**CURSO:**

**Estructura de Información**

**Trabajo Académico:**

**Suma y Multiplicación de Matrices Cuasi-vacías**

**Alumno:**

**Raúl Marcelo Armas Calderón**

**Código:**

**290078898**

**ÍNDICE**

1.- Introducción …………................................................................ 3

2.- Métodos …………................................................................ 3

3.- Resultados …………................................................................ 5

4.- Conclusiones …………................................................................ 7

5.- Bibliografía …………................................................................ 7

1.- Introducción

El presente trabajo presenta la implementación de las operaciones de suma y multiplicación de matrices cuasi-vacías aplicando el paradigma de la programación orientada a objetos.

La aplicación permite ingresar los datos de dos matrices, las cuales están restringidas a matrices de tamaño máximo 10 x 10 (100 celdas).

Las matrices son representadas por una estructura abstracta de datos (clase) en donde se almacenan el tamaño de la matriz (número de líneas y número de columnas) y las posiciones de las celdas que tienen valores diferentes a cero con sus respectivos valores.

Estas estructuras de datos abstractas se almacenan en filas (archivos). Una fila almacena la primera matriz y otra fila almacena la segunda matriz.

Las operaciones implementadas son la suma y multiplicación de matrices y las matrices resultado de estas operaciones también se almacenan en otra fila.

Se han considerado algunas consistencias en la realización de las operaciones, tales como, validar que el tamaño de las dos matrices, que operan en la operación de suma, sean del mismo tamaño (igual número de líneas e igual número de columnas) y que las dos matrices que operan en la multiplicación cumplan la condición de que el número de columnas de la primera matriz sea igual al número de líneas de la segunda matriz.

A continuación se describen los métodos utilizados y posteriormente se muestran los resultados.

2.- Métodos

Se ha aplicado el paradigma de la programación orientada a objetos

Se define la clase MATRIZ y se muestran sus atributos:

class MATRIZ

int QtyLineas;

int QtyColumnas;

int celdas[100][2];

Las variables:

QtyLineas: contiene la cantidad de líneas de la matriz

QtyColumnas: contiene la cantidad de columnas de la matriz

celdas[100][2]: contiene el par, posición de la celda y valor de la celda; y ha sido dimensionada para 100 celdas considerando un tamaño máximo de 10 líneas por 10 columnas.

Los métodos implementados son:

**void IngresarMatriz()**

Permite el ingreso de los datos de las variables de un objeto de la clase MATRIZ

**void MostrarMatriz()**

Muestra los datos de las variables de un objeto de la clase MATRIZ

**void REGISTRAR(char fila[],MATRIZ x)**

Almacena en una fila (archivo) los datos de las variables (estructura de datos abstracta) de un objeto de la clase MATRIZ.

**void LEERFILA(char fila[],MATRIZ y)**

Lee una fila que contiene los datos que corresponden a las variables de un objeto de la clase MATRIZ y los almacena en dichas variables del objeto.

**void LEER(MATRIZ y, int my[10][10], int &ly, int &cy)**

Lee los datos de las variables de un objeto de la clase MATRIZ y los convierte a una representación de la matriz en forma de arreglo bidimensional.

**void ESCRIBIR(MATRIZ y, int my[10][10], int ly, int cy)**

Actualiza los datos de las variables de un objeto de la clase MATRIZ con los datos de una matriz representada en forma de un arreglo bidimensional.

**void VISUALIZAR(char fila[],MATRIZ y)**

Lee una fila que contiene los datos que corresponden a las variables de un objeto de la clase MATRIZ y los almacena en las variables del objeto, luego muestra los datos de las variables del objeto de la clase MATRIZ.

3.- Resultados

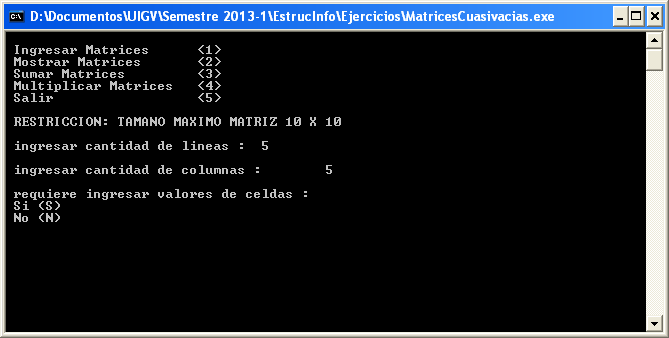
INGRESAR MATRICES:

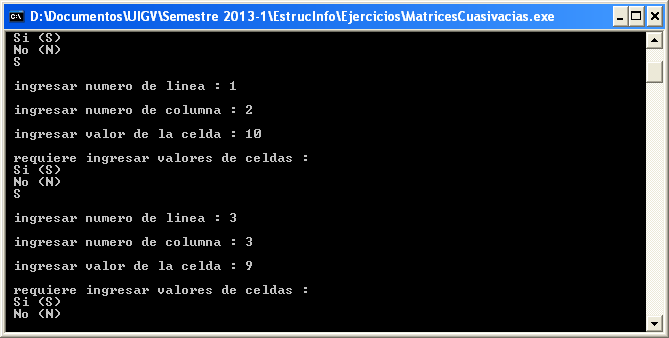
Requiere el ingreso de:

a.- Cantidad de líneas de la matriz.

b.- Cantidad de columnas de la matriz

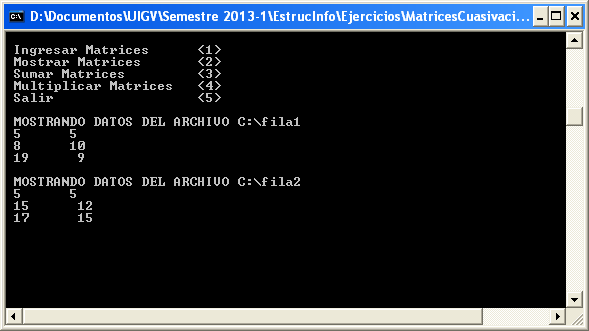
c.- Posiciones de las celdas no nulas con sus respectivos valores.





MOSTRAR MATRICES

Muestra los datos de las dos matrices en el formato especificado en la estructura de datos abstracta (clase MATRIZ).

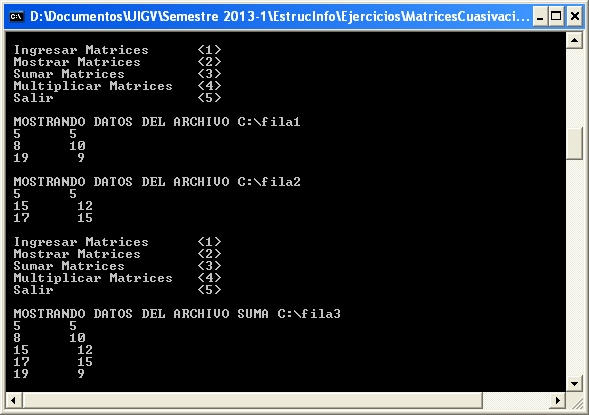


SUMAR MATRICES

Suma las dos matrices almacenadas.

Requiere que las matrices sean del mismo tamaño.

Suma las matrices y almacena el resultado en un archivo en el formato especificado en la clase MATRIZ.

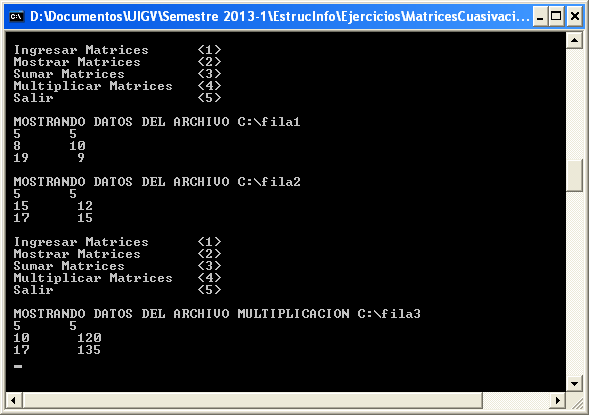


MULTIPLICAR MATRICES

Multiplica las dos matrices almacenadas.

Requiere que la cantidad de columnas de la primera matriz sea igual a la cantidad de líneas de la segunda matriz.

Multiplica las matrices y almacena el resultado en un archivo en el formato especificado en la clase MATRIZ.



4.- Conclusiones

El paradigma de la programación orientada a objetos permite mantener oculto (restricción en el acceso) los datos de los objetos de una clase, permitiendo su acceso solo a través de los métodos definidos dentro de la misma clase.

Esto permite proteger la integridad de los datos evitando que otros métodos de otras clases puedan modificarlos.

5.- Bibliografía

Gregory L. Heileman. Estructura de Datos, Algoritmos, y Programación orientada a Objetos. Editorial McGraw Hill, Primera Edición – 1996.

Andrew C. Staugaard, Jr. Técnicas Estructuradas y Orientadas a Objetos. Editorial Prentice Hall, Segunda Edición – 1997.

Luis Joyanes Aguilar, Programación Orientada a Objetos. Editorial McGraw Hill, Segunda Edición – 1998.

Abe Mizrahi, Michael Sullivan. Matemáticas Finitas. Editorial Limusa Wiley, Segunda Edición – 2001.