

C#编程结构

(1) 参数修饰符

默认按值传递，调用者的原始值和被调用方法的变量之间的值互相不影响

out 按引用传递,输出参数由被调用方法赋值

ref 按引用传递,调用者赋初值 ,关键字行为根据参数是值类型还是引用类型有区别

按值传递引用类型： 被调用方法中可以修改对象的状态数据，但不能改变对对象的引用

按引用传递引用类型： 被调用方法中可以修改对象的状态数据，也可以改变对对象的引用

params **参数数组**

将一组可变数量的相同类型的参数作为单独的逻辑参数传递。能且只能是最后一个参数有 params 修饰符

```
static double Cal(params double[] values)
```

```
1, Cal(1.2,2.2,3.3); 2, double[] data = {1,2.2,3,3,3,3}; Cal(data); 3, Cal().
```

可选参数

方法调用者不指定不必要的参数，而是使用参数的默认值。

但是，可选参数的值必须在编译时确定且必须位于方法签名的最后。

DateTime.Now 的值在运行时确定

```
static double DefalutParam(string name, string param = "default param");  
DefalutParam("name"); DefalutParam("name","optional");
```

命名参数

允许在调用方法时乱序指定参数的值

```
static void DisplayFancyMessage(ConsoleColor textColor =  
ConsoleColor.Blue,  
ConsoleColor backgroundColor = ConsoleColor.White,  
string message = "Test Message"){...}  
DisplayFancyMessage(message: "example");
```

重载

一组名字相同，但是参数类型或是数量不同的方法。

(2) 数组

是一组相同数据类型 (int, string, class) 的数据点

var 隐式语法分配数组，初始化列表中每一项的类型应该是相同的

多维数组：矩形数组， 交错数组： 数组的上界不同

System.Array

Sort Algorithm(windows)

内省排序法 IntroSort()

≤ 3 swap 1/2, 1/3, 2/3

$3 < \leq 16$ 插入排序法 InsertionSort()

$\log_2 N > 0$ 先采用快速排序法 PickPivotAndPartition()

$\log_2 N = 0$ 再使用堆排序法 HeapSort()

(3) 枚举类型 enumeration

enum: 自定义的名/值对的数据类型

enumerator : 实现了 IEnumerable 的类或结构

```
enum EmpType { Low=10, Medium, Large} // 11,12,13
```

```
enum EmpType { Low=10, Medium=13, Large=15}
```

低层存储 enum EmpType : byte { Low=10, Medium=13, Large=15}

System.Enum

(4) 结构类型 struct

C#的 new 关键字会调用默认的构造函数，并且给每个字段赋默认值

结构类型的构造函数必须为每一个字段赋值

(5) 值类型和引用类型

System.ValueType 重载 System.Object 定义的虚方法。来实现值类型对值操作，复制或传递等

值类型复制后，两个变量的操作是独立的

引用类型复制后，都指向同一块内存，相互之间互相影响

包含引用类型的值类型 值类型中一旦包括了引用类型属性，引用类型属性字段会指向同一块内存，其他值类型属性字段的值仍然是互相独立的。

按值传递引用类型： 被调用方法中可以修改对象的状态数据，但不能改变对对象的引用

按引用传递引用类型： 被调用方法中可以修改对象的状态数据，也可以改变对对象的引用

(6) 可空类型

数值数据类型 (char, bool) 都是值类型,一旦实例化自动有默认值 (0, " , false) , 所以值类型不会有值 null

可空数据类型是值类型后加“ ?” , 定义域中增加 null

实际是创建一个泛型 System.Nullable, 提供了一组可空变量类型可以使用的成员集合, 派生自 struct

局部可空变量必须赋一个初始值或是 null 才能使用, `null == null => true`

可空类型赋值。 `if(!a.HasValue) a = true;`