



1

Hàm trong C#

2

Chú thích hàm

3

Truyền tham trị

4

Truyền tham chiếu(ref, out)

```
using System;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            SayHi("Mike");
            Console.ReadLine();
        }

        static void SayHi(string name) {
            Console.WriteLine("Hello " + name);
        }
    }
}
```



1

## Hàm trong C#

### ❑ 1. Khái quát về hàm :

- ❑ *Khi muốn thực thi một đoạn code nào đó nhiều lần, thay vì phải copy đi copy lại đoạn code đó, dẫn đến chương trình bị trùng lặp code rất => Khi đó ta sử dụng hàm*
- ❑ *Hàm là 1 khối lệnh thực hiện 1 công việc hoàn chỉnh (module)  
Hàm còn được gọi là chương trình con, phương thức, hành vi*
- ❑ *Hàm có thể gọi từ chương trình chính (hàm main) hoặc từ 1 hàm khác*
- ❑ *Hàm có 2 loại: Trả về giá trị (có **return** )  
- **Không trả về giá trị**: gọi là hàm thủ tục (procedure)*
- ❑ *Công dụng :*
  1. Chia nhỏ phân việc của dự án
  2. Tái sử dụng: khi cần chỉ cần gọi lại chương trình con mà o cần phải viết lại



1

## Hàm trong C#

### ❑ 2. Khai báo hàm :

```
<kiểu trả về> <Tên hàm> (<kiểu biến 1><tham số 1>,.....)
{
    Nội dung hàm;
    return <giá trị trả về>;
}
```

**Chú ý:** nếu là hàm thủ tục thì dùng **void** thay cho kiểu trả về

### ❑ 3. Đặt tên hàm:

- Giống quy tắc đặt tên biến
- Nên ghi tường minh (VD: *BinhPhuong()* , *TinhGiaiThua()* )



1

## Hàm trong C#

### ❑ 4 . Ví dụ (có return )

```
static int Tong(int x,int y, int z)
{ return x + y + z; }
```

```
static int TinhGiaiThua(int n)
{
    int gt = 1;
    for (int i = 1; i < n; i++)
        gt*=i;
    return gt;
}
```

### ❑ 5. Ví dụ (Hàm thủ tục, không có return)

```
//ví dụ static void
2 references
static void Xinchao(string m)
{
    if (m == "nam")
        Console.WriteLine("Xin chào, mình là boy");
    else if (m == "nu")
        Console.WriteLine("Xin chào, mình là girl");
}
```



2

## Chú thích hàm

### ❑ 6 . Chú thích hàm :

*Sử dụng **///** ngay trên hàm cần chú thích*

```
/// <summary>
/// Đây là hàm tính tổng 3 số
/// </summary>
/// <param name="x">nhập số nguyên x</param>
/// <param name="y">nhập số nguyên y</param>
/// <param name="z">nhập số nguyên z</param>
/// <returns>tra về tổng 3 số nguyên</returns>
1 reference
static int Tong(int x,int y, int z)
{ return x + y + z; }
```



3

## Truyền tham trị

❑ 7 . Truyền tham trị : *không thay đổi* giá trị biến sau khi gọi hàm

```
static void Thamtri(int a)
{
    a=a+1;
    Console.WriteLine("a trong hàm tham trị là: " + a);
}
```

```
//truyền tham trị
int a = 1;
Console.WriteLine("a trước khi gọi hàm truyền tham trị " + a);
Thamtri(a);
Console.WriteLine("a sau khi gọi hàm truyền tham trị " + a);
```

```
a trước khi gọi hàm truyền tham trị 1
a trong hàm tham trị là: 2
a trước khi gọi hàm truyền tham trị 1
```



## 4 Truyền tham chiếu(ref, out)

Có thay giá trị của biến sau khi gọi hàm

### ❑ 8 . Truyền tham chiếu (truyền tham biến) - ref :

```
static void ThamChieuRef(ref int b)
{
    b++;
    Console.WriteLine("b trong hàm tham chiếu ref là: " + b);
}
```

// ref **phải khởi tạo giá trị cho biến trước khi gọi hàm**  
//int b; // sẽ báo lỗi

```
int b= 2;
Console.WriteLine("b trước gọi hàm truyền tham chiếu ref " + b);
ThamChieuRef(ref b);
Console.WriteLine("b sau gọi hàm truyền tham chiếu ref " + b);
```

b trước khi gọi hàm truyền tham chiếu ref 2  
b trong hàm tham chiếu ref là: 3  
b sau khi gọi hàm truyền tham chiếu ref 3

### ❑ 9 . Truyền tham chiếu (truyền tham biến) - out :

// out **phải gán giá trị cho biến trước khi thoát khỏi hàm**

```
static void ThamChieuOut(out int c)
{
    c = 5;
    Console.WriteLine("c trong hàm tham chiếu ref là: " + c);
}
```

c trong hàm tham chiếu ref là: 5  
c sau khi gọi hàm truyền tham chiếu ref 5

```
int c;
ThamChieuOut(out c);
Console.WriteLine("c sau khi gọi hàm truyền tham chiếu ref " + c);
```

