

MÔN : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bài thực hành số 3.3 : Viết chương trình "Mã đi tuần tra"

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 người dùng nhập kích thước bàn cờ, tọa độ hàng cột của con mã xuất phát.
 2. tìm tất cả phương án để giúp con mã đi tuần qua tất cả các ô cờ, mỗi ô chỉ được phép đi qua 1 lần.
 3. in các phương án dạng ma trận 2 chiều 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 giúp con mã đi tuần qua tất cả các ô cờ :

- + Ở mỗi vị trí, con mã có tối đa 8 khả năng đi tiếp.
- + Xuất phát từ vị trí đầu, lập tìm vị trí con mã kế tiếp (từ khả năng 1 tới 8) :
 - Nếu tìm được thì tăng chỉ số nước đi 1 đơn vị rồi tiếp tục cho đến khi con mã đi hết các ô, lúc này đã tìm được 1 cách mới.
 - Nếu không tìm được thì giảm chỉ số nước đi 1 đơn vị, thử tiếp các khả năng còn lại của vị trí cũ này. Nếu chỉ số nước đi < 0 thì hết cách.

2. Để lưu vết đi con mã, ta dùng biến array Nuocdi[MAX], mỗi phần tử Nuocdi[i] chứa tọa độ con mã tương ứng và chỉ số cách đã xử lý lần cuối trước đây (-1 : chưa, 8 : 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. Madituan), 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 Madituan
{
    class Program
    {
```

```

// Kiểu chứa thông tin về 1 bước đi
private struct ItemRec {
    public int x, y;
    public int huong;
};
// Các biến dữ liệu chính
static int Size;
static int[,] Banco;
static ItemRec[] Nuocdi;
static int SoNuocdi;
static int SoNghiem;

// Chương trình chính
static void Main(string[] args)
{
    Khoidong();
    while (DiBuocNua())
        if (SoNuocdi == Size * Size-1)
        {
            //tìm được 1 cách mới, in cách đi mới tìm được
            InKetqua();
            Banco[Nuocdi[SoNuocdi].x, Nuocdi[SoNuocdi].y] = -1;
            //lùi con mã 1 bước
            SoNuocdi = SoNuocdi - 1;
            //tiếp tục tìm cách đi khác
            Nuocdi[SoNuocdi].huong = Nuocdi[SoNuocdi].huong + 1;
        }
    //chờ người dùng ấn Enter để đóng cửa sổ Console lại.
    Console.Write("Ấn Enter để dừng chương trình : ");
    Console.Read();
}

//hàm khởi động các giá trị đầu của chương trình
static void Khoidong() {
    int i,j;
    int row, col;
    String buf;
    do //chờ nhập kích thước bàn cờ
    {
        Console.Write("Nhập số hàng (cột) của bàn cờ: ");
        buf = Console.ReadLine(); Size = Int32.Parse(buf);
    } while (Size <= 3 || Size > 9);
    do //chờ nhập chỉ số hàng xuất phát
    {
        Console.Write("Nhập chỉ số hàng xuất phát của con mã : ");
        buf = Console.ReadLine(); row = Int32.Parse(buf);
    } while (row < 0 || Size <= row);
    do //chờ nhập chỉ số cột xuất phát
    {
        Console.Write("Nhập chỉ số cột xuất phát của con mã : ");
        buf = Console.ReadLine(); col = Int32.Parse(buf);
    } while (col < 0 || Size <= col);
}

```

```

//phân phối vùng nhớ chứa array Banco và danh sách Nuocdi
Banco = new int[Size, Size];
Nuocdi = new ItemRec[Size * Size];
for (i = 0; i < Size * Size; i++)
    Nuocdi[i] = new ItemRec();
//khởi động trạng thái đầu cho các vị trí
for (i = 0; i < Size; i++)
    for (j = 0; j < Size; j++)
        Banco[i, j] = -1; // con mã chưa đi qua ô [i,j]

SoNghiem = 0;
// Thiết lập nước đi đầu tiên của con mã
Nuocdi[0].x = col;
Nuocdi[0].y = row;
Nuocdi[0].huong = 0;
SoNuocdi = 0;
//chọn khả năng đi tiếp của vị trí đầu tiên của con mã
Banco[Nuocdi[SoNuocdi].x, Nuocdi[SoNuocdi].y] = 0;
}

//hàm tìm vị trí kế tiếp của con mã
// trả về TRUE nếu tìm được, FALSE nếu không tìm được
static bool DiBucNua() {
    int x=0, y=0;
    bool RetVal;
    RetVal = false;
    do { // lặp tìm vị trí kế
        while (RetVal==false && Nuocdi[SoNuocdi].huong < 8) {
            switch (Nuocdi[SoNuocdi].huong) { //thử hướng đi hiện tại
                case 0 :
                    x = Nuocdi[SoNuocdi].x + 2;
                    y = Nuocdi[SoNuocdi].y - 1;
                    break;
                case 1 :
                    x = Nuocdi[SoNuocdi].x + 1;
                    y = Nuocdi[SoNuocdi].y - 2;
                    break;
                case 2 :
                    x = Nuocdi[SoNuocdi].x - 1;
                    y = Nuocdi[SoNuocdi].y - 2;
                    break;
                case 3 :
                    x = Nuocdi[SoNuocdi].x - 2;
                    y = Nuocdi[SoNuocdi].y - 1;
                    break;
                case 4 :
                    x = Nuocdi[SoNuocdi].x - 2;
                    y = Nuocdi[SoNuocdi].y + 1;
                    break;
                case 5 :
                    x = Nuocdi[SoNuocdi].x - 1;
                    y = Nuocdi[SoNuocdi].y + 2;

```

```

        break;
    case 6 :
        x = Nuocdi[SoNuocdi].x + 1;
        y = Nuocdi[SoNuocdi].y + 2;
        break;
    case 7 :
        x = Nuocdi[SoNuocdi].x + 2;
        y = Nuocdi[SoNuocdi].y + 1;
        break;
    }
    if (0 <= x && x < Size && 0 <= y && y < Size && Banco[x,y] == -1) {
        // nếu được thì ghi nhận
        SoNuocdi = SoNuocdi + 1;
        Banco[x,y] = SoNuocdi;
        Nuocdi[SoNuocdi].x = x;
        Nuocdi[SoNuocdi].y = y;
        Nuocdi[SoNuocdi].huong = 0;
        RetVal = true;
    } else // nếu không tìm được vị trí kế tiếp
        Nuocdi[SoNuocdi].huong = Nuocdi[SoNuocdi].huong + 1;
    }
    if (RetVal == false && SoNuocdi != 0) {
        // nếu không tìm được vị trí kế thì lùi con mã 1 bước
        Banco[Nuocdi[SoNuocdi].x, Nuocdi[SoNuocdi].y] = -1;
        SoNuocdi = SoNuocdi - 1;
        Nuocdi[SoNuocdi].huong = Nuocdi[SoNuocdi].huong + 1;
    }
    } while (!RetVal && (SoNuocdi != 0));
    return RetVal;
}

// In kết quả con mã đi trên bàn cờ
static void InKetqua()
{
    int h, c;
    SoNghiem = SoNghiem + 1;
    Console.WriteLine("Cách đi thứ : " + SoNghiem);
    for (h = 0; h < Size; h++)
    {
        // Hiển thị hàng lưới ngang bàn cờ
        for (c = 0; c < Size; c++) Console.Write("+----");
        Console.WriteLine("+");
        // Hiển thị nội dung hàng thứ h bàn cờ
        for (c = 0; c < Size; c++)
            Console.Write("| {0:D2} ", Banco[h, c]);
        Console.WriteLine("|");
    }
    // Hiển thị hàng lưới ngang cuối cùng của bàn cờ
    for (c = 0; c < Size; c++) Console.Write("+----");
    Console.WriteLine("+");
}
} //hết class Program

```

}//hết namespace

4. Chọn menu Debug.Start Debugging để dịch và chạy ứng dụng. Hãy nhập kích thước bàn cờ và tọa độ miêu tả vị trí xuất phát của con mã rồi xem và kiểm tra kết quả hiển thị của chương trình.
5. Lập lại bước 4 nhiều lần với nhiều kích thước khác nhau và tọa độ xuất phát khác nhau để quan sát kết quả. Lưu ý do thuật giải rất nhiều bước, còn máy thì tốc độ hạn chế nên nếu chọn kích thước từ 8 trở lên thì sẽ tốn rất nhiều ngày mới có thể có kết quả.