

Programación

Tema 3



Estructuras de almacenamiento

Tema 3

Tema 3

1. Declaración de arrays
2. Creación de arrays
3. Acceso a los elementos de un array
4. Manipulación de vectores
5. Manipulación de matrices

Arrays

- Def :
- Un array es una estructura de datos que contiene una colección de datos del **mismo tipo**
 - **Ejemplos:**
 - Temperaturas mínimas de los últimos 30 días
 - Valor de las acciones de una empresa durante la ultima semana
- Terminología
- Los arrays de una dimensión se conocen con el nombre de **vectores** 
- Los arrays de dos dimensionales se conocen con el nombre de **matrices** 

Arrays

- Propiedades
- Los arrays se utilizan como **contenedores** para almacenar datos relacionados
 - En vez de declarar variables por separado para cada uno de los elementos del array
- Todos los **datos** incluidos en el array son **del mismo tipo**.
 - Se pueden declarar arrays de tipo int o tipo float , pero dentro del mismo arrays no se pueden mezclar datos de tipo int con float
- El tamaño de un array se establece cuando se crea el array con el operador **new**
- A los elementos del array se accede a través de la posición que ocupan dentro del conjunto de elementos del array

Declaración

- Se utilizan corchetes para indicar que se trata de un array y no de una simple variable del tipo especificado

Declaración de un array unidimensional (vector)

TIPO IDENTIFICADOR[];

TIPO[] IDENTIFICADOR;

Declaración de un array bidimensional (matriz)

TIPO IDENTIFICADOR[][]; 

TIPO[][] IDENTIFICADOR;



TIPO : es el tipo de dato de los elementos del vector

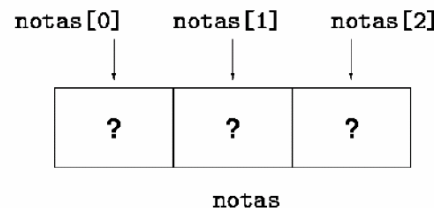
IDENTIFICADOR : es el identificador de la variable

Creación de arrays

- Los arrays se crean con el operador **new**
- Vectores
 - `vector = new tipo[num_elementos];`
 - `float [] notas = new float [ALUMNOS]`
 - `Int[] temperaturas = new int[7];`
- Matrices
 - `matriz = new tipo[num_filas][num_columnas]`
 - `int[][] temperaturas = new int[12][5];`

Acceso a los datos de un array

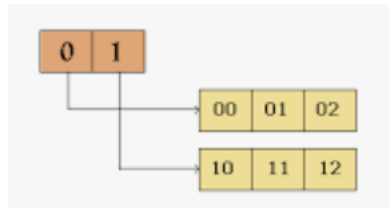
- Para acceder a los elementos de un array, utilizaremos los índices que indican su posición dentro del array
- Vectores
 - `vector[índice]`
 - En java, el índice del primer elemento de un vector es siempre 0
 - El tamaño de un array puede obtenerse utilizando la propiedad `vector.length`
 - Por tanto, el índice de la última componente es `vector.length-1`
 - Ejemplo
 - `float[] notas = new float[3]`



Acceso a los datos de un array

- Matriz

- `matriz[índice_fila][índice_columna];`
- Una matriz es un vector de vectores
- En java, el índice de la primera componente es siempre 0, por lo que `matriz[0][0]` será el primer elemento de la matriz.
- El tamaño del array puede obtenerse utilizando la propiedad `array.length`
 - `matriz.length` : nos dará el número de filas
 - `Matriz[0].length` : nos dará el número de columnas



Inicialización en la declaración

- Podemos asignar un valor inicial a los elementos del array en la propia declaración
 - `int vector[] = {1,2,3,4,5};`
 - `int matriz[][] = { {1,2,3},{4,5,6} };`
- El compilador deduce automáticamente las dimensiones del array

Manipulación de vectores y matrices

- Las operaciones se realizan componente a componente
- Ejemplo : suma de los elementos de un vector

```
int    i;  
int    n = datos.length;  
float  suma = 0;  
  
for (i=0; i<n; i++)  
    suma = suma + datos[i];
```

- No es necesario utilizar todos los elementos de un vector, por lo que, al trabajar con ellos, se puede utilizar una variable entera adicional que nos indique el número de datos que realmente estamos utilizando

Manipulación de vectores y matrices

- El tamaño del vector nos dice cuanta memoria se ha reservado para almacenar datos del mismo tipo, no cuantos datos del mismo tipo tenemos realmente en el vector.
- Ejemplo: suma de los n primeros elementos de un vector

```
int    i;  
float suma = 0;  
  
for (i=0; i<n; i++)  
    suma = suma + datos[i];
```

Manipulación de vectores y matrices

- Cuando se pasa un array como parámetro, se copia una referencia al array y no al conjunto de valores en sí.
- Por tanto, tenemos que tener cuidado con los efectos colaterales que se producen si, dentro de un módulo modificamos un vector que recibimos como parámetro

Biblioteca de Array

- La biblioteca de clases de Java incluye una clase auxiliar llamada **java.útil.Arrays** que incluye como métodos algunas de las tareas que se realizan más a menudo con vectores
- `Arrays.sort(v)` : ordena los elementos de un vector
- `Arrays.equals(v1,v2)` comprueba si dos vectores son iguales
- `Arrays.fill(v,val)` rellena el vector `v` con el valor de `val`.
- `Arrays.toString(v)` devuelve una cadena que representa el contenido del vector
- `Arrays.binarySearch(v,k)` busca el valor de `k` dentro del vector `v`, **que previamente debe estar ordenado**