

## Bài 3

# Cấu trúc rẽ nhánh và vòng lặp

# Mục tiêu

- Câu lệnh điều kiện
- Vòng lặp
- Lệnh nhảy

# Các câu lệnh điều kiện

- Các câu lệnh điều kiện cho phép bạn chuyển hướng thực thi code của chương trình theo tình huống (Conditional statements) với mục đích làm cho khả năng thực thi của chương trình được linh .
- Ứng với mỗi tình huống khác nhau sẽ có những cách giải quyết khác nhau, các câu lệnh này bao gồm một số dạng như: `if ...`; `if ... else ...`; `if...else...if`; `switch ...` (Các câu lệnh dạng này còn được gọi với cụm từ : “Cấu trúc rẽ nhánh”).

# Câu lệnh “if”

if(condition)

{

Các câu lệnh

}

Ví dụ

```
int a = 10;  
if (a % 2 == 0)  
    Console.WriteLine("{0} la so chan", a);
```

# Câu lệnh “if...else”

- Cú pháp

if(condition)

{

các câu lệnh

}

else

{

các câu lệnh

}

```
int a = 10;  
if (a % 2 == 0)  
    Console.WriteLine("{0} la so chan", a);  
else  
    Console.WriteLine("{0} la so le", a);
```

# Câu lệnh “if...else...if”

- Cú pháp

if(condition\_1)

{các câu lệnh}

else if(condition\_2)

{các câu lệnh}

else if(condition\_3)

{các câu lệnh}

...

else

{các câu lệnh}

```
double diem = 5.6;
if (diem < 5)
    Console.WriteLine("Xep loai yeu");
else if (diem < 7)
    Console.WriteLine("Xep loai trung binh");
else if (diem < 9)
    Console.WriteLine("Xep loai kha");
else
    Console.WriteLine("Xep loai gioi");
```

# Câu lệnh if lồng nhau

- Cú pháp chung

```
if(condition)
{
    //một hoặc nhiều câu lệnh if
    if(condition)
    {
        //một hoặc nhiều câu lệnh if
        if(điều_kiện)
        {...}
    }
}
```

```
double diem = 5.6;
if (diem >= 0 && diem <= 10)
{
    if (diem < 5)
        Console.WriteLine("Xep loai yeu");
    else if (diem < 7)
        Console.WriteLine("Xep loai trung binh");
    else if (diem < 9)
        Console.WriteLine("Xep loai kha");
    else
        Console.WriteLine("Xep loai gioi");
}
else
    Console.WriteLine("Diem nhap sai");
```

# Câu lệnh “switch..case”

- Cú pháp

```
switch(expression)
{
    case value1:
        các lệnh
        break;
    case value2 :
        các lệnh
        break;
    ...
    case valuen :
        các lệnh
        break;
    default:
        các lệnh
}
```

```
int thu = 5;
switch (thu)
{
    case 2:
        Console.WriteLine("Thu 2"); break;
    case 3:
        Console.WriteLine("Thu 3"); break;
    case 4:
        Console.WriteLine("Thu 4"); break;
    case 5:
        Console.WriteLine("Thu 5"); break;
    case 6:
        Console.WriteLine("Thu 6"); break;
    case 7:
        Console.WriteLine("Thu 7"); break;
    case 8:
        Console.WriteLine("Chu nhat"); break;
    default:
        Console.WriteLine("Sai thu"); break;
}
```



# Vòng lặp

- Vòng lặp cho phép bạn thực thi một hoặc nhiều lệnh lặp đi lặp lại trong một điều kiện được xác định, thường thì cấu trúc lặp trong các ngôn ngữ lập trình hiện đại chia làm 4 dạng:



# Câu lệnh “for”

- Cú pháp

for(initialize; condition ; increment/decrement)

{

statements

}

```
for (int i = 1; i <= 10; i++)  
{  
    Console.WriteLine(i);  
}
```

# Câu lệnh “do...while”

- Cú pháp

```
do  
{  
    statements  
}while(condition);
```

```
int a = 0;  
do  
{  
    Console.Write("Nhap 1 so trong khoang 10-100:");  
    a = Convert.ToInt32(Console.ReadLine());  
} while (a < 10 || a > 100);  
Console.WriteLine("\n So vua nhap la a={0}", a);
```

# Câu lệnh “while”

- Cú pháp

```
while(condition)
{
    một hoặc nhiều câu lệnh
}
```

```
int i = 10;
while (i > 0)
{
    Console.WriteLine(i);
    i--;
}
```

# Câu lệnh “foreach”

- Cú pháp

foreach(data\_type variable\_name in array)

{

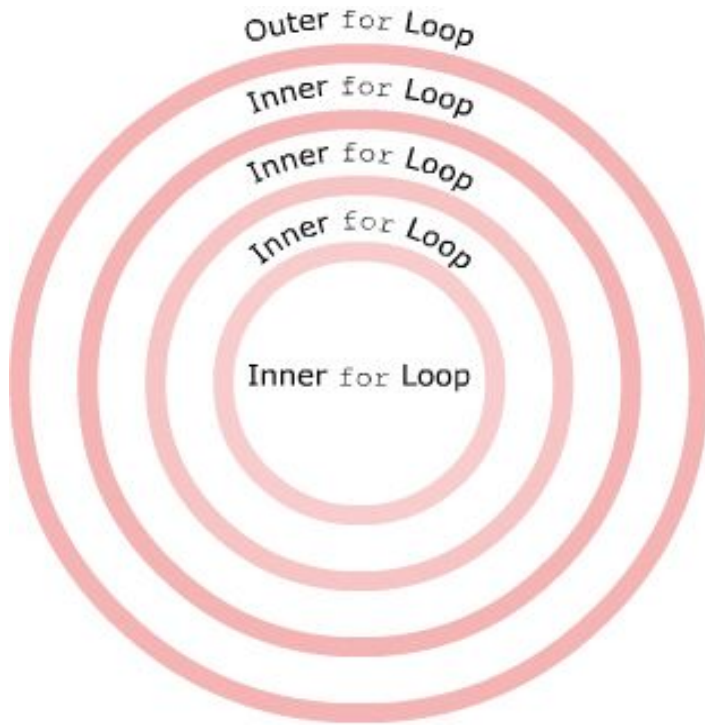
một hoặc nhiều câu lệnh

}

```
string[] names = { "Lan", "Dung", "Diep", "Hang", "Anh", "Tu" };  
foreach (string name in names)  
{  
    Console.WriteLine(name);  
}
```

# Vòng lặp lồng nhau

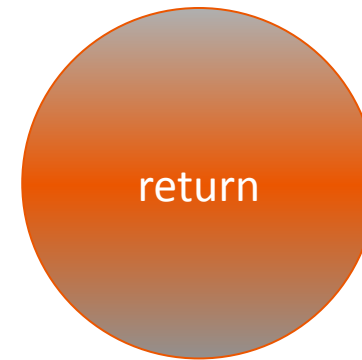
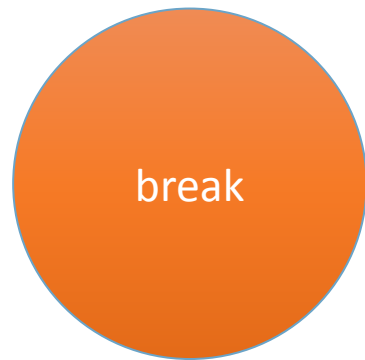
- Các vòng lặp được đặt lồng vào nhau theo từng mục đích sử dụng



```
for (int i = 2; i < 10; i++)  
{  
    Console.WriteLine("Bang cuu chuong so {0}", i);  
    for (int j = 1; j <= 10; j++)  
    {  
        Console.WriteLine("{0}*{1}={2}", i, j, i * j);  
    }  
}
```

# Các lệnh nhảy


- Các câu lệnh nhảy được sử dụng để chuyển điều khiển chương trình từ một vị trí này tới một vị trí khác
- C# hỗ trợ 4 câu lệnh nhảy



# Câu lệnh “break”

- Câu lệnh break được sử dụng trong các cấu trúc lặp và cấu trúc switch, khi gặp lệnh break, điều khiển chương trình sẽ thoát ra khỏi cấu trúc đó và thực hiện các lệnh tiếp theo

```
for(initialize; condition ; increment/decrement)
{
    ....
    if(điều_kiện_đúng)
        break;
    ....
}
```



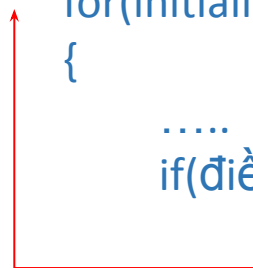
Thoát khỏi vòng lặp



# Câu lệnh “continue”

- Câu lệnh continue được sử dụng trong các cấu trúc lặp, khi gặp lệnh này máy sẽ kết thúc lần lặp hiện tại và chuyển điều khiển chương trình tới vị trí bắt đầu của vòng lặp để thực hiện lần tiếp theo

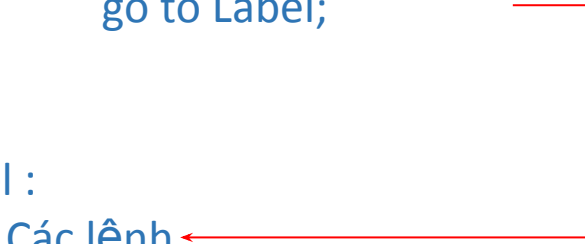
```
for(initialize; condition ; increment/decrement)
{
    ....
    if(điều_kiện_đúng)
        continue;
    ....
}
```



# Câu lệnh “go to”

- Câu lệnh này chuyển điều khiển thực thi chương trình tới vị trí được gán nhãn:

```
if(condition)
{
    go to Label;
}
....
Label :
    Các lệnh
```

A red line with an arrowhead at the end illustrates the jump. It starts from the right side of the 'go to Label;' statement, extends horizontally to the right, then turns 90 degrees downward, and finally turns 90 degrees leftward to point at the 'Label :' statement.

# Câu lệnh “return”

- Câu lệnh return sẽ trả về giá trị của một biểu thức, nó thường sử dụng trong các phương thức để trả về giá trị khi phương thức được thực thi
- Cú pháp:

`return (expression);`

```
static int Add(int a, int b)
{
    return (a + b);
}
```

# HỎI ĐÁP





# TRẢI NGHIỆM THỰC HÀNH