

# Công nghệ .NET

## Bài 3 – Làm quen với C#

Phạm Ngọc Hưng – Khoa CNTT

Email: [hung.phamngoc@phenikaa-uni.edu.vn](mailto:hung.phamngoc@phenikaa-uni.edu.vn)

Nguồn: Nguyễn Thành Trung – Khoa CNTT

# Nội dung

- Cấu trúc rẽ nhánh
  - If ... else
  - switch ... case
- Cấu trúc lặp
  - while
  - do ... while ...
  - for
- Mảng (Array)

# Cấu trúc rẽ nhánh

- Cấu trúc rẽ nhánh if-else  
if (condition)

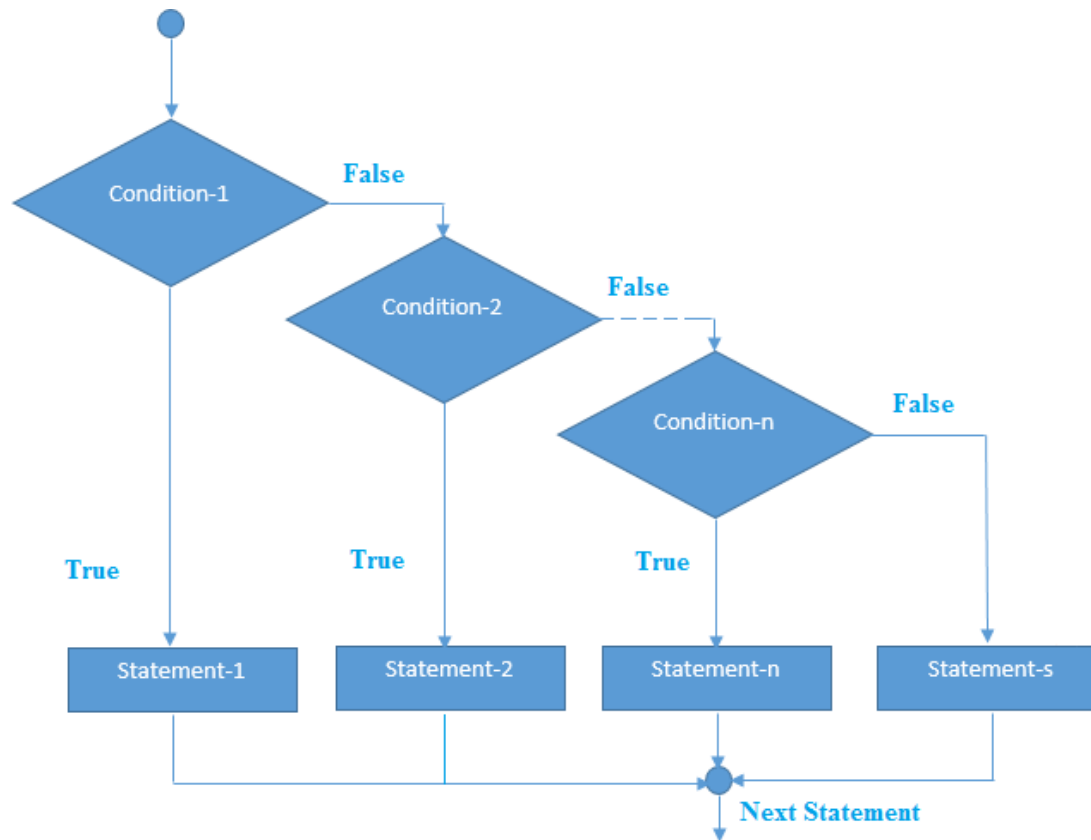
```
{  
    statements 1  
} else if {  
    statements 2  
    ...  
} else {  
    statements n  
}
```

# Cấu trúc rẽ nhánh

- Cấu trúc rẽ nhánh if-else
  - Trong đó:
    - Condition là biểu thức logic (trả về giá trị True hoặc False), trong đó sử dụng các phép toán so sánh: >, <, >=, <=, ==, != và các phép toán logic &&, ||, !
    - Statement 1,2,...,n: lệnh/khối lệnh
  - Một vài lưu ý
    - Nếu Statement là lệnh đơn, có thể bỏ cặp dấu { }
    - If là bắt buộc, còn nhánh else { } là không bắt buộc
    - Có thể lồng nhiều if-else với nhau

# Cấu trúc rẽ nhánh

- Cấu trúc rẽ nhánh if-else



# Cấu trúc rẽ nhánh

- Ví dụ: Nhập từ bàn phím một số nguyên là điểm theo thang 10 của sinh viên. Hiển thị ra màn hình điểm chữ (A, B, C, D).

# Cấu trúc rẽ nhánh

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

namespace ConsoleApp1
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Nhap vao diem theo thang 10:");
            int num = Convert.ToInt32(Console.ReadLine());
            if (num < 0 || num > 10)
            {
                Console.WriteLine("Khong dung dinh dang");
            }
            else if (num >= 0 && num < 4)
            {
                Console.WriteLine("Fail");
            }
            else if (num >= 4 && num < 5)
            {
                Console.WriteLine("Diem D");
            }
        }
    }
}
```

# Cấu trúc rẽ nhánh

```
else if (num >= 5 && num < 7)
{
    Console.WriteLine("Diem C");
}
else if (num >= 7 && num < 8)
{
    Console.WriteLine("Diem B");
}
else if (num >= 8 && num < 9)
{
    Console.WriteLine("Diem A");
}
else
{
    Console.WriteLine("Diem A+");
}
Console.ReadKey();
}
```



# Cấu trúc rẽ nhánh

- Cấu trúc switch ... case

- Cú pháp:

```
switch (expression)
```

```
{
```

```
    case <value_1>:
```

```
        //Các lệnh
```

```
        break;
```

```
    case <value_n>:
```

```
        //Các lệnh
```

```
        break;
```

```
    default:
```

```
        //Các lệnh
```

```
        break;
```

```
}
```

# Cấu trúc rẽ nhánh

- Cấu trúc switch ... case
  - Ví dụ

0 references

```
static void Main(string[] args)
{
    Console.WriteLine("Enter a number:");
    int num = Convert.ToInt32(Console.ReadLine());

    switch (num)
    {
        case 10: Console.WriteLine("It is 10"); break;
        case 20: Console.WriteLine("It is 20"); break;
        case 30: Console.WriteLine("It is 30"); break;
        default: Console.WriteLine("Not 10, 20 or 30"); break;
    }
}
```

# Cấu trúc lặp while

- Cú pháp:

```
while (condition )  
{  
    //Các lệnh  
}
```

- Chừng nào biểu thức logic còn nhận giá trị true thì còn thực hiện các lệnh

# Cấu trúc lặp while

- Cú pháp:

do

{

//Các lệnh

} while ( condition )

- Thực hiện các lệnh rồi kiểm tra giá trị biểu thức logic. Nếu biểu thức logic còn nhận giá trị true thì thực hiện lặp lại các lệnh
- Có thể sử dụng nhiều cấu trúc lặp lồng nhau

# Cấu trúc lặp for

- Cú pháp:

```
for (initialization; condition; incr/decr)
{
    //Khối lệnh
}
```

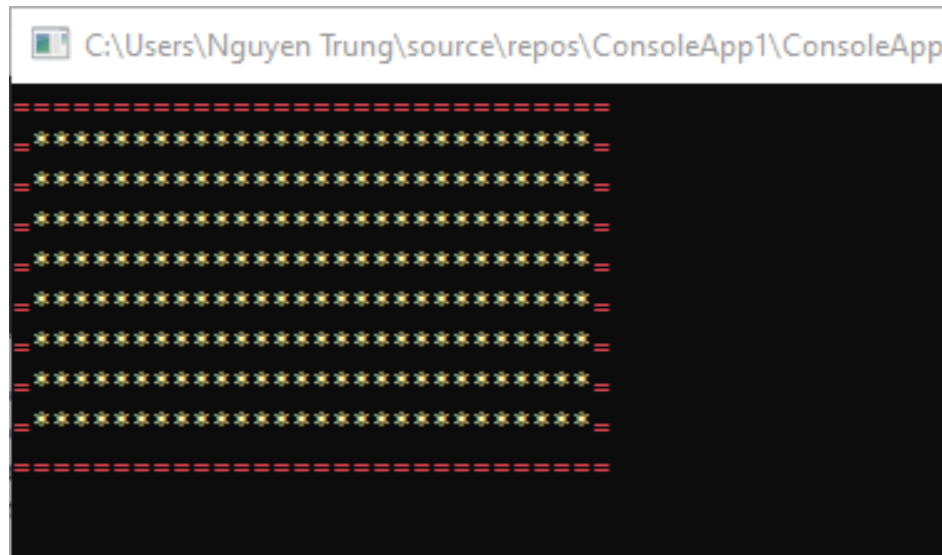
- Thực hiện khối lệnh một số lần xác định

# Điều khiển vòng lặp

- **break**: thoát khỏi vòng lặp. Khi gặp lệnh này, tất cả các lệnh đứng sau nó sẽ không được thực hiện nữa, đồng thời kết thúc vòng lặp.
- **continue**: thoát khỏi chu kỳ thực hiện hiện tại của vòng lặp. Khi gặp lệnh này, tất cả các lệnh đứng sau nó không thực hiện nữa, vòng lặp sẽ chuyển sang chu kỳ tiếp theo

# Cấu trúc lặp for

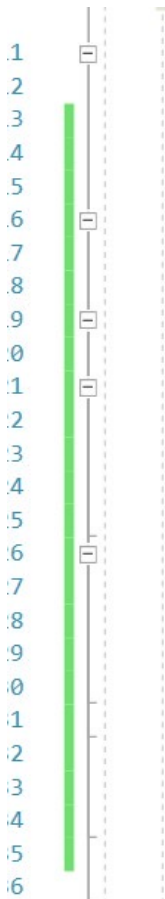
- Bài tập: Sử dụng vòng lặp for in ra hình như sau



- Gợi ý:
  - Sử dụng 2 vòng lặp for lồng nhau để in từ trên xuống và in từng hàng
  - Sử dụng cách thiết lập màu cho ký tự

# Cấu trúc lặp for

- Thực hiện:

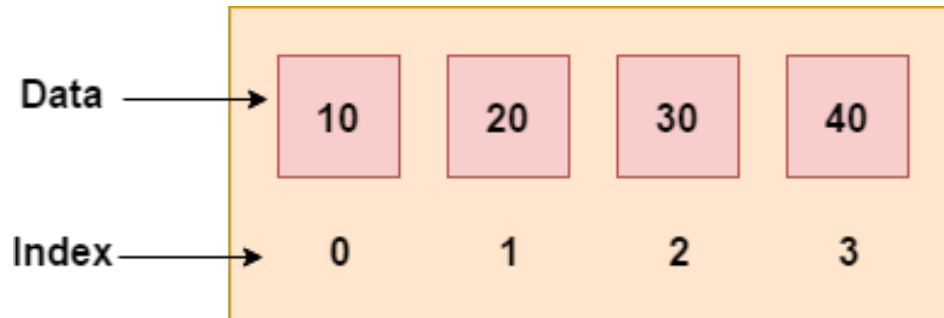


```
0 references
static void Main(string[] args)
{
    int N = 10;
    int M = 30;
    //Ve tu tren xuong
    for (int i = 0; i < N; i++)
    {
        //Ve tu trai sang
        for (int j = 0; j < M; j++)
        {
            if(i % (N-1) == 0 || ((i % (N-1) != 0 ) && (j % (M-1) == 0)))
            {
                Console.ForegroundColor = ConsoleColor.Red;
                Console.Write("=");
            }
            else
            {
                Console.ForegroundColor = ConsoleColor.Yellow;
                Console.Write("*");
            }
        }
        //Xuong dong sau moi lan
        Console.WriteLine();
    }
    Console.ReadKey();
}
```



# Mảng

- Mảng
  - Giống như C, mảng trong C# là một tập hợp các phần tử cùng kiểu dữ liệu có vị trí bộ nhớ liền nhau. Trong C#, mảng là một object của kiểu cơ sở `System.Array`.
  - Các phần tử phân biệt với nhau bằng chỉ số phần tử. Trong C# chỉ số phần tử là các số nguyên không âm và bắt đầu từ 0, 1, 2, 3...



# Mảng

- Đặc điểm:

- Các phần tử trong mảng dùng chung một tên và được truy xuất thông qua chỉ số phần tử.
- Một mảng cần có giới hạn số phần tử mà mảng có thể chứa.
- Phải cấp phát vùng nhớ mới có thể sử dụng mảng.
- Vị trí ô nhớ của các phần tử trong mảng được cấp phát liên tiếp nhau.

# Mảng

- Có 3 cách khai báo và khởi tạo giá trị cho mảng
  - Cách 1: **Khai báo**, **cấp phát** vùng nhớ
    - Ví dụ: `int[] numList = new int[10];`
    - Trong C#, khi khai báo biến mà không gán giá trị thì biến sẽ nhận một giá trị mặc định (int – 0), (bool – false), (object – null)...
  - Cách 2: **Khai báo**, **cấp phát** và **khởi tạo** giá trị cho mảng
    - `int[] numList = new int[] {1, 2, 3, 4, 5};`
  - Cách 3: **Khởi tạo** giá trị cho mảng
    - `int[] numList = {1, 2, 3, 4, 5};`
    - Bản chất ở đây khi biên dịch, trình biên dịch vẫn xem xét số phần tử được khởi tạo để cấp phát bộ nhớ sau đó mới khởi tạo giá trị cho mảng.

# Mảng

- Sử dụng mảng

- Các phần tử của mảng được truy xuất thông qua chỉ số phần tử và cặp dấu []. Có thể xem các phần tử của mảng như là các biến đơn và thao tác như thao tác với biến bình thường

- Ví dụ:

```
//Khai báo, cấp phát và khởi tạo mảng kiểu string tên là K13Class  
string[] K13Class = new string[] { "Nam", "Nga", "Viet"};  
//Truy xuất đến phần tử thứ 2 của mảng  
Console.WriteLine(K13Class[1]);
```

# Mảng

- Sử dụng mảng
  - Một số thuộc tính/phương thức

Tên thuộc tính /phương thức	Ý nghĩa
<b>Length</b>	Thuộc tính trả về số nguyên kiểu <b>int</b> là số phần tử tối đa của mảng
<b>Sort()</b>	Phương thức thực hiện sắp xếp mảng theo một thứ tự
<b>Clear()</b>	Phương thức xóa hết dữ liệu trong mảng và đưa về giá trị mặc định của kiểu. Lưu ý là chỉ xóa <b>giá trị</b> , vùng nhớ vẫn còn và có thể tiếp tục sử dụng mà không cần cấp phát lại.
<b>Copy()</b>	Sao chép giá trị của mảng ra một vùng nhớ mới
<b>Reverse()</b>	Phương thức thực hiện đảo ngược thứ tự của mảng 1 chiều

- Ví dụ: Cho mảng số nguyên `numList[5]`.
  - `numList.sort()` sẽ cho phép sắp xếp mảng.

# Mảng

- Sử dụng mảng

- Duyệt mảng: Cho mảng số nguyên numList[5].

- Cách 1

```
for (int i = 0; i < 5; i++)  
{  
    // Do something  
}
```

- Cách 2

```
for (int i = 0; i < numList.Length; i++)  
{  
    // Do something  
}
```

# Mảng

- Bài tập: Nhập vào từ bàn phím một số nguyên là năm dương lịch bất kỳ, in ra màn hình tên gọi theo năm âm lịch.
  - Ví dụ: 2021 có tên năm âm lịch là Tan Suu
  - Gợi ý:
    - Khởi tạo giá trị 2 mảng string:
      - Can = { "Canh", "Tan", "Nham", "Quy", "Giap", "At", "Binh", "Dinh", "Mau", "Ky" };
      - Chi = { "Than", "Dau", "Tuat", "Hoi", "Ty", "Suu", "Dan", "Meo", "Thin", "Ty", "Ngo", "Mui" };
    - Lấy phần dư phép chia số năm dương lịch cho 10 sẽ ra tên Can, cho 12 sẽ ra tên Chi.
    - Cú pháp để in giá trị ra màn hình
      - `Console.Write("{0} {1} {...}", <giá trị 0>, <giá trị 1>, <giá trị n>);`

# Mảng

- Thực hiện:

```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        int namDl;
        string[] Can = { "Canh", "Tan", "Nham", "Quy", "Giap", "At", "Binh", "Dinh", "Mau", "Ky" };
        string[] Chi = { "Than", "Dau", "Tuat", "Hoi", "Ty", "Suu", "Dan", "Meo", "Thin", "Ty", "Ngo", "Mui" };

        Console.Write(" Moi ban nhap mot nam bat ky: ");

        // Nhap vao nam Duong lich va ep kieu so nguyen
        namDl = Int32.Parse(Console.ReadLine());

        Console.WriteLine("Nam {0} co nam am lich la: {1} {2}", namDl, Can[namDl % 10], Chi[namDl % 12]);

        Console.ReadLine();
    }
}
```