



# LẬP TRÌNH .NET 1

# NỘI DUNG CHƯƠNG TRÌNH MÔN VISUAL STUDIO.NET (ASP.NET)

- Bài 1: Lập trình hướng đối tượng C#
- Bài 2: Lập trình Web với ASP.NET MVC 5
- Bài tập lớn cuối kỳ – bảo vệ

# NỘI DUNG BUỔI 1:

- GIỚI THIỆU VỀ NGÔN NGỮ C#
- NHẬP XUẤT TRONG C#
- CÁC CÂU LỆNH ĐIỀU KIỆN, CÂU LỆNH LẶP TRONG C#

# CẤU TRÚC CHƯƠNG TRÌNH C#

- Một chương trình C# gồm các phần sau:
- Khai báo Namespace
- Một class
- Phương thức của lớp
- Thuộc tính của lớp
- Một phương thức Main
- Lệnh và biểu thức
- Chú thích (Comment)

# Cấu trúc chương trình C#

```
using System;  
namespace Csharp  
{
```

Khai báo thư  
viện

Khai báo  
namespace

```
class TestCsharp  
{
```

Khai báo lớp

Hàm chính  
toàn chương  
trình

```
static void Main(string[] args)  
{  
    /* chương trình in dòng chữ Hello World trong C# */  
    Console.WriteLine("Hello World");  
    Console.ReadKey();  
}  
}
```

```
}
```

# CẤU TRÚC CHƯƠNG TRÌNH C#

- **Lưu ý:**
- C# là phân biệt kiểu chữ (case sensitive).
- Tất cả lệnh và biểu thức phải kết thúc với một dấu chấm phẩy (;).
- Sự thực thi chương trình bắt đầu tại phương thức Main.
- Để tạo chú thích trong chương trình C#, ta sử dụng:
  - dấu // nội dung chú thích,
  - /\*nội dung chú thích\*/

# KHAI BÁO VÀ SỬ DỤNG BIẾN

- **Cú pháp:**
- `<Kiểu dữ liệu> <Tên biến>;`
- Trong đó:
- `<Kiểu dữ liệu>` có thể là: Kiểu dữ liệu cơ bản, Kiểu dữ liệu có cấu trúc, . . .
- `<Tên biến>`: Là tên do người dùng đặt và phải tuân thủ theo quy tắc đặt tên
- Vd:
- `int BienKieuSoNguyen;`
- `float BienKieuSoThuc;`
- `string BienKieuChuoi;`

# XUẤT DỮ LIỆU TRONG C#

- **Console.Write(giá trị cần in ra màn hình);**
- Lệnh này dùng để in giá trị trong dấu ngoặc đơn ra màn hình. Giá trị có thể là 1 ký tự, chuỗi ký tự, số.
- **Console.WriteLine(giá trị cần in ra màn hình);**
- Lệnh này có tác dụng tương tự lệnh trên nhưng sau khi in giá trị ra màn hình thì hệ thống sẽ đặt con trỏ xuống dòng tiếp theo.
- **Ví dụ:** Console.Write("Xin chào \n");  
Console.WriteLine("Ban ten la gi");  
Console.Write("Ban bao nhieu tuoi");  
Console.Write(Environment.NewLine);  
Console.Write("Ban tu dau den");  
Console.ReadLine();



# XUẤT DỮ LIỆU TRONG C#

- *Lưu ý:*
  - Lệnh `Console.ReadLine()`; ở cuối chương trình để dừng màn hình chương trình lại. Nếu không chương trình sẽ chạy xong rồi tự tắt.
  - Đa số mọi người thường dùng lệnh `Console.WriteLine` để xuống dòng, rất ít người dùng `\n` hoặc `Environment.NewLine`

# NHẬP DỮ LIỆU TRONG C#

- Console.Read();

Lệnh này có tác dụng đọc 1 ký tự từ bàn phím và trả về kiểu số nguyên là **mã ASCII** của ký tự đó. Lệnh này không đọc được các phím chức năng như Ctrl, Shift, Alt, Caps Lock, Tab, .....

**Ví dụ** mình gõ **a** thì hệ thống sẽ trả về kết quả là **97**, gõ **A** thì trả về là **65**.

- Console.ReadLine();

Lệnh này có tác dụng đọc dữ liệu từ bàn phím cho đến khi xuống dòng (tức nghĩa là người dùng gõ **Enter** thì dừng).

# NHẬP DỮ LIỆU TRONG C#

- `Console.ReadKey(tham số kiểu bool);`  
Lệnh này cũng dùng để đọc một ký tự từ bàn phím nhưng trả về kiểu **ConsoleKeyInfo** (là một kiểu dữ liệu có cấu trúc được định nghĩa sẵn để chứa những ký tự của bàn phím bao gồm các phím chức năng). Tham số kiểu **bool** bao gồm 2 giá trị: **true** hoặc **false**. Nếu là **true** thì phím được ấn sẽ không hiển thị lên màn hình **console** mà được đọc ngầm. Nếu là **false** thì phím được ấn sẽ hiển thị lên màn hình **console**. Nếu để trống thì mặc định sẽ là **false**.
- Ví dụ:
- `Console.ReadKey();`  
`Console.ReadKey(false);`  
`Console.ReadKey(true);`

# BÀI TẬP THỰC HÀNH

- 1. Viết chương trình nhập vào 1 số nguyên và in số đó ra màn hình
- 2. Viết chương trình nhập vào 2 số nguyên và thực hiện hoán đổi giá trị 2 số nguyên đó
- Vd: nhập vào số thứ nhất: 2
- nhập vào số thứ hai: 7
- Giá trị số nguyên thứ nhất sau khi hoán đổi: 7
- Giá trị số nguyên thứ hai sau khi hoán đổi: 2
- 3. Viết chương trình tính chu vi và diện tích hình chữ nhật.

# CÂU LỆNH ĐIỀU KHIỂN

- Câu lệnh if
- Câu lệnh switch..case

# CẤU TRÚC IF ELSE DẠNG THIỂU

- **If ([Biểu thức điều kiện]) <Câu lệnh thực hiện>**
- If là từ khóa bắt buộc.
- <Biểu thức điều kiện> là biểu thức dạng boolean (trả về true hoặc false).
- <Câu lệnh thực hiện> là câu lệnh muốn thực hiện nếu <Biểu thức điều kiện> là đúng.
- **Ý nghĩa:**
- Nếu <Biểu thức điều kiện> trả về true thì thực hiện <Câu lệnh thực hiện> ngược lại thì không làm gì cả.

# CẤU TRÚC IF ELSE DẠNG THIỂU

Ví dụ:

```
static void Main(string[] args)
{
    int a, b; a = int.Parse(Console.ReadLine());
    b = int.Parse(Console.ReadLine());
    int m = a;
    if (m < b) m = b;
    Console.WriteLine("Max = " + m);
    Console.ReadLine();
}
```

# CẤU TRÚC IF ELSE DẠNG ĐỦ

- **If** <Biểu thức điều kiện>
- <Câu lệnh thực hiện 1>
- **else**
- <Câu lệnh thực hiện 2>
- **If, else** là từ khóa bắt buộc.
- <Biểu thức điều kiện> là biểu thức dạng boolean (trả về true hoặc false).
- <Câu lệnh thực hiện 1> là câu lệnh thực hiện nếu <Biểu thức điều kiện> là đúng.
- <Câu lệnh thực hiện 2> là câu lệnh thực hiện nếu <Biểu thức điều kiện> là sai.
- **Ý nghĩa:**
- Nếu <Biểu thức điều kiện> trả về true thì thực hiện <Câu lệnh thực hiện 1> ngược lại thì thực hiện <Câu lệnh thực hiện 2>.



# CẤU TRÚC IF ELSE DẠNG ĐỦ

Ví dụ:

- Console.WriteLine("Lenh if...else trong C#");
- Console.WriteLine("-----");
- /\* phan dinh nghia bien cuc bo \*/
- int a = 100;
- /\* kiem tra dieu kien \*/
- if (a > 20)
- {
- /\* neu dieu kien la true thi in dong sau: \*/
- Console.WriteLine("a nho hon 20");
- }
- else
- {
- /\* neu dieu kien la false thi in dong sau: \*/
- Console.WriteLine("a khong nho hon 20");
- }
- Console.WriteLine("Gia tri cua a la: {0}", a);
- Console.ReadLine();
- Console.ReadKey();

# BÀI THỰC HÀNH CÂU LỆNH IF

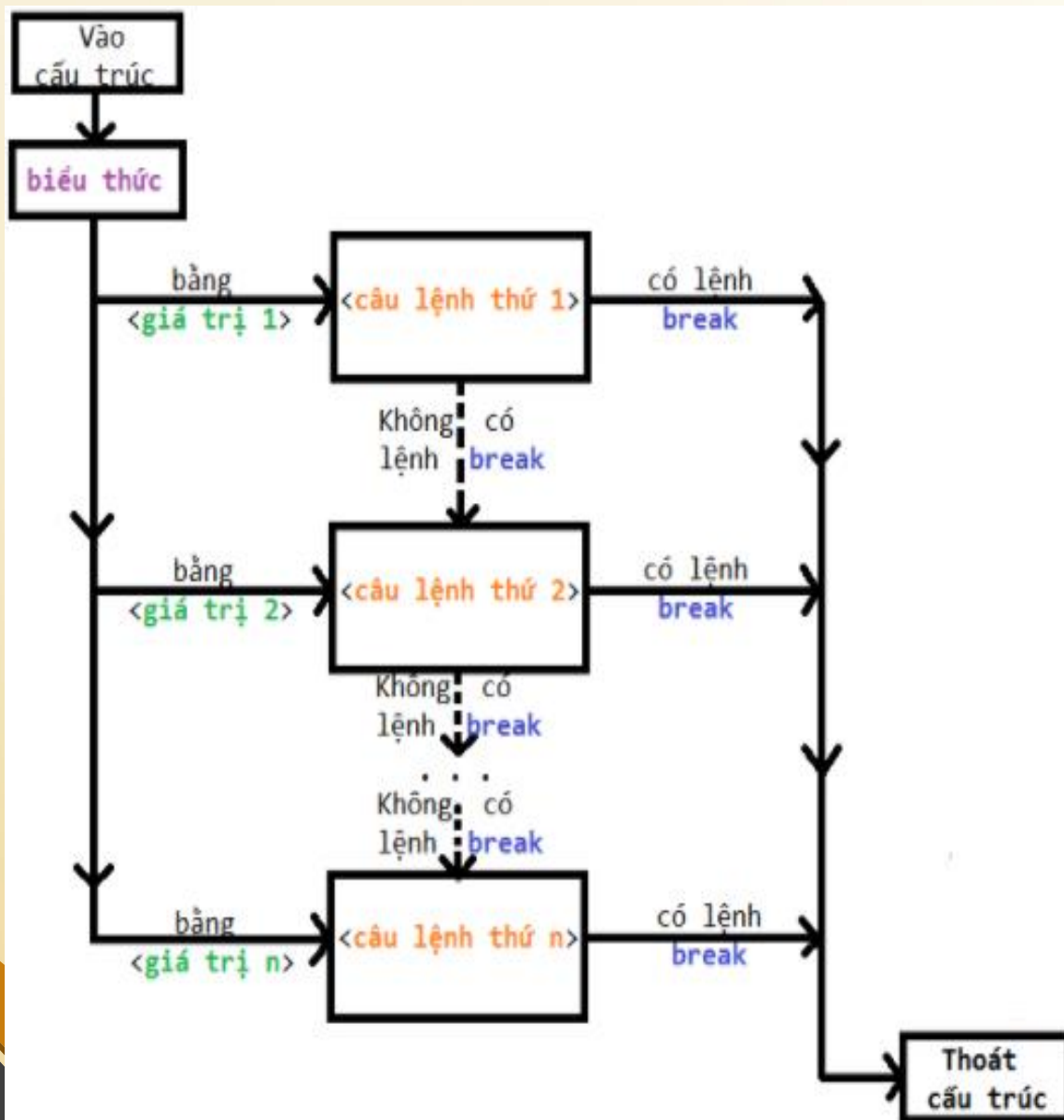
- 1. Viết chương trình kiểm tra số nguyên nhập vào là số chẵn hay số lẻ.
- 2. Viết chương trình tìm số lớn nhất trong 2 số nguyên được nhập vào từ bàn phím
- 3. Viết chương trình tìm số nhỏ nhất trong 3 số nguyên được nhập vào từ bàn phím
- 4. Viết chương trình giải phương trình bậc nhất  $ax+b=0$
- 5. Viết chương trình giải phương trình bậc 2:  $ax^2+bx+c=0$

# CÂU LỆNH SWITCH..CASE

- **Cấu trúc switch case dạng thiếu**
- **Cú pháp:**
- **switch** (<biểu thức>)
- {
- **case** <giá trị thứ 1>: <câu lệnh thứ 1>;
- **break**;
- **case** <giá trị thứ 2>: <câu lệnh thứ 2>;
- **break**;
- ...
- **case** <giá trị thứ n>: <câu lệnh thứ n>;
- **break**;
- }

**Ý nghĩa:** Duyệt lần lượt từ trên xuống dưới và kiểm tra xem giá trị của <biểu thức> có bằng với <giá trị thứ i> đang xét hay không. Nếu bằng thì thực hiện <câu lệnh thứ i> tương ứng.

# CÂU LỆNH SWITCH..CASE



Ví dụ:

```
int day = 4;
```

```
switch (day)
```

```
{
```

```
case 1:
```

```
    Console.WriteLine("Monday"); break;
```

```
case 2:
```

```
    Console.WriteLine("Tuesday"); break;
```

```
case 3:
```

```
    Console.WriteLine("Wednesday"); break;
```

```
case 4:
```

```
    Console.WriteLine("Thursday"); break;
```

```
case 5:
```

```
    Console.WriteLine("Friday"); break;
```

```
case 6:
```

```
    Console.WriteLine("Saturday"); break;
```

```
case 7:
```

```
    Console.WriteLine("Sunday"); break;
```

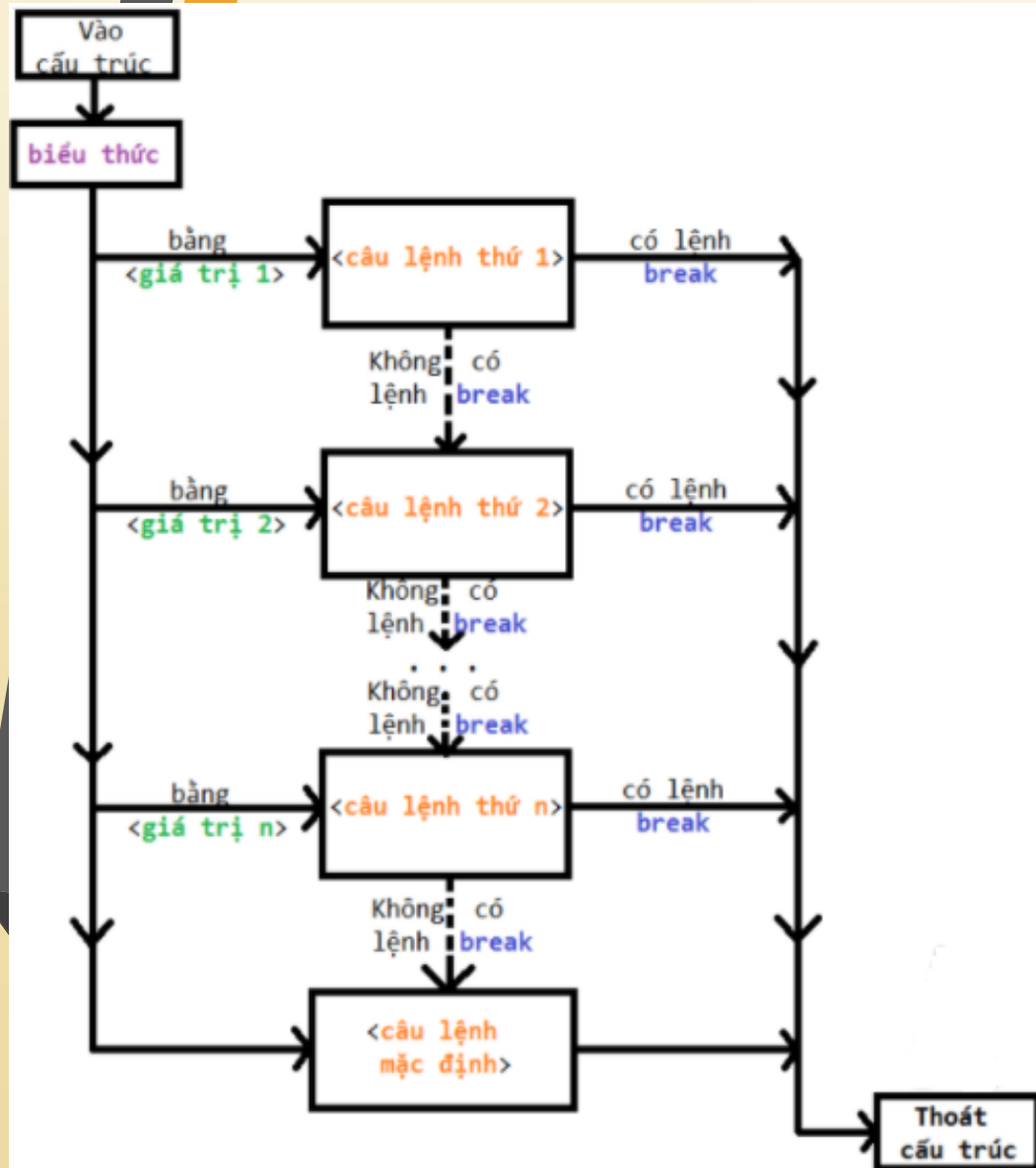
```
}
```

# CÂU LỆNH SWITCH..CASE

- **Cấu trúc switch case dạng đủ**
- **Cú pháp:**
- switch (<biểu thức>)
- {
- case <giá trị thứ 1>: <câu lệnh thứ 1>;
- break;
- case <giá trị thứ 2>: <câu lệnh thứ 2>;
- break;
- ...
- case <giá trị thứ n>: <câu lệnh thứ n>;
- break;
- default: <câu lệnh mặc định>;
- break;
- }

**Ý nghĩa:** Duyệt lần lượt từ trên xuống dưới và kiểm tra xem giá trị của <biểu thức> có bằng với <giá trị thứ i> đang xét hay không. Nếu bằng thì thực hiện <câu lệnh thứ i> tương ứng. Nếu không bằng tất cả các <giá trị thứ i> thì sẽ thực hiện <câu lệnh mặc định>.

# CÂU LỆNH SWITCH..CASE



- `Console.WriteLine("\nnhap vao mau sac yeu thich cua ban !!!");`
- `string color = Console.ReadLine();`
- `switch (color)`
- `{`
- `case "red":`
- `Console.WriteLine("ban thich mau do");`
- `break;`
- `case "yellow":`
- `Console.WriteLine("ban thich mau vang");`
- `break;`
- `case "pink":`
- `Console.WriteLine("ban thich mau hong");`
- `break;`
- `default:`
- `Console.WriteLine("mau ban khong thich mau do,`
- `mau vang va mau hong");`
- `break;`
- `}`

# CÂU LỆNH SWITCH..CASE

- **Lưu ý:**
- <giá trị thứ i> phải có kiểu dữ liệu giống với kiểu dữ liệu của giá trị của biểu thức.
- <câu lệnh thứ i> có thể gồm nhiều câu lệnh và không nhất thiết phải đặt trong cặp dấu ngoặc nhọn { } nhưng tốt hơn bạn nên đặt trong cặp dấu { } để code được rõ ràng hơn.
- Nếu case đang xét không rỗng (có lệnh để thực hiện) thì bắt buộc phải có lệnh nhảy (cụ thể là lệnh break) sau đó.
- Đối với case cuối cùng dù có câu lệnh để thực hiện hay không vẫn phải có lệnh break để thoát khỏi cấu trúc.

# BÀI THỰC HÀNH CÂU LỆNH SWITCH..CASE

1. Viết chương trình kiểm tra số nguyên nhập vào là số chẵn hay số lẻ
2. Viết chương trình nhập vào một tháng và cho biết tháng đó thuộc mùa nào trong năm.
3. Viết chương trình đổi năm dương lịch sang năm âm lịch

Năm dương % 10	0	1	2	3	4	5	6	7	8	9
Can	Canh	Tân	Nhâm	Quý	Giáp	Ất	Bính	Đinh	Mậu	Kỷ

Năm dương % 12	0	1	2	3	4	5	6	7	8	9	10	11
Can	Thân	Dậu	Tuất	Hợi	Tý	Sửu	Dần	Mẹo	Thìn	Tỵ	Ngọ	Mùi



# CÂU LỆNH LẶP FOR

- **Cú pháp:**
- **for** ([Khởi tạo]; [Điều kiện lặp]; [Bước lặp lại])
- {
- // Khối lệnh được lặp lại. Có thể bỏ trống
- }
- **Trong đó:**
- Các phần [Khởi tạo]; [Điều kiện lặp]; [Bước lặp lại] hoàn toàn có thể để trống nhưng phải có dấu chấm phẩy (;)
- Mỗi đoạn [Khởi tạo]; hay [Điều kiện lặp]; hay [Bước lặp lại] là một câu lệnh riêng.

# CÂU LỆNH LẶP FOR

- Ví dụ lệnh for:
- `for (int i = 0; i < 5; i++)`
- `{`
- `Console.WriteLine(i);`
- `}`

B1:  $i=0$ ,  $0<5$ , in 0, xuống hàng, tăng  $i$  lên 1 ( $i=1$ )

B2:  $1<5$ , in 1, xuống hàng, tăng  $i$  lên 1 ( $i=2$ )

B3:  $2<5$ , in 2, xuống hàng, tăng  $i$  lên 1 ( $i=3$ )

B4:  $3<5$ , in 3, xuống hàng, tăng  $i$  lên 1 ( $i=4$ )

B5:  $4<5$ , in 3, xuống hàng, tăng  $i$  lên 1 ( $i=5$ )

B6:  $5<5$ , sai điều kiện, kết thúc lệnh lặp

# CÂU LỆNH LẶP FOR

- **Lưu ý:**
- Khi viết thiếu các thành phần [Khởi tạo]; [Điều kiện lặp]; [Bước lặp lại] hết sức lưu ý để đảm bảo khởi tạo và lặp số lần hữu hạn, nếu không dễ dẫn tới vòng lặp vô tận, ví dụ đây là vòng lặp vô tận:
- `for (; ) {}` hoặc `for (; );`

# CÂU LỆNH LẶP FOREACH

- **Cú pháp**
- **foreach (<kiểu dữ liệu> <tên biến tạm> in <tên mảng hoặc tập hợp>)**
- {
- // Code xử lý
- }
- Trong đó:
  - foreach, in là từ khoá bắt buộc.
  - kiểu dữ liệu là kiểu dữ liệu của các phần tử trong mảng hoặc tập hợp.
  - tên biến tạm là tên 1 biến tạm đại diện cho phần tử đang xét khi duyệt mảng hoặc tập hợp.
  - tên mảng hoặc tập hợp là tên của mảng hoặc tập hợp cần duyệt.

# CÂU LỆNH LẶP FOREACH

- Nguyên lý hoạt động:
  - Vòng lặp đầu tiên, gán giá trị của phần tử đầu tiên trong mảng vào biến tạm.
  - Thực thi khối lệnh trong thân vòng lặp.
  - Tiếp tục gán giá trị của phần tử thứ 2 vào biến tạm rồi thực thi khối lệnh.
  - Cứ tiếp tục như vậy với phần tử thứ 3, 4, 5 ..... cho đến hết mảng thì kết thúc.
- Qua nguyên lý hoạt động, ta có thể thấy:
  - **foreach** duyệt tuần tự các phần tử trong mảng từ phần tử đầu tiên đến phần tử cuối cùng.
  - Tại một thời điểm, chỉ có thể thao tác với giá trị của phần tử đang xét mà không thể tương tác với các phần tử đứng trước nó hay đứng sau nó.
  - Không thể thay đổi giá trị của các phần tử vì lúc này giá trị của nó đã được sao chép ra một 1 biến tạm và ta chỉ có thể thao tác với biến tạm.
  - Nếu cố thay đổi giá trị của biến tạm sẽ gây ra lỗi.

# CÂU LỆNH LẶP FOREACH

- Ví dụ lệnh foreach:
- `string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};`
- `foreach (string i in cars)`
- `{`
- `Console.WriteLine(i);`
- `}`

# BÀI TẬP THỰC HÀNH LỆNH LẶP FOR

- 1. Viết chương trình in ra màn hình các số nguyên từ 1 đến n (n nhập vào từ bàn phím)
- 2. Viết chương trình tính tổng các số nguyên từ 1 đến n
- 3. Viết chương trình in ra màn hình như sau:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
..... N									
- N nhập vào
- 4. Viết chương trình kiểm tra số nguyên n có phải là số hoàn hảo hay không? N nhập vào từ bàn phím
- 5. Viết chương trình tính đếm có bao nhiêu số nguyên tố nằm trong khoảng từ 1 đến n, n nhập vào từ bàn phím

# CÂU LỆNH LẶP WHILE

- **Cú pháp:**
- **while (<Điều kiện lặp>)**
- **{**
- **// khối lệnh lặp lại**
- **}**
- **Điều kiện lặp** là một biểu thức logic bắt buộc phải có với kết quả trả về bắt buộc là true hoặc false.
- Từ khóa while biểu thị đây là một vòng lặp while. Các câu lệnh trong khối lệnh sẽ được lặp lại đến khi không còn thỏa mãn **điều kiện lặp** sẽ kết thúc vòng lặp while.



# CÂU LỆNH LẶP WHILE

- Ví dụ về lệnh while:
- `int i = 0;`
- `while (i < 5)`
- `{`
- `Console.WriteLine(i);`
- `i++;`
- `}`

- **BÀI TẬP THỰC HÀNH LẬP WHILE**  
Viết chương trình in màn hình các số nguyên chẵn từ 1 đến n (n nhập vào)

- 2. Viết chương trình tính tổng các số nguyên lẻ nằm trong khoảng từ 1 đến n
- 3. Viết chương trình in ra dãy số fibonacci sau:  
1                    1            2            3            5            8            ....            N (N nhập vào, các số trong dãy số này  $\leq N$ )
- 4. Viết chương trình in ra màn hình như sau:

- 
- The diagram is divided into two main sections by a vertical line. The left section shows a single character 'X' at the top, followed by 'XX', 'XXX', and 'XXXX' at the bottom. A vertical double-headed arrow spans the height of the 'X' and 'XX' characters. The right section shows 'XXXX' at the top, followed by 'XXX', 'XX', and 'X' at the bottom. A vertical double-headed arrow spans the height of the 'XXXX' and 'XXX' characters. In the center, between the two sections, is the text 'H chiều cao hình, được nhập vào từ bàn phím' (Height of the shape, entered from the keyboard). At the bottom of each section, there is a long string of 'X's: 'XXXXX' on the left and 'XXXXXXXXXX' on the right.

# CÂU LỆNH LẶP DO..WHILE

- **Cú pháp:**
- **do**
- {
- // khối lệnh lặp lại
- } **while (<Điều kiện lặp>);**
- **Điều kiện lặp** là một biểu thức logic bắt buộc phải có với kết quả trả về bắt buộc là true hoặc false.
- Từ khóa do while biểu thị đây là một vòng lặp do while. Các câu lệnh trong khối lệnh sẽ được lặp lại đến khi không còn thỏa mãn điều kiện lặp sẽ kết thúc vòng lặp do while.
- **Lưu ý:** vòng lặp do while sẽ thực hiện câu lệnh trong khối code xong rồi mới kiểm tra điều kiện lặp. Cuối vòng lặp do while có dấu ; ở cuối.

# CÂU LỆNH LẶP DO..WHILE

- Ví dụ lệnh do..while
- `int i = 0;`  
do {
- `Console.WriteLine(i);`
- `i++; } while (i < 5);`

# BÀI TẬP THỰC HÀNH LỆNH LẶP DO..WHILE

- 1. Viết chương trình in ra màn hình các số nguyên chẵn nằm trong khoảng từ 1 đến n (n nhập vào)
- 2. Viết chương trình in ra màn hình như sau:

● X

XXXXX

# X

● XX

XXX

H chiều cao hình, được  
nhập vào từ bàn phím

H chiều cao hình, được  
nhập vào từ bàn phím

● XXX

XX

XXXXXX

● XXXX

X

XXXXXXX

# LỆNH BREAK

- Trong **C#**, lệnh break dùng để ngắt vòng lặp (for, while, do-while) hoặc câu lệnh switch-case.

- **Ví dụ:**

- for (int i = 1; i <= 30; i++)
  - {
    - if (i == 20)
      - {
        - break;
- Console.WriteLine(" Gia tri thu {0}", i);
- }

- Console.ReadLine();

# LỆNH CONTINUE

- Trong C# lệnh continue được dùng để bỏ qua 1 lần lặp nào đó và tiếp tục thực hiện các lần lặp kết tiếp cho đến khi hết vòng lặp.

- Ví dụ:

- ```
for (int countRow = 1; countRow < 10; countRow++)  
{  
    if (countRow == 6)  
    {  
        continue;  
    }  
    Console.Write("Hang thu {0}:", countRow);  
    for (int countCol = 1; countCol <= 10; countCol++)  
    {  
        Console.Write("{0, 5}", countCol);  
    }  
    Console.WriteLine();  
}
```

# Câu hỏi ôn tập

• Câu 1: Điền vào chỗ trống sao cho in giá trị biến  $i < 6$  ra màn hình

• `int i = 1;`

• `_____ (i __ 6)`

• `{`

• `Console.WriteLine(i);`

•

`_____ ;`

• `}`

• Câu 2: Điền vào chỗ trống để dừng vòng lặp khi  $i=5$

• `for (int i = 0; i < 10; i++)`

• `{`

• `if (i == 5)`

• `{`

•

`_____ ;`

• `}`

• `Console.WriteLine(i);`

• `}`



# Câu hỏi ôn tập

- Câu 3: Điền vào chỗ trống để nhảy đến giá trị kế tiếp khi i đạt giá trị là 4

- for (int i = 0; i < 10; i++)

- {

- if (i == 4)

- {

- \_\_\_\_\_ ;

- }

- Console.WriteLine(i);

- }

- Câu 4: Điền vào chỗ trống bằng cách sử dụng vòng lặp do..while để in giá trị biến  $i < 6$  ra màn hình

- int i = 1;

- \_\_\_\_\_ ,

- {

- Console.WriteLine(i);

- \_\_\_\_\_ ;

- } (i \_\_\_\_ 6);

# Câu hỏi ôn tập

- Câu 5: Điền vào chỗ trống để in ra màn hình 5 chữ Yes trên 5 dòng
- `for(int i = 0; i < 5; ____)`
- `{`
- `Console.WriteLine("Yes");`
- `}`
- Câu 6: Hoàn thành đoạn code sau:
- `int day = 2;`
- `switch (____)`
- `{`
- `____1: Console.WriteLine("Monday"); break;`
- `____2: Console.WriteLine("Tuesday"); ____ ;`
- `}`

# Câu hỏi ôn tập

• Câu 6: Điền vào chỗ trống để in ra chữ Yes khi  $x=y$ , ngược lại in chữ No

• `int x = 50;`

• `int y = 50;`

• `_____ (x _____y)`

• `{`

• `Console.WriteLine("Yes");`

• `}`

• `_____ ,`

• `{`

• `Console.WriteLine("No");`

• `}`

Câu 7: Điền vào chỗ trống để in số 1 khi  $x=y$ , in số 2 khi  $x>y$  , còn lại in số 3

`int x = 50;`

`int y = 50;`

`_____ (x y)`

`{`

`Console.WriteLine("1");`

`}`

`_____ (x y)`

`{`

`Console.WriteLine("2");`

`}`

`_____ { Console.WriteLine("3");`

`}`