



Fundamentos de Programación 2

Ing. Marco Aedo López

Arreglos (Arrays): Búsquedas y Ordenamientos



Capítulo 1



Objetivos

- Comprender la estructura de datos **arreglo**
- Saber declarar, crear e inicializar arreglos y referirse a sus elementos
- Utilizar los arreglos para almacenar, ordenar y examinar listas y tablas de valores
- Aplicar los arreglos en técnicas de **búsqueda** y **ordenamiento** básicas
- Comprender el uso de arreglos como argumento de métodos
- Comprender los arreglos **bidimensionales**



Contenidos

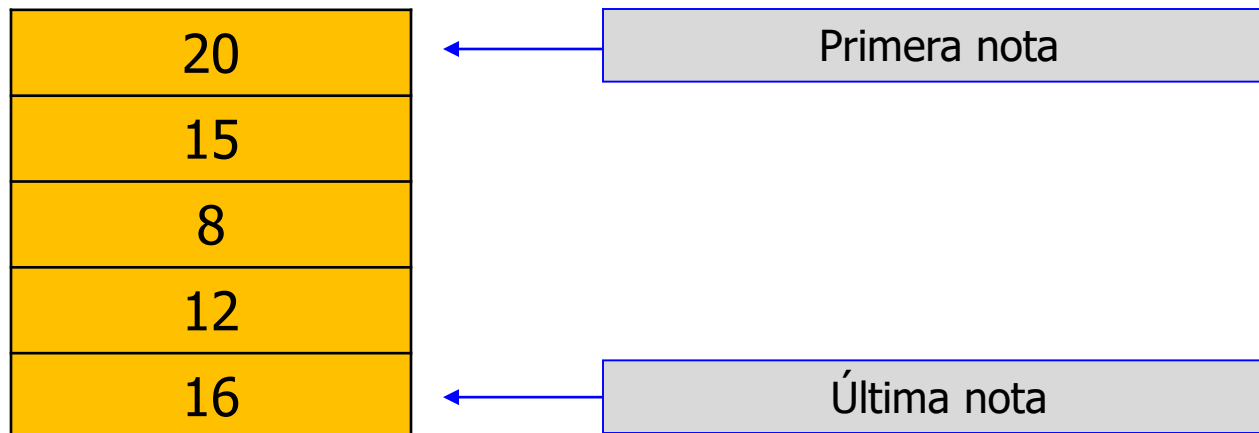
- Fundamentos de los arreglos
- Declaración y creación de arreglos
- Inicialización de elementos de arreglos
- Valores por default de elementos
- Propiedad `length`
- Arreglos de objetos
- Arreglos parcialmente llenados
- Copiando un arreglo
- Histogramas
- Búsquedas y ordenamientos
- Arreglos bidimensionales



1. Fundamentos de los Arreglos

- Un **Arreglo** es una estructura de datos que almacena un grupo de datos relacionados
- Todos los datos son del mismo tipo y tienen el mismo nombre, se diferencian por su índice
- Ejemplo: Arreglo que almacena notas del curso FP
- Cada una de las 5 cajas es llamada un **Elemento** del arreglo y cada caja almacena un valor (una nota)

notasFP



1. Fundamentos de los Arreglos

- Un arreglo usa corchetes [] alrededor del índice para acceder a sus elementos específicos
- El número de índice empieza en **0** en lugar de 1, y el índice del último valor es igual a la **longitud - 1**

notasFP		¿Cómo acceder a cada elemento?	
20		notasFP[0]	} ← 5 elementos
15		notasFP[1]	
8		notasFP[2]	
12		notasFP[3]	
16		notasFP[4]	



1. Fundamentos de los Arreglos

- Se puede cambiar la segunda nota del arreglo:

```
notasFP[1] = 11;
```

- Ahora quiero imprimir la primera nota del arreglo:

```
System.out.println(notasFP[0]);
```



EJEMPLO

- Hacer un programa que cree un arreglo para n notas enteras, luego que permita ingresarlas y finalmente que las imprima


```
import java.util.Scanner;

public class Fundamentos2 {
    public static void main(String[] args){

        Scanner scan = new Scanner(System.in);
        int[ ] notasFP;
        int tam;
        int nota;

        System.out.print("Cuantas notas desea ingresar?");
        tam = scan.nextInt();
        notasFP = new int[tam];
        for (int i=0; i<tam; i++){
            System.out.print("Ingresa nota: ");
            nota = scan.nextInt();
            notasFP[i] = nota;
        }
        System.out.println("Lista ingresada:");
        for (int i=0; i<tam; i++)
            System.out.println((i + 1) + ". " + notasFP[i]);
    }
}
```



EJERCICIO 1

- Hacer un programa que cree un arreglo para n notas enteras, luego que permita ingresarlas y finalmente que las imprima. Usando métodos.



2. Declaración de Arreglos

- Un arreglo es una variable, por lo que debemos declararla previamente
- Sintaxis:

<tipoElementos> [] <nombreArreglo>;

- Ejemplos:

```
int[] notasFP;  
double[] horasTrabajadas;  
String[] nombres;
```



3. Creación de Arreglos

- En Java los arreglos son objetos
- Luego, como todo objeto, un arreglo debe ser creado/instanciado con el operador `new`
- Sintaxis:

`<nombreArreglo> = new <tipoElementos> [<tamaño>] ;`

`<tipoElementos>` indica el tipo de cada elemento en el arreglo

`<tamaño>` indica el número de elementos en el arreglo

- Ejemplo:

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

Creación de
arreglo



3. Creación de Arreglos

- Se puede combinar la declaración con la creación de arreglos
- Sintaxis:

`<tipoElementos> [] <nombreArreglo> = new <tipoElementos> [<tamaño>];`

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

- En una simple sentencia se declara e instancia/crea un arreglo de 10 elementos enteros



3. Creación de Arreglos

Declaración:

```
<tipoElem> [ ] <nombreArreglo>; //variación 1
```

```
<tipoElem> <nombreArreglo> [ ]; //variación 2
```

Creación:

```
<nombreArreglo> = new <tipoElem> [<tamaño>];
```

Declaración y Creación:

```
<tipoElem> [ ] <nombreArreglo> = new <tipoElem> [<tamaño>];
```

Variación 1

```
double [ ] notas;  
notas = new double[12];
```

Variación 2

```
double notas [ ];  
notas = new double[12];
```

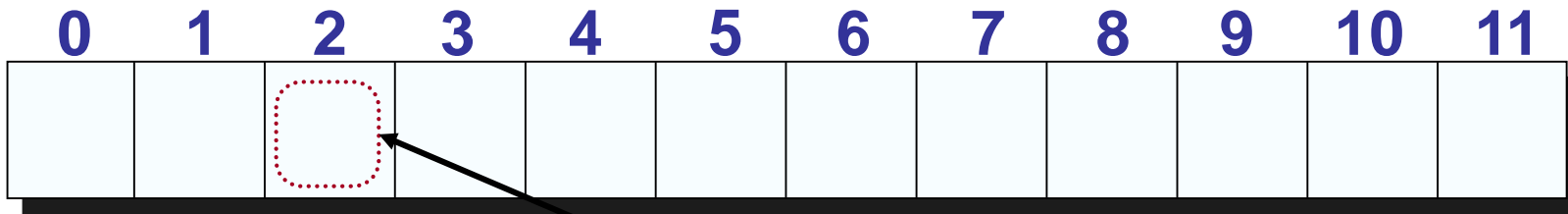


Un array es un objeto!

3. Creación de Arreglos

```
double[] notas = new double[12];
```

notas



El índice en la primera posición del arreglo es **0** y va hasta **tamaño-1**

notas[2]

El índice se refiere al elemento en la posición 2



4. Inicialización de Arreglos

- Se puede inicializar los valores usando una lista inicializadora
- Sintaxis:
$$\langle \text{tipoElementos} \rangle [] \langle \text{nombreArreglo} \rangle = \{ \langle \text{listaValoresElementos} \rangle \};$$
- Ejemplo:

```
String[] estudiantes= {"Juan", "Adriana", "Carla"};
```
- Cuando inicializamos así, el tamaño del arreglo es igual al número de elementos de la lista inicializadora
- Notar que no se usa el operador `new`



4. Inicialización de Arreglos: Valores por default de un arreglo

- En un arreglo se le da los valores por default a cada elemento según su tipo:

Tipo Elemento	Valor por default
Entero	0
Punto Flotante	0.0
Booleano	false
Referencia	null



5. Características de los Arreglos en Java

- El compilador reserva la cantidad de **memoria** apropiada
- Los índices son denotados como expresiones en corchetes: **[]**
- El **tipo base** del arreglo puede ser cualquier tipo: tipos de dato primitivos, referencias (objetos) o definidos por el usuario
- El **tamaño** del arreglo puede ser especificado en tiempo de ejecución
- El tipo del índice es entero, y el rango de los índices va desde **0 ... n-1**, en donde *n* es el número de elementos del arreglo
- El **control** de no exceder los índices lo realiza el programador
- Atributo **length** especifica el numero de elementos del arreglo
- El arreglo es un objeto
 - Presenta las mismas características de todos los objetos

Ejemplo (1/2)

■ Las definiciones:

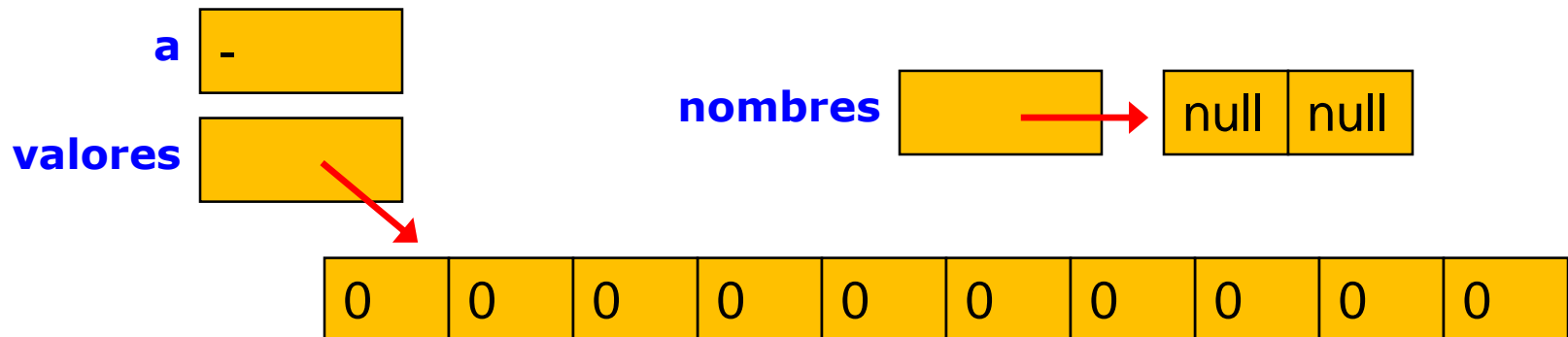
```
char[] a;
```

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

```
String[] nombres = new String[2];
```

■ Producen:

- Objeto arreglo **a** declarado pero no creado, ni inicializado
- Objeto arreglo **valores** referencia a 10 nuevos elementos de tipo entero. Por defecto cada elemento se inicializa con 0
- Objeto arreglo **nombres** referencia 2 elementos tipo String. Cada elemento es inicializado por defecto con **null**



- Por simplicidad, algunas veces representamos



como:

[illegible]



6. Atributo `length`

- Imagine el siguiente arreglo de cadenas:

```
String[] colors = {"blue", "gray", "lime", "teal", "yellow"};
```

- Lo imprimimos usando el atributo `length`

```
for (int i=0; i<colors.length; i++)  
    System.out.println(colors[i]);
```

Atributo `length` es el tamaño o cantidad de elementos del arreglo

- `nombreArreglo.length`
- Recordar:
 - `length()` es un método de `String`
 - `length` es un atributo de Arreglo



EJERCICIO 2

- Hacer un programa que cree un arreglo con capacidad para 100 números de celulares (que sólo acepte números de 9 dígitos), que permita ingresar los números requeridos hasta que el usuario ingrese una “q” y finalmente que imprima sólo los números ingresados. Confiar que el usuario siempre ingresa dígitos.



EJERCICIO 3

- Hacer un programa que cree un arreglo con capacidad para 100 números de celulares (que sólo acepte números de 9 dígitos), que permita ingresar los números requeridos hasta que el usuario ingrese una “q” y finalmente que imprima sólo los números ingresados. Verificar que los números de celular sólo contengan dígitos. Utilizar:

`char charAt(int index)` (método de `String`)

`boolean isDigit(char ch)` (método de `Character`)