

## MÔN : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

### Bài thực hành số 8.2 : Xây dựng chương trình tính tổng 2 ma trận

#### I. Mục tiêu :

- Giúp SV làm quen với cách thức viết code để đọc dữ liệu ở dạng chuỗi văn bản từ file, mã hóa chuỗi thành dạng thích hợp để xử lý/tính toán bên trong máy, cách thức để giải mã thông tin trong máy thành dạng chuỗi để ghi ra file văn bản hầu giúp người dùng có thể kiểm tra dễ dàng kết quả.

#### II. Nội dung :

- Viết code để đọc dữ liệu của 2 ma trận từ 2 file văn bản, tính ma trận tổng, xuất kết quả ma trận tổng ra file văn bản để người dùng dễ dàng kiểm tra kết quả.

#### III. Chuẩn đầu ra :

- Sinh viên nắm vững và lập trình thành thạo các đoạn code để đọc/ghi dữ liệu văn bản từ file văn bản vào các biến bên trong chương trình.

#### IV. Qui trình :

1. Chạy VS .Net, chọn menu File.New.Project để hiển thị cửa sổ New Project.
2. Mở rộng mục Visual C# trong TreeView "Project Types", chọn mục Windows, chọn icon "Console Application" trong listbox "Templates" bên phải, thiết lập thư mục chứa Project trong listbox "Location", nhập tên Project vào textbox "Name:" (td. TongMT), click button OK để tạo Project theo các thông số đã khai báo.
3. Ngay sau Project vừa được tạo ra, cửa sổ soạn code cho chương trình được hiển thị. Thêm lệnh using sau đây vào đầu file :

using System.IO;

4. Viết code cho thân class Program như sau :

```
class Program {
    static double[,] A;    //ma trận A
    static double[,] B;    //ma trận B
    static double[,] S;    //ma trận S
    static int hang, cot;
    //hàm đọc ma trận vào biến bộ nhớ
    static void ReadMT(string path, ref double[,] A, ref int hang, ref int cot) {
        //1. tạo đối tượng quản lý file
        FileStream stream = new FileStream(path, FileMode.Open);
        //2. tạo đối tượng phục vụ đọc file
        StreamReader reader = new StreamReader(stream, Encoding.ASCII);
        //3. định nghĩa các biến dữ liệu theo yêu cầu chương trình
        int i, j; string buf = "";
        //4. đọc dữ liệu từ file vào các biến
        ReadItem(reader, ref buf); hang = Int32.Parse(buf);    //đọc số hàng
        ReadItem(reader, ref buf); cot = Int32.Parse(buf);    //đọc số cột
        //phân phối vùng nhớ cho ma trận
        A = new double[hang, cot];
        //đọc từng phần tử ma trận
        for (i = 0; i < hang; i++)
            for (j = 0; j < cot; j++) {
                ReadItem(reader, ref buf);
                A[i, j] = Double.Parse(buf);    //đọc số thực
            }
    }
}
```

```

//5. đóng các đối tượng được dùng lại
reader.Close(); stream.Close();
}

//hàm ghi ma trận ra file text
static void WriteMT(string path, double[,] A, int hang, int cot) {
    //1. tạo đối tượng quản lý file
    FileStream stream = new FileStream(path, FileMode.Create);
    //2. tạo đối tượng phục vụ ghi file
    StreamWriter writer = new StreamWriter(stream, Encoding.ASCII);
    //3. định nghĩa các biến dữ liệu theo yêu cầu chương trình
    int i, j;
    //4. ghi dữ liệu từ các biến ra file
    writer.Write(hang); writer.Write(" "); //ghi số hàng và dấu ngăn
    writer.Write(cot); writer.WriteLine(); //ghi số cột và dấu ngăn
    //ghi ma trận
    for (i = 0; i < hang; i++) { //ghi từng hàng ma trận
        for (j = 0; j < cot; j++) {
            writer.Write(A[i, j]); writer.Write(","); //ghi phần tử i,j
        }
        writer.WriteLine(); //ghi dấu xuống hàng
    }
    //5. đóng các đối tượng được dùng lại
    writer.Close(); stream.Close();
}

//hàm đọc chuỗi miêu tả 1 dữ liệu nào đó
static void ReadItem(StreamReader reader, ref String buf) {
    char ch = '\0';
    //thiết lập chuỗi nhập được lúc đầu là rỗng
    buf = "";
    while (reader.EndOfStream != true) { //lặp đọc bỏ các dấu ngăn
        ch = (char)reader.Read(); //đọc 1 ký tự
        if (ch != ',' && ch != '\r' && ch != '\n')
            break; //nếu là ký tự bình thường thì dừng
    }
    buf += ch.ToString();
    //lặp đọc các ký tự của chuỗi dữ liệu
    while (reader.EndOfStream != true) {
        ch = (char)reader.Read(); //đọc 1 ký tự
        if (ch == ',' || ch == '\r' || ch == '\n')
            return; //nếu là dấu ngăn thì dừng
        buf += ch.ToString(); //chứa ký tự vào bộ đệm
    }
}

//điểm nhập chương trình
static void Main(string[] args) {
    int i, j, h = 0, c = 0;
    //đọc ma trận A từ file c:\A.txt
    ReadMT("c:\\a.txt", ref A, ref hang, ref cot);
    //đọc ma trận B từ file c:\B.txt

```

```

ReadMT("c:\\b.txt", ref B, ref h, ref c);
if (h != hang || c != cot) {
    Console.WriteLine("Hai ma trận A, B không cùng kích thước");
    return;
}
//phân phối ma trận tổng S
S = new double[hang, cot];
//tính ma trận tổng S
for (i = 0; i < hang; i++)
    for (j = 0; j < cot; j++) S[i, j] = A[i, j] + B[i, j];
//xuất ma trận kết quả ra file c:\\S.txt
WriteMT("c:\\S.txt", S, hang, cot);
} //hết hàm Main
} //hết class program

```

5. Chọn menu Debug.Start Debugging để dịch và chạy ứng dụng. Lưu ý là trước khi chạy ứng dụng, hãy dùng chương trình soạn thảo văn bản (Notepad, WordPad,...) soạn thảo 2 file c:\\A.txt, c:\\B.txt chứa dữ liệu của 2 ma trận A và B.

Thí dụ nội dung c:\\A.txt như sau :

```

5, 7
1, 2, 3, 4, 5, 6, 7
8, 9, 10, 11, 12, 13, 14
15, 16, 17, 18, 19, 20, 21
22, 23, 24, 25, 26, 27, 28
29, 30, 31, 32, 33, 34, 35

```

Và nội dung c:\\B.txt như sau :

```

5, 7
2, 3, 4, 5, 6, 7, 8
9, 10, 11, 12, 13, 14, 15
16, 17, 18, 19, 20, 21, 22
23, 24, 25, 26, 27, 28, 29
30, 31, 32, 33, 34, 35, 36

```