

# » BIẾN & CÁC PHÉP TOÁN

Name: Nguyễn Chí Hiếu

Date: 2020

## » NỘI DUNG

1. Biến và Hằng số

2. Các kiểu dữ liệu cơ sở

3. Biểu thức và Các phép toán

## » BIẾN VÀ HẲNG SỐ

### Biến (*variable*)

Là tên đại diện cho một vùng trong bộ nhớ máy tính. Mỗi biến được dùng để lưu trữ dữ liệu khi chạy chương trình.

```
<Kieu_du_lieu> <Ten_bien>;
```

```
1 int n = 5;  
2 double b = 10.5;  
3 char c = 'h';
```

## » BIẾN VÀ HẲNG SỐ

Khai báo (*define*) và khởi tạo (*declare*) biến

- \* Khai báo biến nhằm thực hiện ba thao tác chính: cấp vùng nhớ cho biến, đặt tên biến và quy định kiểu dữ liệu của biến.
- \* Biến có thể được khởi tạo giá trị ngay khi khai báo. Nếu không khởi tạo, ngôn ngữ C# sẽ gán một giá trị mặc định.

```
<Kieu_du_lieu> <Ten_bien>;  
<Kieu_du_lieu> <Danh_sach_bien>;  
<Kieu_du_lieu> <Ten_bien> = <Gia_tri>;
```

## » BIẾN VÀ HẲNG SỐ

Biến (*variable*)

```
1  int a, b;  
2  
3  a = 5;  
4  b = 10;  
5  
6  int c = a + b;  
7  Console.WriteLine("c = {0}", c);
```

## » BIẾN VÀ HẲNG SỐ

### Hằng số (*constant*)

Là một loại biến đặc biệt mà giá trị được gán khi vừa khai báo và không thay đổi khi chạy chương trình.

```
const <Kieu_du_lieu> <Ten_hang> = <Gia_tri>;
```

```
1 const double pi = 3.14;  
2 int r = 10;  
3 double c = 2 * r * pi;  
4  
5 Console.WriteLine("c = {0}", c);
```

## » CÁC KIỂU DỮ LIỆU CƠ SỞ

### Kiểu dữ liệu số nguyên

Ngôn ngữ C# cung cấp 8 kiểu dữ liệu số nguyên như sau:

Kiểu	Tên kiểu	Kích thước (byte)	Khoảng giá trị	Mặc định
sbyte	System.Sbyte	1	$-2^7 \rightarrow 2^7 - 1$	0
short	System.Int16	2	$-2^{15} \rightarrow 2^{15} - 1$	0
int	System.Int32	4	$-2^{31} \rightarrow 2^{31} - 1$	0
long	System.Int64	8	$-2^{63} \rightarrow 2^{63} - 1$	0L
byte	System.byte	1	$0 \rightarrow 2^8 - 1$	0
ushort	System.UInt16	2	$0 \rightarrow 2^{16} - 1$	0
uint	System.UInt32	4	$0 \rightarrow 2^{32} - 1$	0
ulong	System.UInt64	8	$0 \rightarrow 2^{64} - 1$	0

## » CÁC KIỂU DỮ LIỆU CƠ SỞ

Kiểu dữ liệu dấu chấm động (*floating point*)

Kiểu	Tên kiểu	Kích thước (byte)	Khoảng giá trị	Mặc định
float	System.Single	4	$\pm 1.5 \times 10^{-45}$ $\rightarrow \pm 3.4 \times 10^{38}$	0.0F
double	System.Double	8	$\pm 5.0 \times 10^{-324}$ $\rightarrow \pm 1.7 \times 10^{308}$	0.0D
decimal	System.Decimal	16	$\pm 1.0 \times 10^{-28}$ $\rightarrow \pm 7.9228 \times 10^{28}$	0.0M



## » CÁC KIỂU DỮ LIỆU CƠ SỞ

### Kiểu dữ liệu ký tự (*char*)

Kiểu	Tên kiểu	Kích thước (byte)	Khoảng giá trị	Mặc định
char	System.Char	16	U+0000 to U+FFFF	'\0'

\* Kiểu char có thể được mô tả bởi:

\* Ký tự.

\* Chuỗi Unicode gồm: `\u` và 4 ký tự hexa của bảng mã ASCII.

\* Chuỗi hexa: `\x` và 4 ký tự hexa của bảng mã ASCII.

## » CÁC KIỂU DỮ LIỆU CƠ SỞ

Kiểu dữ liệu ký tự (*char*)

```
1 char a = 't';  
2 char b = '\u0074';  
3 char c = '\x0074';  
4  
5 Console.WriteLine("a = {0}", a);  
6 Console.WriteLine("b = {0}", b);  
7 Console.WriteLine("c = {0}", c);
```

```
> a = t  
> b = t  
> c = t
```

## » CÁC KIỂU DỮ LIỆU CƠ SỞ

Chuyển đổi giữa các kiểu dữ liệu số

Ngôn ngữ C# cung cấp hai cách chuyển đổi kiểu dữ liệu:

- \* Không tường minh (*implicit*): được thực hiện mặc định bởi ngôn ngữ C#
- \* Tường minh (*explicit*): người lập trình dùng toán tử để ép kiểu dữ liệu

## » CÁC KIỂU DỮ LIỆU CƠ SỞ

### Ép kiểu tường minh

Gồm có hai loại

- \* Nới rộng (*widening*): từ kiểu dữ liệu nhỏ sang kiểu dữ liệu lớn hơn.
- \* Thu hẹp (*narrowing*): từ kiểu dữ liệu lớn sang kiểu dữ liệu nhỏ hơn  
→ *mất mát dữ liệu*.

```
1 byte b = 5;
2 long l = b; // long l = (long) b;
3 Console.WriteLine("b = {0}", b);
4 Console.WriteLine("l = {0}", l);
5
6 double d = 10.5;
7 int i = (int) d;
8 Console.WriteLine("d = {0}", d);
9 Console.WriteLine("i = {0}", i);
```

## » CÁC KIỂU DỮ LIỆU CƠ SỞ

Kiểu liệt kê (*enum*)

Là tập hợp các hằng số có cùng kiểu dữ liệu với nhau.

```
enum <Ten_bien>
{
    Ten_hang_1 = <Gia_tri_1>,
    Ten_hang_2 = <Gia_tri_2>,
    //...
    Ten_hang_n = <Gia_tri_n>
}
```

## » CÁC KIỂU DỮ LIỆU CƠ SỞ

Kiểu liệt kê (*enum*)

```
1  enum WeekDay
2  {
3      Monday, Tuesday, Wednesday, Thursday, Friday,
        Saturday, Sunday
4  }
5
6  static void Main(string[] args)
7  {
8      WeekDay day = WeekDay.Monday;
9      Console.WriteLine(day);
10 }
```

> Monday

## » CÁC KIỂU DỮ LIỆU CƠ SỞ

Kiểu liệt kê (*enum*)

```
1  enum Direction
2  {
3      Stop = 0, Left = 1, Up = 2, Right = 3, Down = 4
4  }
5
6  static void Main(string[] args)
7  {
8      Direction dir = Direction.Left;
9      Console.WriteLine(dir);
10 }
```

> Left

## » BIỂU THỨC

### Khái niệm

Trong ngôn ngữ lập trình, biểu thức (*expression*) là sự kết hợp giữa các toán hạng (*operand* và toán tử/phép toán (*operator*) để trả về một giá trị.

- \* Toán hạng: biến, hằng, hàm, ...
- \* Toán tử: phép toán số học, logic, ...

```
1  int a, b, c;  
2  
3  a = 5;  
4  b = 10;  
5  c = a + b;
```



## » CÁC PHÉP TOÁN

### Phân loại phép toán

Trong ngôn ngữ C#, các phép toán được chia thành các loại sau:

- \* Phép gán
- \* Phép toán số học
- \* Phép toán quan hệ và logic
- \* Phép toán trên bit

## » CÁC PHÉP TOÁN

### Phép toán gán

\* Gán giá trị cho một biến.

```
1 <ten_bien> = <gia_tri>;  
2 <ten_bien> = <ten_bien>;  
3 <ten_bien> = <bieu_thuc>;
```

```
1 int a, b, c;  
2  
3 a = 5;  
4 b = a + 5; // 10  
5 c = a = b; // c = ?
```

## » CÁC PHÉP TOÁN

### Phép toán số học

#### \* Phép toán 2 ngôi

Phép toán	Phép toán C#	Biểu thức toán học	Biểu thức C#
Cộng	+	$a + b$	<code>a + b</code>
Trừ	-	$a - b$	<code>a - b</code>
Nhân	*	$ab$	<code>a * b</code>
Chia	/	$\frac{a}{b}$	<code>a / b</code>
Chia lấy phần dư	%	$a \bmod b$	<code>a % b</code>

- \* *Phép chia: cần kiểm tra rằng buộ̣c số chia phải khác 0.*
- \* *Phép chia lấy phần dư: chỉ áp dụng đối với dữ liệu kiểu char, int, long.*

## » CÁC PHÉP TOÁN

### Phép toán số học

#### \* Phép toán 2 ngôi

```
1  int a, b;
2  a = 5;
3  b = 10;
4
5  Console.WriteLine("a/b = {0}", a / b);
6
7  Console.WriteLine("a/b = {0}", (double) a / b);
8  Console.WriteLine("a/b = {0}", a / (double) b);
9  Console.WriteLine("a/b = {0}", (double) a / (double) b);
10
11 Console.WriteLine("a%b = {0}", a % b);
```

## » CÁC PHÉP TOÁN

### Phép toán số học

\* Phép toán 1 ngôi (*các phép toán tăng giảm giá trị*)

Phép toán	Phép toán C#	Biểu thức C#
Tăng giá trị lên 1	++	a++ hay ++a
Giảm giá trị xuống 1	--	a-- hay --a

## » CÁC PHÉP TOÁN

### Phép toán số học

\* Phép toán 1 ngôi (*các phép toán tăng giảm giá trị*)

```
1  int a = 5;
2  a++;
3  Console.WriteLine("a = {0}", a);
4
5  int b = 5;
6  ++b;
7  Console.WriteLine("b = {0}", b);
```

## » CÁC PHÉP TOÁN

### Phép toán số học

\* Phép toán 1 ngôi (các phép toán tăng giảm giá trị)

```
1  int a = 5;
2  //a++;
3  Console.WriteLine("a = {0}", a++);
4
5  int b = 5;
6  //++b;
7  Console.WriteLine("b = {0}", ++b);
```

## » CÁC PHÉP TOÁN

### Các phép toán quan hệ

Biểu thức thực hiện phép toán quan hệ chỉ trả về chân trị đúng (*true*) hay sai (*false*).

Phép toán	Phép toán C#	Biểu thức toán học	Biểu thức C#
Lớn hơn	>	$a > b$	<code>a &gt; b</code>
Nhỏ hơn	<	$a < b$	<code>a &lt; b</code>
Bằng	==	$a = b$	<code>a == b</code>
Khác	!=	$a \neq b$	<code>a != b</code>
Lớn hơn hay bằng	>=	$a \geq b$	<code>a &gt;= b</code>
Nhỏ hơn hay bằng	<=	$a \leq b$	<code>a &lt;= b</code>



## » CÁC PHÉP TOÁN

### Các phép toán logic

Biểu thức thực hiện phép toán logic chỉ trả về chân trị đúng (*true*) hay sai (*false*).

Phép toán	Phép toán C#	Biểu thức toán học	Biểu thức C#
Phủ định	!	$\neg a$	!a
Và	&&	$a \wedge b$	a && b
Hay		$a \vee b$	a    b

## » CÁC PHÉP TOÁN

### Các phép toán logic

```
1  int a, b;  
2  bool kq;  
3  
4  a = 5;  
5  b = 10;  
6  
7  kq = (a < b) && (b % a == 0);  
8  Console.WriteLine(kq);
```

> True

## » CÁC PHÉP TOÁN

Thứ tự thực hiện các phép toán logic

Độ ưu tiên	Phép toán C#
1	!
2	&&
3	

## » CÁC PHÉP TOÁN

Thứ tự thực hiện các phép toán logic

Ví dụ

Cho biết giá trị trả về của các biểu thức sau:

a) `1 < 2 || 5 < 10 && 2 == 7`

b) `(1 < 2 || 5 < 10) && 2 == 7`

c) `1 < 2 && 5 < 10 || 2 == 7`

d) `1 < 2 && 5 < 10 || !(2 == 7)`

## » CÁC PHÉP TOÁN

### Phép toán trên bit

Phép toán	Phép toán trong C#	Biểu thức C#
NOT	$\sim$	$a \sim b$
AND	$\&$	$a \& b$
OR	$ $	$a   b$
XOR	$\wedge$	$a \wedge b$
SHIFT LEFT	$\ll$	$a \ll b$
SHIFT RIGHT	$\gg$	$a \gg b$

## » CÁC PHÉP TOÁN

### Phép toán trên bit

p	$\sim p$
0	1
1	0

### Ví dụ

Cho số nguyên dương  $n = 5 = (101)_2$ . Phép toán phủ định bit được tính như sau:

$$\sim n = \sim (101_2) = (010)_2 = 2_{10}.$$

## » CÁC PHÉP TOÁN

### Phép toán trên bit

p	q	$p \& q$	$p \mid q$	$p \wedge q$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

## » CÁC PHÉP TOÁN

Phép toán trên bit

Ví dụ

Cho số nguyên dương  $n = 5 = (101)_2$ .

\* Phép dịch trái 1 bit

$$n \ll 1 = (1010)_2 = 10.$$

\* Phép dịch phải 1 bit

$$n \gg 1 = (10)_2 = 2.$$



## » ĐỘ ƯU TIÊN CỦA CÁC PHÉP TOÁN

Độ ưu tiên	Phép toán C#	Kết hợp
1	[ ] ( ) . -> ++ - (postfix)	trái sang phải
2	sizeof & * + - ! ++ - (prefix)	phải sang trái
3	typecast	phải sang trái
4	* / %	trái sang phải
5	+ -	trái sang phải
6	<< >>	trái sang phải
7	< > <= >=	trái sang phải
8	&	trái sang phải
9	^	trái sang phải
10		trái sang phải
11	&&	trái sang phải
12		trái sang phải
13	? :	phải sang trái
14	= *= /= %= += -= «= »= &= =  =	phải sang trái
15	,	trái sang phải

## » ĐỘ ƯU TIÊN CỦA CÁC PHÉP TOÁN

```
1  double kq;  
2  
3  kq = 6 / 2 * (1 + 2);  
4  Console.WriteLine(kq);  
5  
6  double a, b;  
7  a = 5;  
8  b = 10;  
9  kq = a++ * --b / (b - a);  
10 Console.WriteLine(kq);
```

> 9

> 15

## » RÚT GỌN BIỂU THỨC

Một số luật logic

Phủ định của phủ định:

$$\neg\neg p \Leftrightarrow p$$

Luật DeMorgan:

\*

$$\neg(p \wedge q) \Leftrightarrow (\neg p) \vee (\neg q)$$

$$\neg(p \&\& q) \Leftrightarrow (!p) || (!q)$$

\*

$$\neg(p \vee q) \Leftrightarrow (\neg p) \wedge (\neg q)$$

$$\neg(p || q) \Leftrightarrow (!p) \&\& (!q)$$

## » RÚT GỌN BIỂU THỨC

### Ví dụ

Nhập một số nguyên dương  $n$ , kiểm tra  $n$  có thỏa một trong hai điều kiện sau:

- \*  $n$  là số lẻ
- \*  $n$  là số chẵn và thỏa tất cả các điều kiện sau:
  - \*  $0 < n < 100$
  - \*  $n$  không chia hết cho 6
  - \*  $n$  không chia hết cho 8

## » RÚT GỌN BIỂU THỨC

```
1  (  
2  (n % 2 != 0 || n > 0)  
3  && (n % 2 != 0 || n < 100)  
4  && (n % 2 != 0 || (!(n % 6 == 0 || n % 8 == 0))  
5  )
```

```
1  (  
2  (n % 2 != 0)  
3  || (n > 0 && n < 100 && !(n % 6 == 0 || n % 8 == 0))  
4  )
```

```
1  (  
2  (n % 2 != 0)  
3  || (n > 0 && n < 100 && n % 6 != 0 && n % 8 != 0)  
4  )
```

## » NHẬP XUẤT DỮ LIỆU

### Nhập dữ liệu

Sử dụng hàm `Read()` đọc một ký tự và `ReadLine()` đọc chuỗi ký tự nhập từ bàn phím (*chuỗi kết thúc bởi phím Enter*).

- \* Dữ liệu nhập vào cần chuyển sang kiểu dữ liệu phù hợp trước khi sử dụng
- \* Một số hàm dùng chuyển đổi kiểu dữ liệu:
  - \* `int.Parse(String)`, `double.Parse(String)`, ...
  - \* `Convert.ToInt32(String)`, `Convert.ToDouble(String)`, ...

## » NHẬP XUẤT DỮ LIỆU

### Nhập dữ liệu

Sử dụng hàm `Read()` đọc một ký tự và `ReadLine()` đọc chuỗi ký tự nhập từ bàn phím (*chuỗi kết thúc bởi phím Enter*).

```
1 char c1 = Console.Read(); // tra ve int
2 char c2 = (char) Console.Read();
3
4 int n = int.Parse(Console.ReadLine());
5
6 string s = Console.ReadLine();
```

## » NHẬP XUẤT DỮ LIỆU

### Xuất dữ liệu

Sử dụng hàm `Write()` hay `WriteLine()` để xuất chuỗi ký tự ra màn hình. Chuỗi ký tự có thể chứa các biến kết quả bằng những cách sau:

- \* Cách 1: sử dụng ký tự `+` để nối từng cặp chuỗi và giá trị.
- \* Cách 2: định dạng chuỗi để xuất nhiều giá trị bằng cách sử dụng lần lượt các ký tự `{0}`, `{1}`, ... tương ứng với biến `0`, biến `1`, ...
- \* Cách 3: thêm vào trước chuỗi ký tự dấu `$` và các biến đặt trong cặp dấu ngoặc `{}`

```
Console.Write("kq0=" + bien_0 + ",kq2=" + bien_1 + ...);
```

```
Console.Write("kq0={0},kq1={1}", bien_0, bien_1, ...);
```

```
Console.Write($"kq0={bien_0},kq1={bien_1}");
```



## » NHẬP XUẤT DỮ LIỆU

### Xuất dữ liệu

```
1  int a = int.Parse(Console.ReadLine());
2  double b = 10.0/3.0;
3  double c = a + b;
4
5  Console.Write("Ket qua:");
6  Console.WriteLine("a = " + a + ", b = " + b);
7  Console.WriteLine("c = a + b = {0}", c);
8  Console.WriteLine($"c = a + b = {c}");
```

## » NHẬP XUẤT DỮ LIỆU

### Xuất dữ liệu

Bảng 1: Bảng các mã định dạng trong C#.

Ví dụ 1	m = 1127		
Mã	Mô tả	C#	Kết quả
X	Hiển thị số dạng hexa	{0:X}	467
D	Số thập phân với tùy chọn	{0:D5}	01127
Ví dụ 2	n = 1127.1818		
Mã	Mô tả	C#	Kết quả
C	Tiền tệ	{0:C}	1,127.18
E	Số khoa học	{0:E2}	1.13E+003
F	Số thập phân với số lẻ xác định	{0:F3}	1127.182
N	Dấu phẩy ngăn cách phần ngàn	{0:N}	1,127.18
0:0...	Điền chữ số 0	{0:00000.00}	01127.18
0:0...	Điền khoảng trắng	{0:#####.##}	1127.18
%	Số thập phân dạng %	{0:00.00%}	112718.18%

## » BÀI TẬP

1. Viết chương trình nhập vào điểm học kỳ 1 và học kỳ 2 của một học sinh. In ra điểm trung bình cả năm với 2 số lẻ thập phân.
2. Cho một hình chữ nhật có chiều dài gấp 2 lần chiều rộng. Viết chương trình nhập vào chu vi của hình chữ nhật từ bàn phím, tính và in ra diện tích hình chữ nhật.
3. Viết chương trình nhập vào một ký tự c từ bàn phím. In ra mã ASCII của ký tự c và hai ký tự phía trước, phía sau của c.
4. Nhập vào một số nguyên dương n. In ra bit tại vị trí thứ i từ phải sang trái của n. Ví dụ:  $n = 27 = (11011)_2$ , bit thứ 2 từ phải sang trái là 1.
5. Rút gọn các biểu thức điều kiện sau:
  - a)  $!(x!=2 \ || \ x!=7)$
  - b)  $(!(x \ != \ 0 \ \&\& \ y \ / \ x < 1) \ || \ x == 0)$