

MÔN : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bài thực hành số 2.2 : Viết chương trình xử lý đa thức

I. Mục tiêu :

- Giúp SV ôn lại qui trình viết 1 ứng dụng VC# đơn giản, tập trung vào việc sử dụng các kiểu dữ liệu cơ bản định sẵn.

II. Nội dung :

- Xây dựng chương trình nhỏ chạy ở chế độ textmode (console), thực hiện các yêu cầu sau :

1. miêu tả kiểu để lưu giữ từng toán hạng của đa thức gồm 2 thông tin : hệ số và số mũ.
2. sắp xếp danh sách các toán hạng của đa thức theo thứ tự giảm dần của số mũ, phương pháp sắp xếp là Insertion Sort.
3. chuẩn hóa đa thức : mỗi số mũ chỉ xuất hiện trong đa thức 1 lần.

Thí dụ có đa thức gốc $R(x) = x^3 + x^2 + x + 2x^2 + 2x + 2$

sau khi chuẩn hóa thành $R(x) = x^3 + 3x^2 + 3x + 2$

4. Nhân 2 đa thức.

III. Chuẩn đầu ra :

- Sinh viên nắm vững và viết thành thạo các ứng dụng nhỏ chạy ở chế độ textmode và biết cách dùng các kiểu dữ liệu cơ bản định sẵn khi cần.

IV. Phân tích :

1. Ta dùng kiểu struct để miêu tả thông tin về mỗi số hạng của đa thức như sau :

```
struct SoHanh {  
    public int somu; //số mũ  
    public float heso; //hệ số  
};
```

2. Ý tưởng sắp xếp danh sách theo giải thuật "Insertion Sort" như sau :

Lập xác định vị trí đúng cho từng số hạng i (từ 0 tới n) của đa thức :

- Lập tìm số hạng có số mũ lớn nhất từ vị trí i tới n
- Hoán vị nó với số hạng i

Từ thuật giải được viết bằng ngôn ngữ tự nhiên ở trên, ta sẽ dịch nó thành hàm SortItems() được viết bằng VC# như trong mã nguồn dưới đây.

3. Chuẩn hóa đa thức : được thực hiện thông qua 2 bước chính :

- sắp xếp đa thức theo thứ tự số mũ giảm dần (gọi hàm SortItems() ở bước 2).
- duyệt tuần tự từ đầu đến cuối đa thức để cộng các số hạng liên tiếp có cùng số mũ lại thành 1 số hạng.

Từ thuật giải được viết bằng ngôn ngữ tự nhiên ở trên, ta dịch nó thành hàm Chuanhoa() được viết bằng VC# như trong mã nguồn dưới đây.

4. Nhân đa thức : được thực hiện thông qua 2 bước chính :

- lập nhân từng số hạng của đa thức 1 với tất cả các số hạng của đa thức 2. Kết quả được lưu vào đa thức kết quả.
- Chuẩn hóa đa thức kết quả.

Từ thuật giải được viết bằng ngôn ngữ tự nhiên ở trên, ta dịch nó thành hàm Nhan() được viết bằng VC# như trong mã nguồn dưới đây.

V. 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. Dathuc), click button OK để tạo Project theo các thông số đã khai báo.
3. Ngay sau khi Project vừa được tạo ra, cửa sổ soạn code cho chương trình được hiển thị. Hiệu chỉnh code của file Program.cs để có nội dung như sau :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Dathuc {
class Program {
//định nghĩa kiểu miêu tả 1 số mũ
private struct Sohang {
    public float heso; //hệ số
    public int somu; //số mũ
    public Sohang(float h, int s) { heso = h; somu = s; }
};

//hàm sắp xếp đa thức theo thứ tự số mũ giảm dần
static void SortItems (Sohang[] dathuc, int soluong) {
//định nghĩa các biến cần dùng
int i,j, max;
Sohang temp;
//Lặp xác định vị trí đúng cho từ số mũ
for (i = 0; i < soluong-1; i++) {
    //Tìm phần tử có số mũ lớn nhất từ vị trí i
    max = i;
    for (j = i + 1; j < soluong; j++)
        if (dathuc[max].somu < dathuc[j].somu)
            max = j;
    //hoán vị phần tử có số mũ lớn nhất về vị trí i
    temp = dathuc[i];
    dathuc[i] = dathuc[max];
    dathuc[max] = temp;
}
}

//hàm chuẩn hóa đa thức, mỗi số mũ chỉ có 1 phần tử
static int Chuanhoa (Sohang[] dathuc, int soluong) {
//định nghĩa các biến cần dùng
int i,j;
Sohang[] temp = new Sohang[soluong];
//sắp xếp thứ tự các phần tử trong đa thức
SortItems(dathuc,soluong);
//Lập việc gộp các phần tử có số mũ giống nhau
j = 0; temp[0] = dathuc[0];
i = 1;
```

```

while (i < soluong)
    if (temp[j].somu == dathuc[i].somu) {
        temp[j].heso += dathuc[i].heso;
        i++;
    } else {
        j++;
        temp[j] = dathuc[i];
        i++;
    }
//copy kết quả về dathuc gốc
for (i = 0; i <= j; i++)
    dathuc[i] = temp[i];
//return số phần tử trong đa thức chuẩn hóa
return j+1;
}

//hàm in đa thức để kiểm tra
static void Indathuc(Sohang[] dathuc, int soluong) {
int i;
    if (soluong == 0) return;
    //in phần tử đầu tiên của đa thức
    Console.WriteLine(dathuc[0].heso + "x^" + dathuc[0].somu + " ");
    //lặp in từng phần tử còn lại của đa thức
    for (i = 1; i < soluong; i++)
        if (dathuc[i].heso < 0)
            Console.WriteLine(dathuc[i].heso + "x^" + dathuc[i].somu + " ");
        else
            Console.WriteLine(" + " + dathuc[i].heso + "x^" + dathuc[i].somu + " ");
    //in ký tự xuống dòng
    Console.WriteLine();
}

//hàm nhân 2 đa thức
static void Nhan (Sohang[] dathuc1, int soluong1, Sohang[] dathuc2, int soluong2, ref Sohang[]
dathuc3, ref int soluong3) {
//định nghĩa các biến cần dùng
int i,j,k;
    //cấp phát bộ nhớ chứa đa thức kết quả
    dathuc3 = new Sohang[soluong1*soluong2];
    k = 0;
    //lặp nhân từng số hạng của đa thức 1
    for (i=0; i < soluong1; i++)
        //với tất cả các số hạng của đa thức 2
        for (j = 0; j < soluong2; j++) {
            dathuc3[k].somu = dathuc1[i].somu+dathuc2[j].somu;
            dathuc3[k].heso = dathuc1[i].heso*dathuc2[j].heso;
            k++;
        }
    //chuẩn hóa đa thức kết quả
    soluong3 = Chuanhoa (dathuc3,k);
}

```

```

//chương trình chính thử dùng các hàm chức năng
static void Main(string[] args) {
    Sohang[] dathuc1 = new Sohang[] { new Sohang(1,3), new Sohang(1,2), new
    Sohang(1,1),new Sohang(-2,2),new Sohang(2,1),new Sohang(2,0)};
    //1. thử sắp xếp và in ra để kiểm tra
    Console.WriteLine("Đa thức gốc là : ");
    Indathuc(dathuc1,6);
    SortItems(dathuc1,6);
    Console.WriteLine("Đa thức được sort là : ");
    Indathuc(dathuc1,6);

    Sohang[] dathuc2 = new Sohang[] { new Sohang(1,3), new Sohang(1,2), new Sohang(1,1),
    new Sohang(-2,2), new Sohang(2,1),new Sohang(2,0)};
    //2. thử chuẩn hóa đa thức và in ra để kiểm tra
    Console.WriteLine("Đa thức gốc là : ");
    Indathuc(dathuc2,6);
    int somu = Chuanhoa(dathuc2,6);
    Console.WriteLine("Đa thức được chuẩn hóa là : ");
    Indathuc(dathuc2,somu);

    //3. thử nhân 2 đa thức và in ra để kiểm tra
    Sohang[] dathuc3 = new Sohang[] { new Sohang(1,3), new Sohang(1,2), new Sohang(1,1),
    new Sohang(-2,2), new Sohang(2,1),new Sohang(2,0)};
    Sohang[] dathuc4 = new Sohang[] { new Sohang(1,3), new Sohang(1,2), new Sohang(1,1),
    new Sohang(-2,2), new Sohang(2,1),new Sohang(2,0)};
    Sohang[] dathuc5 = new Sohang[1];
    int soluong=1;
    Nhan(dathuc3, 6, dathuc4, 6, ref dathuc5, ref soluong);
    Console.WriteLine("Đa thức kết quả là : ");
    Indathuc(dathuc5, soluong);
    //4. chờ người dùng ấn Enter để đóng cửa sổ Console lại.
    Console.Write("Ấn Enter để dừng chương trình : ");
    Console.Read();
} //hết hàm Main
} //hết class Program
} //hết namespace

```

4. Chọn menu Debug.Start Debugging để dịch và chạy ứng dụng. Hãy kiểm tra kết quả hiển thị của chương trình.
5. Thử hiệu chỉnh lại nội dung các đa thức trong mã nguồn rồi chạy lại phần mềm để kiểm tra kết quả.
6. Hãy suy nghĩ và hiệu chỉnh lại phần mềm để có thể nhập đa thức từ người dùng.