C# 中的结构体 (struct)

作者：Czyyyyy
原文链接：https://ld246.com/article/1604421918934
来源网站：链滴

Struct

声明方式

```
struct StructName{
    MemberDeclarations;
}
```

注意点

1. struct是值类型，而class是引用类型

2. struct在栈中分配空间，而class在堆中分配空间

3. struct类型的变量不能为null

```
struct Point{
    public int X;
    public int Y;
    public void Fun(){ }
}

class Program{
  static void Main(){
     //Point p = null; //编译不通过
     Point p = new Point(); //正确写法
  }
}
```

4. struct的两个变量不会/不能引用同一个对象

5.

把一个struct对象复制给另一个struct对象，就是将一个struct对象的值复制给另一个struct对象(深拷贝，对应的值指向不同的地址空间)。这和复制class变量不同，复制class变量时只复制引用。

6. 对于每个struct，都存在预定义的无参构造函数，且这个无参构造函数不能被删除或重定义。struct也可以有构造函数和静态构造函数(静态构造函数会在调用显示声明的构造函数时或引用结构的静态成员时被自动调用)，但不允许有析构函数。

7.

可以不使用new运算符传概念struct实例，如果这样做将会有如下限制：

1. 只有在显式地设置数据成员后才能使用它们的值：

```
  Point p;
  //Console.WriteLine(p.X);//编译不通过: Struct field 'p.X' might not be initialized before accessing
  p.X = 3;
  Console.WriteLine(p.X);  //√
```

2. 在对所有的数据成员赋值后，才能调用他们的函数成员

原文链接：C# 中的结构体 (struct)

```
Point p;
p.X = 3;
//p.Fun(); //编译不通过
p.Y = 0;
p.Fun();  //√  X 和 Y都赋值以后可以正常的调用了
```

8. 在声明struct时，不允许为字段赋默认值

```
struct Point
{
    public int X = 3; //编译错误,不允许的写法
    public int Y = 2; //编译错误，不允许的写法
}
```

9. struct是隐式密封的，不能被继承
10. struct作为返回值时，将创建它的副本并从函数成员返回
11. 当struct作为值参数时，将创建实参结构的副本。
12. 当struct用作ref或out参数时，传入的方法时该struct变量的一个引用。