



Laboratorio de Estructuras de Datos

Práctica 7. Arreglos multidimensionales

Unidad Temática: 1. Introducción a las estructuras de datos y estructuras fundamentales

📅 👤 Profesor: Dr. Aldonso Becerra Sánchez

Índice

1	Objetivo de la tarea	1
2	Tiempo aproximado de realización	1
3	Fecha de entrega	1
4	Fecha de entrega con extensión y penalización	1
5	Introducción	1
6	Actividades a realizar	1
6.1	Actividad Inicial	1
6.2	Actividad 1	1
6.3	Actividad 2	1
6.4	Actividad 3	2
6.5	Actividad 4	2
6.6	Actividad 5	2
6.7	Actividad 6	2
6.8	Actividad 7	2
7	Contáctame	2
	References	2

requiere conocer a priori la cantidad de elementos a guardar [1], [2], [3].

6. Actividades a realizar

6.1. Actividad Inicial

Lea primero toda la práctica ⚠️. No inicie a programar sin leer todo cuidadosamente primero. Recuerde que debe generar el reporte en formato IDC con todos sus componentes.

6.2. Actividad 1

Primero genere la Introducción 📄.

6.3. Actividad 2

Información importante

Esta actividad debe entrar en la parte de *Desarrollo* 📄.

Defina un TDA llamado “ArregloListaInfoEstatica2”, que permita realizar las siguientes operaciones:

- Constructor: ArregloListaInfoEstatica2 (int filas, int cols).
- Constructor: ArregloListaInfoEstatica2 (int filas, int cols, Object valor). El valor es el contenido con que se debe inicializar toda la matriz.
- Leer un elemento de la matriz: Object dato(int indiceFila, int indiceCol).
- Cambiar un elemento: boolean cambiar(int indiceFila, int indiceCol, Object valor).
- Obtener el número de elementos en renglones y columnas.
- Imprimir los datos de la matriz: void imprimirXRenglones(). Imprime la matriz en formato renglón por renglón.
- Imprimir los datos de la matriz: void imprimirXColumnas(). Imprime la matriz en formato columna por columna.
- Hacer la transpuesta de la matriz: void transpuesta().
- Inicializar todos los elementos de la matriz: void llenar(Object valor).
- Generar y regresar una “copia” de la matriz: ArregloListaInfoEstatica2 copiar().

1. Objetivo de la tarea

Comprender el uso de los arreglos multidimensionales como TDA.

2. Tiempo aproximado de realización

📅 ⌚ 5 horas.

3. Fecha de entrega

📅 10 de septiembre de 2024.

4. Fecha de entrega con extensión y penalización

📅 11 de septiembre agosto de 2024.

5. Introducción

La facilidad que los arreglos multidimensionales tienen para permitir guardar datos en forma de columnas/renglones, los hace pertinentes para la resolución de muchos problemas donde se requiere esta situación. El único detalle con esta cuestión es que es poco flexible el número de elementos que podemos manipular, ya que se

11. Generar un método que diga si el contenido de dos matrices es igual: boolean esIgual(ArregloListaInfoEstatica2 matriz2). Posición por posición como contenido.
12. Defina un método que genere un vector columna con los datos especificados por el usuario, a partir de la matriz vacía: boolean vectorColumna(int numFilas, Object valor).
13. Defina un método que genere un vector renglón con los datos especificados por el usuario, a partir de la matriz vacía: boolean vectorRenglon(int numColumnas, Object valor).
14. Defina un método que permita crear/redimensionar/substituir la tabla actual por una pasada como argumento. boolean redefinir(ArregloListaInfoEstatica2 matriz2).
15. Defina un método que permita agregar un ArregloListaInfoEstatica pasada como argumento como renglón de la matriz existente. Para esto deberá validar los tamaños del arreglo que cumplan con el cometido de la matriz. boolean agregarRenglon(ArregloListaInfoEstatica arreglo).
16. Defina un método que permita agregar una ArregloListaInfoEstatica pasada como argumento como columna de la matriz existente. Para esto deberá validar los tamaños del arreglo que cumplan con el cometido de la matriz. boolean agregarColumna(ArregloListaInfoEstatica arreglo).
17. Defina un método que permita agregar una matriz nueva a la matriz actual. Para esto deberá validar que sea como elementos agregados en columnas (hacia la derecha). boolean agregarMatrizXColumna(ArregloListaInfoEstatica2 matriz2).
18. Defina un método que permita agregar una matriz nueva a la matriz actual. Para esto deberá validar que sea como elementos agregados en filas (hacia abajo). boolean agregarMatrizXRenglon(ArregloListaInfoEstatica2 matriz2).
19. Defina un método que permita pasar una matriz 2D a matriz 3D. Para eso deberá llevar la matriz 2D actual más un argumento que es un arreglo de matrices. public ArregloListaInfoEstatica3 aMatriz3(ArregloListaInfoEstatica matrices). La variable matrices contiene las tablas (la tercera dimensión, profundidad) que permitirán agregarse a la matriz actual 2d y generar una matriz 3D, la cual deberá ser regresada. No olvide validar las dimensiones.
20. Pasar una matriz 2D a vector columna. ArregloListaInfoEstatica2 aVectorColumna(). Este método convierte la matriz actual acomodando cada columna una debajo de otra para formar un vector columna.
21. Pasar una matriz 2D a vector renglón. ArregloListaInfoEstatica2 aVectorRenglon(). Este método convierte la matriz actual acomodando cada renglón uno en seguida del otro (hacia la derecha) para formar un vector renglón.
22. Eliminar una columna de una tabla boolean eliminarColumna(TipoColumna tipoCol). TipoColumna es un enumerado para columna izquierda o derecha.
23. Eliminar una fila de una tabla boolean eliminarRenglon(TipoRenglon tipoRenglon). TipoRenglon es un enumerado para fila superior o inferior.
24. Eliminar una fila de una posición de una matriz boolean eliminarRenglon(int renglon). “renglon” indica la fila a eliminar.
25. Eliminar una columna de una posición de una matriz boolean eliminarColumna(int columna). “columna” indica la columna a eliminar.


6.4. Actividad 3

Pruebe el funcionamiento del programa de la actividad 2 con todo y sus capturas de pantalla.


6.5. Actividad 4

Realice la sección de Código agregado  (diagrama de clases UML).




6.6. Actividad 5

Realice la sección de Pre-evaluación  (use los lineamientos establecidos).

6.7. Actividad 6

Finalmente haga las Conclusiones .

6.8. Actividad 7


Subir los entregables (pdf  y zip  con código )  a Moodle.


7. Contáctame

Puedes contactarme a través de los siguientes medios.

 <https://moodle.ingsoftware.uaz.edu.mx/>

 a7donso@gmail.com

 Cubículo

 Salón CC2-IS

Referencias

- [1] O. Cairo y S. Guardati, *Estructura de datos*. McGraw-Hill.
- [2] L. Joyanes Aguilar, *Fundamentos de programación, algoritmos u estructura de datos*. McGraw-Hill.
- [3] M. A. Weiss, *Estructura de datos en Java*. Addison Wesley.