Apache NetBeans

#27 ARREGLOS

CONCEPTOS

Java

PROGRAMA FÁCIL CON JAVA

Fundamentos de programación

Unidad 4: Organización de datos

Competencia específica a desarrollar:

Conoce y aplica estructuras de datos en un lenguaje de programación que permita la organización de datos en la resolución de problemas reales..

Unidad	Tema	Subtemas
4	Organización de datos	4.1 Arreglos 4.2 Unidimensionales: conceptos básicos, operaciones y aplicaciones. 4.3 Multidimensionales: conceptos básicos, operaciones y aplicaciones. 4.3 Estructuras o registros.

Matriz

Es una estructura homogénea compuesta por varios elementos, todos del mismo tipo, y almacenados consecutivamente en memoria.

Las matrices se pueden clasificar según su

- dimensión: unidimensionales y multidimensionales

- contenido: numéricas, de caracteres y de referencias a objetos.

Cada elemento puede ser accedido directamente por el nombre de la variable matriz seguido del subíndice entre corchetes.

edad[0] = 18

En el caso de una matriz bidimensional, se accede con 2 subíndices (fila, columna) entre corchetes. **edades[0][0] = 18**

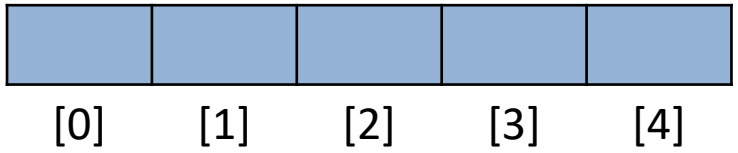
Matriz

Desde un punto de vista matemático,

en el caso de un subíndice:

v = [a0, a1, a2, …, ai, …, an]

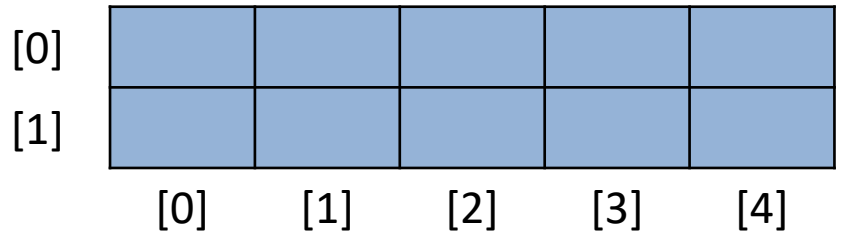
Ejemplo: **Calificaciones[5]**



o bien si se utilizan 2 subíndices:

$$m = \begin{pmatrix} a00, a01, a02, \ldots, a0j, \ldots, a0n \\ a10, a11, a12, \ldots, a1j, \ldots, a1n \\ \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\ ai0, ai1, ai2, \ldots, \quad aij, \ldots, ain \end{pmatrix}$$

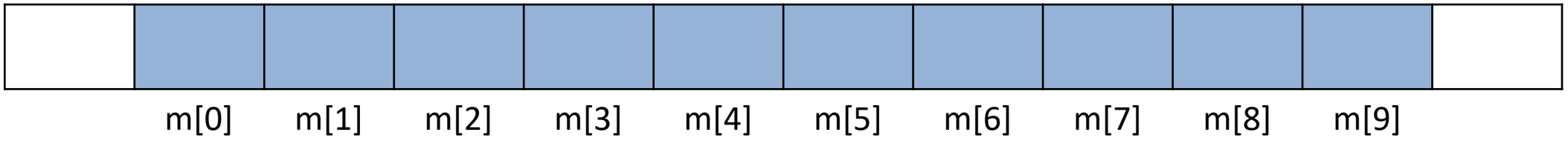
Ejemplo: **Calificaciones[2][5]**



Esta matriz se representa con los subíndices de fila y columna.

Matriz

Considera una matriz unidimensional de enteros, llamada m, la cual contiene 10 elementos.



El tamaño de la matriz es n

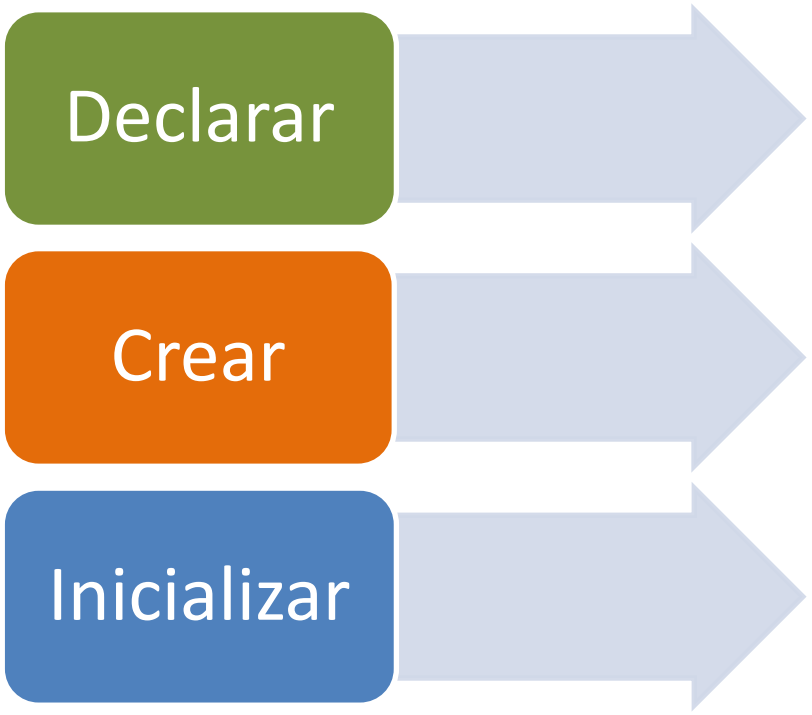
El primer subíndice empieza en 0

El último subíndice termina en n-1

Los subíndices son enteros consecutivos

Un subíndice puede ser cualquier expresión entera positiva

Acciones para utilizar una Matriz



int [] numeros; **o** **int numeros [];**

numeros = new int [4]

numeros [0] = 2
numeros [1] = 4
numeros [2] = 20
numeros [3] = 25

Lo mejor es leer los valores con un ciclo.

Matrices numéricas unidimensionales

Declarar

tipo [] *nombre;* *tipo nombre* [];

Ejemplos

int [] numeros;	int numeros [];
int [] lista;	int lista [];
float [] precio;	float precio [];
float [] b;	float b [];
int [] vector;	int vector [];
CClase [] x;	CClase x [];

El tamaño será especificado cuando se cree la matriz, operación que se hará durante la ejecución del programa.

Matrices numéricas unidimensionales

Crear

Reservar la cantidad de memoria necesaria para contener todos sus elementos y asignar al nombre de esa matriz una referencia a ese bloque.

nombre = new *tipo*[*tamaño*];

expresión entera positiva

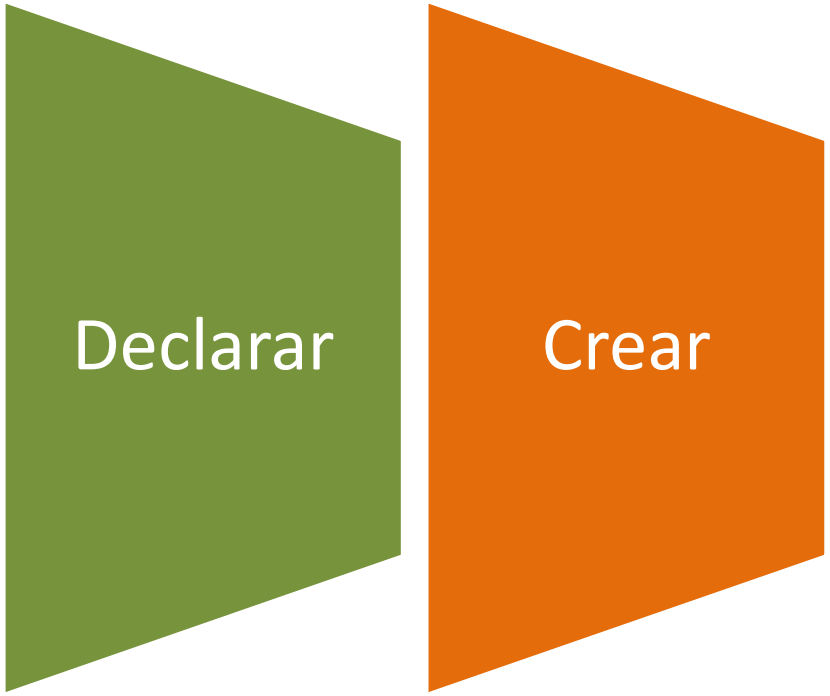
Ejemplos

numeros = new int[10];
lista = new int[t];
precio = new float[15];
b = new float[20+i];
vector = new int[12];
x = new CClase[25];

Declaración y creación en un paso

tipo[] *nombre* = *new tipo*[*tamaño*];

tipo nombre[] = *new tipo*[*tamaño*];



Ejemplos

```
int [] numeros = new int[10];
int lista [] = new int[55];
float [] precio = new float[15];
float b [] = new float[20];
int [] vector = new int[12];
CClase [] x = new CClase[25];
```

Formas de inicialización:

Inicializar

- Por default:
  - 0 si el arreglo es numérico
  - ‘\u0000’ si el arreglo es de caracteres
  - false si el arreglo es booleano
  - null si el arreglo es de objetos
- En tiempo de programación:

```
int [] edades  = { 20, 19, 21, 18, 22, 17 };
float [] precio = { 25.50F, 15.35F, 205.4F, 22.5F };
```

- En tiempo de ejecución:

```
for ( i = 0; i < 10; i++ )
        edades[i] = Leer.datoInt( );
```

Límites: Las posiciones en el arreglo van de **0** a **n – 1**, donde ***n*** es el tamaño del arreglo.

Fuera de los límites, Java lanza una *excepción* del tipo:

ArrayIndexOutOfBoundsException

Longitud: La propiedad ***length*** contiene el tamaño del arreglo.

Mostrar: Sentencia para visualizar los elementos de un arreglo:
System.out.println(Arrays.toString(nums));

Ejemplo:

```
int [] vector = new int[5];
for ( i = 0; i < vector.length; i++ )
        vector[i] = Leer.datoInt( );
System.out.println(Arrays.toString(vector));
```

Ejemplo 4.1

Construye el programa en Java que, en tiempo de programación, cree un arreglo con 5 edades, 4 gastos y 3 nombres. Que sume el total de los gastos, y después que muestre los elementos de cada arreglo.

```
import java.util.Arrays;

public class Ejemplo41 {

    public static void main(String[] args) {
        int edades[]  = {17, 18, 19, 20, 21};
        float []gasto = {5.25F, 10.5F, 15.75F, 20.0F };
        String nombre[] = {"Maty", "Mau", "Jana" };
        float suma = gasto[0]+gasto[1]+gasto[2]+gasto[3];
        System.out.println(edades[2]);
        System.out.println("Total de gastos $"+suma);
        System.out.println(Arrays.toString(edades));
        System.out.println(Arrays.toString(gasto));
        System.out.println(Arrays.toString(nombre));
    }
}
```

Ejemplo 4.2

Construye el programa en Java que cree un arreglo para guardar 5 edades leídas del teclado. Después mostrar las edades mediante un ciclo.

```
import java.util.Scanner;

public class Ejemplo42 {

    public static void main(String[] args) {
        Scanner Leer = new Scanner(System.in);
        int edades[] = new int[5];
        System.out.print("Introduce la edad 1: ");
        edades[0] = Leer.nextInt();
        System.out.print("Introduce la edad 2: ");
        edades[1] = Leer.nextInt();
        System.out.print("Introduce la edad 3: ");
        edades[2] = Leer.nextInt();

        for (int i = 0; i < edades.length; i++) {
            System.out.println("Edad "+i+": "+edades[i]);
        }
    }
}
```

Ejemplo 4.3

Construye el programa en Java que mediante un ciclo guarde en un arreglo N gastos leídos del teclado. Después mostrar cada uno de los gastos y la suma de todos los gastos.

```
import java.util.Scanner;

public class Ejemplo43 {

    public static void main(String[] args) {
        Scanner Leer = new Scanner(System.in);
        System.out.print("¿Cuántos gastos son? ");
        int N = Leer.nextInt();
        float gastos[] = new float[N], suma=0;
        for (int i = 0; i < gastos.length; i++) {
            System.out.print("Introduce el gasto "+i+": ");
            gastos[i] = Leer.nextFloat();
        }
        for (int i = 0; i < gastos.length; i++) {
            System.out.println("Gasto "+i+": "+gastos[i]);
            suma = suma + gastos[i];
        }
        System.out.println("Total de gastos $"+suma);
    }
}
```