

14 | Arreglos y Matrices

Un mecanismo natural para representar matrices matemáticas en la computadora es con arreglos de dos dimensiones. En esta práctica harás uso de esa técnica.

META

Que el alumno aprenda a utilizar arreglos como estructura para almacenar, recuperar y manipular información de objetos.

OBJETIVOS

Al finalizar la práctica el alumno será capaz de:

- Crear arreglos de dos dimensiones.
- Recorrer los arreglos, visitando elemento por elemento, para creación y manipulación de datos.
- Identificar el tamaño de los arreglos en todas sus dimensiones.
- Poner en práctica sus conocimientos sobre el uso de ciclos para realizar operaciones con arreglos.
- Utilizar arreglos como atributos de objetos.

ANTECEDENTES

Arreglos 2D

Para representar arreglos de más dimensiones, Java utiliza arreglos de arreglos. Es decir, arreglos de referencias a arreglos con una dimensión menos. Así los arreglos 2D son arreglos de arreglos 1D. Para crearlos sólo hay que agregar más corchetes y

especificar más tamaños. Para los arreglos 2D especificaremos primero el número de renglones y luego el número de columnas.

```
1 // Arreglo con cinco renglones y tres columnas.
2 double números = new double[5][3];
```

Accederemos a cada posición indicando el renglón y la columna que nos interesan.

```
1 // Accede al dato en el renglón cero, columna 1.
2 double unDoble = números[0][1];
```

Arreglos como atributos

Podemos utilizar variables tipo arreglo del mismo modo que cualquier otra variable.

Listing 14.1: CajitaRectangular.java

```
1 public class CajitaRectangular {
2     int números[][];
3
4     public Cajita(int renglones, int columnas) {
5         números = new int[renglones][columnas];
6     }
7
8     public int getRenglones() {
9         return números.length;
10    }
11
12    // Cuidado con este.
13    public int getColumnas() {
14        if (números.length > 0) {
15            return números[0].length
16        } else {
17            return 0;
18        }
19    }
20 }
```

EJERCICIOS

1. Crea una clase llamada Matriz2D. Como atributo tendrá un arreglo 2D de tipo double. En el constructor se deberá indicar el número de renglones y columnas que tendrá la matriz.

2. Sobreescribe el método `toString` para que imprima en pantalla el contenido de la matriz.
3. Agrega dos métodos: uno que devuelva el número de renglones y otro, el número de columnas.
4. Crea un método estático que reciba un arreglo como parámetro con los datos que se le asignarán a la matriz y devuelva una `Matriz2D` con esos datos. Ojo: debes revisar que tenga las dimensiones correctas para cada renglón.
5. Prueba tu código creando e imprimiendo algunas matrices.
6. Implementa las siguientes operaciones de matrices (recuerda probar cada uno de ellos en una clase de uso):
 - a) Suma de matrices: recibe como parámetro otro objeto tipo `Matriz2D` y devuelve una matriz nueva con el resultado.
TIP: Los atributos privados de otros objetos de la misma clase son accesibles directamente con el operador ..
 - b) Resta de matrices: recibe como parámetro otro objeto tipo `Matriz2D` y devuelve una matriz nueva con el resultado.
 - c) Multiplicar una matriz por un escalar: recibe como parámetro un `double` y devuelve una matriz nueva con el resultado.
 - d) Multiplicación de matrices: recibe como parámetro otra `Matriz2D` y devuelve una matriz nueva con el resultado.