

值传递和引用传递

值类型和引用类型

```
//值类型: int double char decimal bool enum struct(结构体)
//引用类型: string 数组 自定义类 集合 object 接口

//值类型存在栈上
//引用类型的值在堆上, 而栈中存着指向堆中的地址
```

传递方式

```
//值传递: 仅传递值的本身
//因此, 将n1传给n2后, 改变n1并不会对n2有影响

//引用传递: 传递值所在的地址(引用)
//因此, 两者在堆中就是同一块空间, 改变其一, 另一个也会跟着改变

//注: string和其他引用类型的不同之处:
//每次给string赋值时, 都会在堆中新开辟一片空间, 因此赋值的两个string不会指向同一块堆中地址
```

举例

```
//创建一个对象时, 程序执行的顺序:
//1.在堆中开辟一片空间
//2.在该空间上创建对象
//3.调用对象的构造函数

static void Main(string[] args){
    int number1 = 10;
    TestTwo1(number1);
    System.Console.WriteLine(number1); //结果为10, 因为TestTwo1中并没有返回结果

    int number2 = 10;
    TestTwo2(ref number2);
    System.Console.WriteLine(number2); //结果为20, 因为ref能把一个变量以参数的形式带到方法中运行, 并将运行后的值带出来

    //注: ref会将值传递变为地址(引用)传递
    //原本number2与n不是同一块栈地址,
    //但使用ref后, 会将number2的地址赋给n,
    //两者的地址相同, 因此n一变, number2也跟着变
}

public static void TestTwo1(int n){
```

```
        n += 10;
    }
    public static void TestTwo2(ref int n){
        n += 10;
    }
```