

## **MÔN : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

### **Bài thực hành số 4.2 : Viết chương trình tạo từ**

#### **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 như struct và array.

#### **II. Nội dung :**

- Xây dựng chương trình nhỏ chạy ở chế độ textmode (console), thực hiện các yêu cầu sau :
  1. chờ user nhập vào chuỗi có n ký tự khác nhau.
  2. tìm tất cả phương án để tạo từ dài n ký tự, mỗi ký tự là ký tự do user nhập, mỗi ký tự được dùng 1 lần trong từ.
  3. in các từ tạo được để người dùng xem dễ dàng.

#### **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, nhất là kiểu struct và array, khi cần.

#### **IV. Phân tích :**

1. Ta dùng thuật giải backtracking để tìm các phương án tạo từ như sau :
  - + Lập tìm chỉ số gốc của ký tự được dùng cho từng vị trí (cột) trong từ, từ trái sang phải :
    - Nếu sắp được cho cột c thì tăng c 1 đơn vị để tiếp tục cột kế tiếp, nếu cột c đã vượt qua MAX thì đã tìm được 1 từ mới.
    - Nếu không sắp được cho cột c thì giảm c 1 đơn vị để thử tiếp các khả năng còn lại của cột cũ này. Nếu c < 0 thì hết cách.
2. Để lưu vết xử lý trên từng cột, ta dùng biến array chiso[MAX], mỗi phần tử chiso[c] chứa chỉ số ký tự gốc đã xử lý lần cuối trước đây (-1 : chưa, MAX : hết).  
Từ thuật giải được viết bằng ngôn ngữ tự nhiên ở trên, ta dịch nó thành chương trình 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. TaoTu), 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 TaoTu
{
    class Program
    {
        //định nghĩa các thuộc tính dữ liệu
        const int MAXLEN = 20; //số ký tự max cần dùng để tạo từ
        static String srcstr; //chuỗi các ký tự khác nhau để tạo từ
```

```

static int sotu; //số từ đã tìm được
static int[] chiso = new int[MAXLEN + 1]; //danh sách chỉ số ký tự đã thử trong từng cột
static int sokytu; //số ký tự gốc được dùng để tạo từ
static int c; //vị trí cột đang sắp

//Điểm nhập của chương trình
static void Main(string[] args)
{
    //nhập chuỗi ký tự dùng để tạo từ
    do
    {
        Console.Write("Hãy nhập chuỗi các ký tự cần dùng : ");
        srcstr = Console.ReadLine();
        //xác định số ký tự được dùng
        sokytu = srcstr.Length;
    } while (sokytu >= MAXLEN);

    // khởi động trạng thái xuất phát
    for (c = 0; c < sokytu; c++) chiso[c] = -1;
    //bắt đầu đặt ký tự chỉ số 0 vào cột đầu (cột 0) của từ cần xây dựng
    chiso[0] = 0;
    //bắt đầu tìm ký tự cho cột thứ 2 (chỉ số là 1) của từ cần xây dựng
    c = 1;
    while (Tim1cach())
    {
        // tìm được 1 từ mới
        InKetqua();
        //lùi lại cột trước để tiếp tục tìm cách sắp khác
        c--;
    }
    // hết cách --> dừng chương trình
    Console.WriteLine("Số từ xây dựng được là " + sotu);
    //chờ người dùng ấn Enter để đóng cửa sổ Console lại.
    Console.Write("Ấn Enter để dừng chương trình : ");
    Console.Read();
    return;
}

//cố gắng tìm 1 phương án sắp các ký tự thành 1 từ mới theo yêu cầu
static bool Tim1cach() {
    int h;
    while (c < sokytu)
    {
        // tìm chỉ số ký tự cho ký tự cột c
        h = timkytu(c);
        if (h >= 0)
        { //tìm được ==> lưu chỉ số vào danh sách kết quả rồi tăng chỉ số cột để tiếp tục
            chiso[c++] = h;
            continue;
        }
        //trường hợp không tìm được ký tự ở cột c
        if (c > 0)

```

```

    {
        //cột c chưa phải là cột đầu tiên => xét lại cột c từ ký tự đầu rồi lùi cột c
        chiso[c--] = -1;
        continue;
    }
    //hết cách
    return false;
}
return true;
} //hết hàm Tim1cach

//hàm tìm chỉ số hàng cho con hậu ở cột c
static int timkytu(int c)
{
    int h, hmin = chiso[c] + 1;
    for (h = hmin; h < sokytu; h++) // thử ký tự chỉ số h
        if (duocpheap(h, c)) return h;
    return -1;
}

// Kiểm tra xem có thể xếp ký tự thứ h vào vị trí thứ c trong từ cần xây dựng
static bool duocpheap(int h, int c)
{
    int k;
    for (k = 0; k < c; k++)
        if (chiso[k] == h) return false;
    return true;
}

//Kiểm tra từ, nếu có nghĩa thì trả về TRUE, nếu không trả về FALSE
static bool conghia(String str)
{
    // Bạn viết lại hàm này theo yêu cầu riêng
    //ở đây ta coi mọi từ đều có nghĩa
    return true;
}

//hàm in từ mới tìm theo yêu cầu
static void InKetqua()
{
    String desstr;
    int k;
    //Xây dựng từ mới tìm được
    desstr = "";
    for (k = 0; k < sokytu; k++)
        desstr += srcstr[chiso[k]];
    if (conghia(desstr)) //kiểm tra từ có nghĩa trước khi in
    {
        Console.WriteLine("Từ thứ " + (++sotu) + " : " + desstr);
    }
}
}

```

```
    }//hết class Program  
}//hết namespace
```

4. Chọn menu Debug.Start Debugging để dịch và chạy ứng dụng, nhập 1 chuỗi ký tự gốc rồi xem và kiểm tra kết quả hiển thị của chương trình.
5. Lặp lại bước 4 cho nhiều chuỗi ký tự có độ dài khác nhau.