

C#

§2: Kiểu dữ liệu, biến và biểu thức

1. Các kiểu dữ liệu cơ bản C#

Data Type	Description	Size(bits)	Range	Example
int	Whole numbers	32	-2^{31} through $2^{31} - 1$	<code>int nSize;</code>
long	Whole numbers (bigger range)	64	-2^{63} through $2^{63} - 1$	<code>long lSize;</code>
float	Floating-point numbers	32	$\pm 1.5 \times 10^{45}$ through $\pm 3.4 \times 10^{38}$	<code>float fDelta;</code>
double	Double-precision (more accurate) floating-point numbers	64	$\pm 5.0 \times 10^{324}$ through $\pm 1.7 \times 10^{308}$	<code>double dDelta;</code>
decimal	Monetary values	128	28 significant figures	<code>decimal decKe;</code>
string	Sequence of characters	16 bits per character	Not applicable	<code>string strName;</code>
char	Single character	16	0 through $2^{16} - 1$	<code>char chrAns;</code>
bool	Boolean	8	True or false	<code>bool bRet;</code>

2. Khái niệm về biến

- Biến** - để tên gọi cho 1 vùng nhớ có khả năng chứa 1 giá trị
- Để khai báo để chứa dữ liệu của Ct

Khai báo biến

Eq: kiểu biến = [giá trị]

- int age
- string chuan = "sin chao";

Chú ý:

- phân biệt hoa và thường
- k° chứa ký tự đặc biệt
- k° đặt biến trùng vs từ khóa

❖ Khi khai báo biến nên khởi tạo giá trị mặc định cho biến

❖ int nSize=0;

❖ String strName="";

```
namespace b3;
class Program
{
    static void Main(string[] args)
    {
        int soLuong = 0;
        double diemToan = 9.5;
        int x1 = 1;

        string name = "Hoàng Minh Hải";
        string ten = "Hải";

        //long long= 15, sai

        Console.OutputEncoding = System.Text.Encoding.UTF8;
        Console.WriteLine("điểm toán={0}", diemToan);
        Console.WriteLine("tên={0} , điểm={1} ", name, diemToan);

        Console.ReadLine();
    }
}
```

} khai báo main
Hàng viết
k° để trùng
lệnh
gửi

giống format

Ép kiểu (.) C#:

❖ Ép kiểu rộng: `int` → `long` → `float` → `double`
`double x = (double)1/2;` → `(double)1 = 1.0`
→ `1.0/2 = 0.5`
`double x = 1/2` ⇔ `int/int = 0` → `0.0`
❖ Ép kiểu hẹp: `double` → `float` → `long` → `int`
`int x = (int)1.0/2 = 0`

Implicit type C#:

C# khai báo biến mà k° cần khai báo

```
Console.WriteLine("Kiểu dữ liệu của t:{0} ", t.GetType());
```

Hằng số (.) C#:

`const + double` **PI** = 3.14;

→ tên hs pi viết hoa

Biểu thức

❖ Biểu thức là một phép toán đơn giản

`int x=5+3;`

❖ Biểu thức là một phép toán phức tạp

`int x=(5+3)*(4-2)/6`

❖ Biểu thức là sự kết hợp giữa các phép toán và các phương thức:

`double x=fn(9)*fn(8)+2-(8*2)+3`

S 4: phép toán C#

Toán tử gán:

Ký hiệu	Mô tả	Ví dụ
=	Gán toán tử hạng hai cho toán tử hạng nhất	<code>a = 1</code>
+=	Cộng hoặc nối chuỗi toán hạng sau vào toán hạng đầu và gán kết quả cho toán hạng đầu	<code>a += 1</code>
-=	Trừ toán hạng sau khỏi toán hạng đầu và gán kết quả cho toán hạng đầu.	<code>a -= 1</code>
*=	Nhân toán hạng sau vào toán hạng đầu và gán kết quả cho toán hạng đầu	<code>a *= 2</code>
/=	Chia toán hạng sau cho toán hạng đầu và gán kết quả cho toán hạng đầu	<code>a /= 2</code>

* Toán tử số học: `+`, `-`, `*`, `/`, `%`,
lấy phần dư

```
int a = 5;  
a += 2;  
Console.WriteLine("a={0}", a);  
Console.ReadLine();
```

```

int b;
Console.WriteLine("Enter a number b-->");
b = int.Parse(Console.ReadLine());
//console.WriteLine("nhập của bạn");
//nhập của bạn cần phải là số về 0
//using System;
if (b % 2 == 0)
    Console.WriteLine("{0} là số chẵn", b);
else
    Console.WriteLine("{0} là số lẻ", b);

int year;
Console.WriteLine("Enter a number of the year-->");
year = int.Parse(Console.ReadLine());
//kiểm tra năm nhuận là năm chia hết cho 4 nhưng không chia hết cho 100 hoặc 400
if (year % 4 == 0 && year % 100 != 0 && year % 400 != 0)
{
    Console.WriteLine("{0} là năm nhuận", year);
}
else
{
    Console.WriteLine("{0} không phải là năm nhuận", year);
}
Console.ReadLine();

```

Trong postfix và prefix

C++ sẽ th sau
++a , --b thì trê

int a=5, b=8, c=9;

int z=++a - --b + c++ -2;

B1) a=6, 7

B2) ( Độ ưu tiên giữa các toán tử

B3);

B4)

Thứ tự	Toán tử
1	Các toán tử đơn như: +, -, ++, --
2	Các toán tử số học *, /, %, -

Thứ tự	Toán tử
1	Các toán tử đơn như: +, -, ++, --
2	Các toán tử số học *, /, %, -
3	Các toán tử quan hệ >, <, >=, <=, ==, !=
4	Các toán tử luận lý &&, , ?:
5	Các toán tử gán =, *=, /=, +=, -=

§ 5 : Ghi chú lập trình

- tìm chuyên nghiệp
- hiểu để giải thuật

Ghi chú nhiều dòng :

```
/*  
 * cmt 2  
*/
```

Ghi chú dạng XML Document:

```
/// <summary>  
/// Write Comment 4  
/// </summary>  
/// <param name="a"></param>  
/// <param name="b"></param>  
private void func1(int a, float b)
```

```
1  /// <summary>  
2  /// Note dùng để tính to  
3  /// </summary>  
4  /// <param name="x">đầu x của int</param>  
5  /// <param name="y">đầu y của int</param>  
6  static void Main(int x, int y)  
7  {  
8  
9  static int f(int x, int y, int z)  
10     {  
11         return x+y+z;  
12     }  
13  
14     static void Main(string[] args)  
15     {  
16         int a = 5;  
17         // = là gộp this là my project  
18         // = hai giải về each notes  
19         // =  
20         int y = 8;  
21         Console.WriteLine("a={0}", a);  
22     }  
23 }
```

§ 5: type casting

type casting is converting a value to a different data type.

• Useful when we accept user input

• Different data types can do different things

```
double a = 3.14;
int b = Convert.ToInt32(a);

int c = 123;
double d = Convert.ToDouble(c)+0.1;

int e = 321;
String f = Convert.ToString(e);

String g = "$";
char h = Convert.ToChar(g);

String i = "true";
bool j = Convert.ToBoolean(i);

Console.WriteLine(b.GetType());
Console.WriteLine(d.GetType());
Console.WriteLine(f.GetType());
Console.WriteLine(h.GetType());
Console.WriteLine(j.GetType());
Console.ReadKey();
```

```
System.Int32
System.Double
System.String
System.Char
System.Boolean
```