



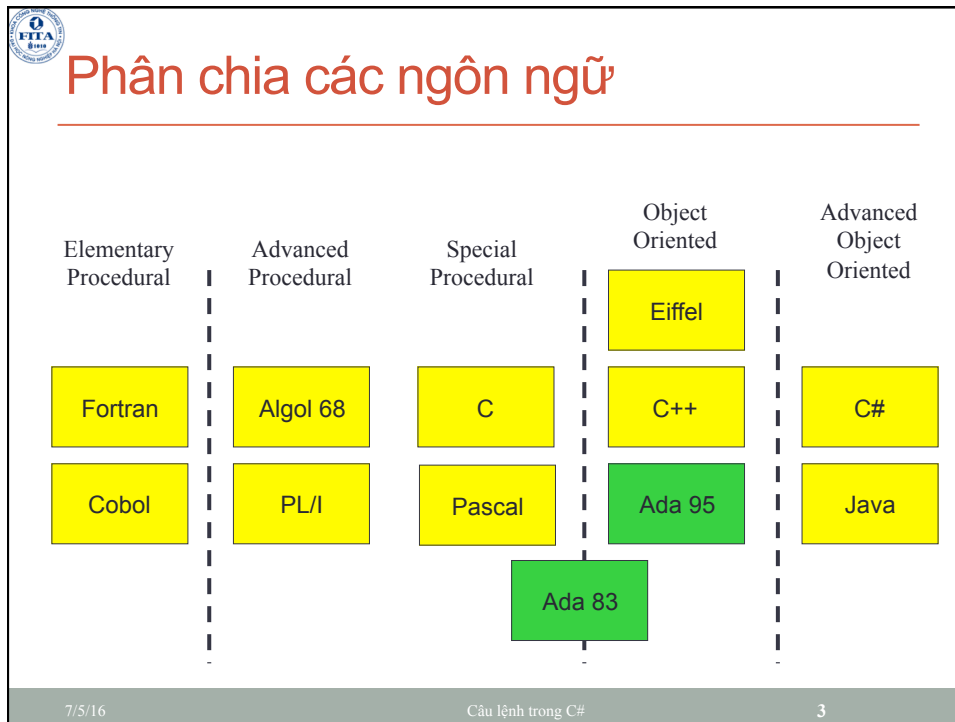
CÂU LỆNH TRONG C#

PHAN TRỌNG TIỀN
BM Công nghệ phần mềm
Khoa Công nghệ thông tin, VNUA
Email: phantien84@gmail.com
Website: <http://timoday.edu.vn>



Nội dung

- ☐ Cấu trúc chương trình C#
- ☐ Kiểu dữ liệu
- ☐ Khai báo biến
- ☐ Các toán tử trong C#
- ☐ Câu lệnh kiểm tra điều kiện C#
- ☐ Câu lệnh lặp C#
- ☐ Câu lệnh xử lý ngoại lệ try-catch C#



Ví dụ chương trình "Hello world"

```

class Hello
{
    static void Main()
    {
        // Use the system console object
        System.Console.WriteLine("Hello, World!");
    }
}

```

- ❑ Tạo một đối tượng kiểu class: **Hello**
- ❑ Chứa một phương thức: **Main**
- ❑ Trong **Main** chứa một dòng code, hiển thị "Hello, World!"
- ❑ Phương thức thực thi hành động này là : WriteLine
- ❑ Từ khoá "static" có nghĩa rằng phương thức Main được gọi không cần phải tạo bản sao của đối tượng này. Nó mà một phương thức của class, không phải phương thức của đối tượng
- ❑ Dòng bắt đầu bằng // là chú thích, không được biên dịch bởi C#

7/5/16 Câu lệnh trong C# 4



Câu lệnh trong C#

- ☐ Phân biệt chữ hoa chữ thường
- ☐ Không sử dụng các khoảng trắng trong các khai báo
- ☐ Sử dụng dấu chấm phẩy (;) để kết thúc câu lệnh
- ☐ Sử dụng dấu mở ngoặc, đóng ngoặc {} để chứa một khối lệnh
- ☐ Chú thích trong C#:
 - ☐ /* một khối lệnh*/
 - ☐ // một dòng lệnh

7/5/16

Câu lệnh trong C#

5



Các kiểu dữ liệu C#

- ☐ Kiểu dữ liệu hệ thống:
 - ☐ Reference: object, string
 - ☐ Signed: sbyte, short, int, long
 - ☐ Unsigned: byte, ushort, uint, ulong
 - ☐ Character: char (2 byte, Unicode)
 - ☐ Floating-point: float, double, decimal
 - ☐ Logical: bool
- ☐ Tên các kiểu dữ liệu này (dạng dễ nhớ) có tên tham chiếu tới các kiểu trong hệ thống
 - ☐ Ví dụ: int = System.Int32

7/5/16

Câu lệnh trong C#

6



Kiểu dữ liệu số

sbyte: -128 : 127

byte: 0 : 255

short: -32768 : 32767

ushort: 0 : 65535

int: -2147483648 : 2147483647

uint: 0 : 4294967295

long: -9223372036854775808 : 9223372036854775807

ulong: 0 : 18446744073709551615

float: -3,402823E+38 : 3,402823E+38

double: -1,79769313486232E+308 : 1,79769313486232E+308

decimal: -79228162514264337593543950335 :
79228162514264337593543950335

7/5/16

Câu lệnh trong C#

7



Kiểu dữ liệu mảng

```
using System;

class ArrayStringDemo{
    public static void Main(){
        string[] a1,
                a2 = {"a", "bb", "ccc"};

        a1 = new string[]{"ccc", "bb", "a"};

        int[,]    b1 = new int[2,4],
                b2 = {{1,2,3,4}, {5,6,7,8}};

        double[][] c1 = { new double[]{1.1, 2.2},
                           new double[]{3.3, 4.4, 5.5, 6.6} };

        Console.WriteLine("Array lengths. a1:{0} b2:{1} c1:{2}",
                           a1.Length, b2.Length, c1.Length);

        Array.Clear(a2,0,3);

        Array.Resize<string>(ref a2,10);
        Console.WriteLine("Lenght of a2: {0}", a2.Length);

        Console.WriteLine("Sorting a1:");
        Array.Sort(a1);
        foreach(string str in a1) Console.WriteLine(str);
    }
}
```

7/5/16

Câu lệnh trong C#

8



Các kiểu dữ liệu hệ thống

☐ Kiểu tham trị (value)

- ☐ Chứa dữ liệu trực tiếp
- ☐ Không thể là null
- ☐ Cư trú bộ nhớ stack

i 123

☐ Kiểu tham chiếu (reference)

- ☐ Có thể tham chiếu tới các đối tượng (**object**)
- ☐ Có thể null
- ☐ Cư trú bộ nhớ heap

s → "Hello world"

7/5/16

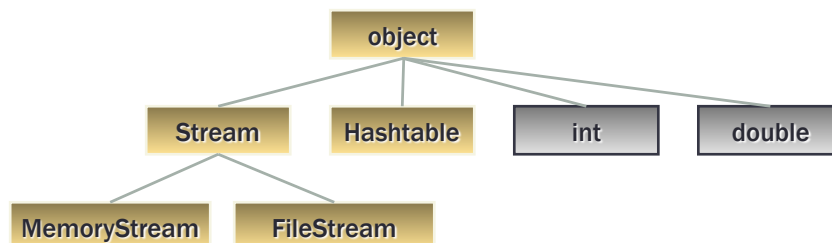
Câu lệnh trong C#

9



Đồng nhất kiểu dữ liệu C#

- ☐ Tất cả các kiểu dữ liệu trong C# đều thừa kế từ đối tượng **object**: classes, enums, arrays, delegates, structs, ...
- ☐ Có sự chuyển đổi ngầm định từ một kiểu dữ liệu bất kỳ tới kiểu **object**



7/5/16

Câu lệnh trong C#

10



Đồng nhất kiểu dữ liệu C# (Boxing)

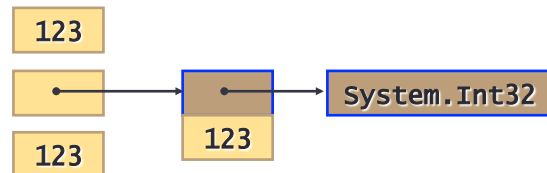
❑ Boxing

- ❑ Xử lý chuyển đổi một kiểu giá trị tới kiểu **object**
- ❑ Giá trị được lưu trữ bởi đối tượng System.Object và được quản lý trên bộ nhớ heap
 - ❑ Bạn có thể tượng tượng nó như một cái hộp (box), rồi copy giá trị tới nó

❑ Unboxing

- ❑ Trích xuất giá trị từ object
- ❑ Kiểm tra kiểu của box, copy giá trị tới nó

```
int i = 123;
object o = (object)i;
int j = (int)o;
```



7/5/16

Câu lệnh trong C#

11



Khai báo biến

- ❑ Kiểu_dữ_liệu tên_biến [= giá_trị_khởi_tạo]

❑ Ví dụ:

```
int number = 0;
string name;
float myFloat = 0.5f;
Bool hotOrNot = true;
//Khai báo hằng số
const int MAX_VALUE = 10000000;
```

- ❑ Các biến phải được khởi tạo hoặc khởi gán trước khi sử dụng
- ❑ Khai báo là hằng số thì giá trị trong hằng không thể thay đổi

7/5/16

Câu lệnh trong C#

12



Kiểu liệt kê (Enum)

```
enum identifier [:base-type]
{enumerator-list}
```

Ví dụ:

```
enum Grades
{
    gradeA = 94;
    gradeAminus = 90;
    gradeBplus = 87;
    gradeB = 84;
}
```

- ❑ Kiểu dữ liệu cơ sở (base-type có thể là bất kỳ kiểu dữ liệu integer (**ushort**, **long**) ngoại trừ kiểu **char**)
- ❑ Mặc định là kiểu **int**
- ❑ Phải ép kiểu (cast) tới kiểu **int** để hiển thị trong câu lệnh **WriteLn**
 - ❑ Ví dụ: **(int)g.gradeA;**

7/5/16

Câu lệnh trong C#

13



Các toán tử trong C#

Level	Category	Operators	Associativity (binary/tertiary)
14	Primary	x.y f(x) a[x] x++ x-- new typeof checked unchecked default delegate	left to right
13	Unary	+ - ! ~ ++x --x (T)x true false sizeof	left to right
12	Multiplicative	* / %	left to right
11	Additive	+ -	left to right
10	Shift	<< >>	left to right
9	Relational and Type testing	< <= > >= is as	left to right
8	Equality	== !=	left to right
7	Logical/bitwise and	&	left to right
6	Logical/bitwise xor	^	left to right
5	Logical/bitwise or	 	left to right
4	Conditional and	&&	left to right
3	Conditional or	 	left to right
2	Null coalescing	??	left to right
1	Conditional	?:	right to left
0	Assignment or Lambda expression	= *= /= %= += -= <<= >>= &= ^= = =>	right to left

7/5/16

Câu lệnh trong C#

14



Câu lệnh điều kiện if

```
if (expression)
    statement1
[else
    statement2]
```

Ví dụ:

```
if (i < 5) {
    System.Console.WriteLine("i là lớn hơn 5");
} else {
    System.Console.WriteLine("i là nhỏ hơn hoặc bằng 5");
}
```

- ☐ C# có cấu trúc câu lệnh if giống C/C++/Java
- ☐ Biểu thức phải được định giá tới một giá trị bool
- ☐ == là toán tử so sánh bằng trong biểu thức
if(l == 5) và if(l = 5) câu lệnh nào đúng?

7/5/16

Câu lệnh trong C#

15



Ví dụ câu lệnh if

```
/* Right, Wrong */
using System;

class IfDemo {
    public static void Main(){
        int i = 0;

        /*
        if (i){
            Console.WriteLine("i is regarded as true");
        }
        else {
            Console.WriteLine("i is regarded as false");
        }
        */

        if (i != 0){
            Console.WriteLine("i is not 0");
        }
        else {
            Console.WriteLine("i is 0");
        }
    }
}
```

7/5/16

Câu lệnh trong C#

16



Câu lệnh Switch

```
switch (expression)
{
    case constant-expression:
        statement(s);
        jump-statement
    [default: statement(s);]
}
```

- ❑ Có thể thay thế bằng câu lệnh **if**
- ❑ Dùng câu lệnh **break** để kết thúc một **case**
- ❑ Có thể dùng câu lệnh **goto** để tiếp tục một **case** khác

Ví dụ

```
const int raining = 1;
const int snowing = 0;
int weather = snowing;
switch (weather) {
    case snowing:
        System.Console.WriteLine("It is snowing!");
        goto case raining;
    case raining:
        System.Console.WriteLine("I am wet!");
        break;
    default:
        System.Console.WriteLine("Weather OK");
        break;
}
```

7/5/16

Câu lệnh trong C#

17



Ví dụ câu lệnh switch

```
/* Right, Wrong */
using System;

class SwitchDemo {
    public static void Main(){
        int j = 1, k = 1;

        /*
        switch (j) {
            case 0: Console.WriteLine("j is 0");
            case 1: Console.WriteLine("j is 1");
            case 2: Console.WriteLine("j is 2");
            default: Console.WriteLine("j is not 0, 1 or 2");
        }
        */

        switch (k) {
            case 0: Console.WriteLine("m is 0"); break;
            case 1: Console.WriteLine("m is 1"); break;
            case 2: Console.WriteLine("m is 2"); break;
            default: Console.WriteLine("m is not 0, 1 or 2"); break;
        }

        switch (k) {
            case 0: case 1: Console.WriteLine("n is 0 or 1"); break;
            case 2: case 3: Console.WriteLine("n is 2 or 3"); break;
            case 4: case 5: Console.WriteLine("n is 4 or 5"); break;
            default: Console.WriteLine("n is not 1, 2, 3, 4, or 5"); break;
        }

        string str = "two";
        switch (str){
            case "zero": Console.WriteLine("str is 0"); break;
            case "one": Console.WriteLine("str is 1"); break;
            case "two": Console.WriteLine("str is 2"); break;
            default: Console.WriteLine("str is not 0, 1 or 2"); break;
        }
    }
}
```

7/5/16

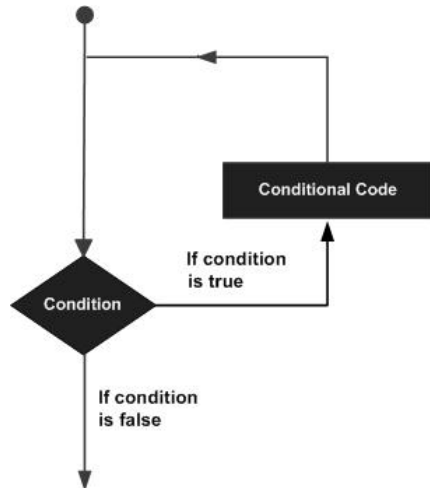
Câu lệnh trong C#

18



Câu lệnh lặp

- ❑ Dùng vòng lặp khi chúng ta cần thực thi một khối lệnh nào đó nhiều lần
- ❑ Các loại vòng lặp:
 - ❑ `while`
 - ❑ `for`
 - ❑ `do ... while`
 - ❑ `foreach`



7/5/16

Câu lệnh trong C#

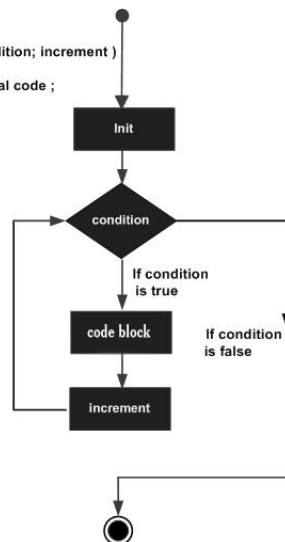
19



Câu lệnh for

```

for( init; condition; increment )
{
    conditional code ;
}
  
```



```

using System;

namespace Loops
{
    class Program
    {
        static void Main(string[] args)
        {
            /* for loop execution */
            for (int a = 10; a < 20; a = a + 1)
            {
                Console.WriteLine("value of a: {0}", a);
            }
            Console.ReadLine();
        }
    }
}
  
```

7/5/16

Câu lệnh trong C#

20

Câu lệnh while

```

while( condition )
{
    conditional code ;
}

```

```

using System;

namespace Loops
{
    class Program
    {
        static void Main(string[] args)
        {
            /* local variable definition */
            int a = 10;

            /* while loop execution */
            while (a < 20)
            {
                Console.WriteLine("value of a: {0}", a);
                a++;
            }
            Console.ReadLine();
        }
    }
}

```

7/5/16 Câu lệnh trong C# 21

Câu lệnh do ... while

```

do {
    conditional code ;
} while (condition)

```

```

using System;

namespace Loops
{
    class Program
    {
        static void Main(string[] args)
        {
            /* local variable definition */
            int a = 10;

            /* do loop execution */
            do
            {
                Console.WriteLine("value of a: {0}", a);
                a = a + 1;
            } while (a < 20);

            Console.ReadLine();
        }
    }
}

```

7/5/16 Câu lệnh trong C# 22



Câu lệnh foreach

```
class Program
{
    static void Main(string[] args)
    {
        int[] numbersArray = { 1,2,3,4,5,6,7,8,9};
        foreach (var item in numbersArray)
        {
            Console.WriteLine(item);
        }
        Console.ReadKey();
    }
}
```

7/5/16

Câu lệnh trong C#

23



Ví dụ câu lệnh foreach

```
/* Right, Wrong */
using System;

class ForeachDemo {
    public static void Main(){

        int[] ia = {1, 2, 3, 4, 5};
        int sum = 0;

        foreach(int i in ia)
            sum += i;

        Console.WriteLine(sum);
    }
}
```

7/5/16

Câu lệnh trong C#

24

Câu lệnh break

```

using System;

namespace Loops
{
    class Program
    {
        static void Main(string[] args)
        {
            /* local variable definition */
            int a = 10;

            /* while loop execution */
            while (a < 20)
            {
                Console.WriteLine("value of a: {0}", a);
                a++;
                if (a > 15)
                {
                    /* terminate the loop using break statement */
                    break;
                }
            }
            Console.ReadLine();
        }
    }
}

```

7/5/16 Câu lệnh trong C# 25

Câu lệnh continue

```

using System;

namespace Loops
{
    class Program
    {
        static void Main(string[] args)
        {
            /* local variable definition */
            int a = 10;

            /* do loop execution */
            do
            {
                if (a == 15)
                {
                    /* skip the iteration */
                    a = a + 1;
                    continue;
                }
                Console.WriteLine("value of a: {0}", a);
                a++;
            } while (a < 20);

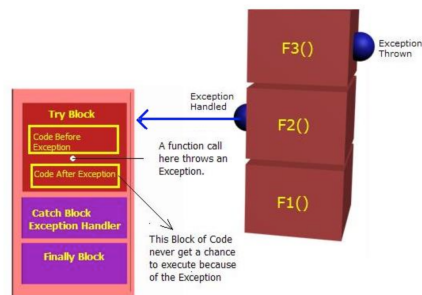
            Console.ReadLine();
        }
    }
}

```

7/5/16 Câu lệnh trong C# 26



Câu lệnh try-catch



```
/* Right, Wrong */
using System;

class TryCatchDemo {
    public static void Main(){
        int i = 5, r = 0, j = 0;

        /*
        r = i / 0;
        Console.WriteLine("r is {0}", r);
        */

        try {
            r = i / j;
            Console.WriteLine("r is {0}", r);
        } catch(DivideByZeroException e){
            Console.WriteLine("r could not be computed");
        }
    }
}
```