

INFORMÁTICA APLICADA

curso 2006-2007

PRÁCTICA 6: Clase CMatriz

Realizar un programa en Dev C++ 5.0, con las siguientes especificaciones:

- ❑ Se debe implementar la clase **CMatriz** que permita manejar matrices de cualquier dimensión. Se utilizará gestión dinámica de memoria para el almacenamiento de los datos de la matriz. Los datos se almacenan en un array bidimensional de tipo **double**.
- ❑ La definición de la clase **CMatriz** se da a continuación:

```
class CMatriz
{
    protected:
        int fil;
        int col;
        double **mat;

    public:
        CMatriz(int fil=TAM_DEF, int col=TAM_DEF); //constructor por defecto
        CMatriz(int fil, int col, double **datos); //constructor con filas, columnas y valores
        CMatriz(CMatriz &orig); //constructor copia
        ~CMatriz(void);

        CMatriz operator + (CMatriz &der);
        CMatriz operator - (CMatriz &der);
        CMatriz operator * (CMatriz &der);

        CMatriz& operator = (const CMatriz &der);

        void Visualizar(void); // visualiza los datos de la matriz
        void IntroducirValores(void);
}; // fin de la clase CMatriz
```

- ❑ A modo de ejemplo se muestra el constructor por defecto (crea una matriz 3x3 y no asigna valores).

```
CMatriz::CMatriz(int filas, int columnas)
{
    fil = filas;
    col = columnas;
    mat = new double*[fil];

    //columnas
    for(int i=0; i < fil; i++)
        mat[i] = new double[col];
}
```

- ❑ Se deberá seguir la estructura típica de proyectos seguida en la asignatura:

Fichero CMatriz.h → Definición de la clase CMatriz

Fichero CMatriz.cpp → Código de la clase CMatriz

Fichero MainCMatriz.cpp → Uso de la clase CMatriz

Tareas:

- ❑ Escribir todos los constructores que se especifican en la clase.
- ❑ Implementar el destructor de la clase. Deberá liberar toda la memoria asignada.



- ❑ Escribir el método Visualizar, que muestra por pantalla todos los datos de la matriz.
- ❑ Escribir el método IntroducirValores, que recoge por teclado todos los elementos de la matriz.
- ❑ Implementar el operador =. En el caso de que las dimensiones de las matrices no sean iguales se mostrará un mensaje de error.
- ❑ Implementar los operadores + y -. En el caso de que las dimensiones de las matrices no sean iguales se mostrará un mensaje de error.
- ❑ Implementar el operador * (multiplicación de matrices). El resultado de multiplicar $C=A*B$, con $A_{n \times m}$ y $B_{m \times k}$, el resultado C es de dimensiones $n \times k$. El elemento (r,c) de la matriz C se calcula de la siguiente manera:

$$C(r,c) = \sum_{j=0}^{m-1} A(r,j) * B(j,c)$$

- ❑ Finalmente, se deberá probar el código escribiendo una función main como la que se muestra a continuación:

```
int main(void)
{
    CMatriz m1;
    CMatriz m2;
    CMatriz m3;

    m1.IntroducirValores();
    m2.IntroducirValores();

    m1.Visualizar();
    m2.Visualizar();

    m3 = m1+m2;
    m3.Visualizar();

    m3 = m1-m2;
    m3.Visualizar();

    m3 = m1*m2;
    m3.Visualizar();

    return 0;
}
```

