -- 19
Longitud = 20
```

¿Por qué?

Asignación de arreglos

```
public class AsignacionArreglos2
{
    public static void main(String[] args)
    {
        int arreglo10[] = new int[10];
        char arreglo20[] = new char[20];

        for (int contador=0; contador < arreglo10.length; contador++)
            arreglo10[contador] = contador;

        arreglo20 = arreglo10;

        for (int contador=0; contador < arreglo20.length; contador++)
            System.out.println(contador + " -- " + arreglo20[contador]);

        System.out.println("Longitud = " + arreglo20.length);
    }
}
```

Error en compilación por T.D. incompatibles!!

Índice de contenidos

- Arreglos:
 - Concepto
 - Declaración y creación
 - Inicialización
 - Acceso
 - Asignación
- **Arreglos multidimensionales**
 - **Concepto**
 - **Declaración y creación**
 - **Inicialización y acceso**

Concepto de **arreglo multidimensional**

- Idea intuitiva de **arreglo multidimensional**:
 - Bidimensional → tabla indexada por (fila, columna)
 - Tridimensional → cubo indexado por (fila, columna, profundidad)
 - ...
- Java permite declarar “**arreglos de arreglos**”
 - CUIDADO: no es lo mismo que arreglos multidimensionales
 - Por ejemplo, podemos declarar un arreglo de 2 dimensiones con diferente número de columnas para cada fila

Declaración de un **arreglo un multidimensional**

■ Arreglos **bidimensionales**:

Sintaxis (2 dimensiones):

tipoDato identificador[][] = **new tipoDato** [long-fila][long-columna];

Ejemplos:

int arreglo1[][] = **new int**[2][4]; // 2 filas y 4 columnas por fila

int arreglo2[][] = **new int**[2][];

arreglo2[0] = **new int**[5]; // Primera fila tiene 5 columnas

arreglo2[1] = **new int**[3]; // Segunda fila tiene 3 columnas

Inicialización y acceso a un **arreglo un** **multidimensional**

- **Inicialización** basada en **lista de inicializadora** o por medio de instrucciones **for** (una por dimensión)

Ejemplo usando lista inicializadora:

```
int arreglo[][] = {{1,2,3}, {35, 6, 57}}; // 2 filas y 3 columnas por fila
```

- **Acceso** a nivel de **componente** o de **arreglo**
 - Nombre del arreglo e índices (índice) de la componente

Ejemplos de acceso:

```
int arreglo[][] = new int[2][3]; // 2 filas y 3 columnas por fila
```

```
arreglo[1][0] = arreglo[0][1]*2; // Nivel de componente
```

```
arreglo[0] = arreglo[1]; // Nivel de arreglo completo
```



Universidad
Zaragoza

