

1 const 和 static readonly

2

3 二者本质的区别在于，const的值是在编译期间确定的，因此只能在声明时通过常量表达式指定◆

◆值。

4 而static readonly是在运行时计算出其值的，所以还可以通过静态构造函数来赋值。

5

6

7 明白了这个本质区别，我们就不难看出下面的语句中static readonly和const能否互换了：

8

9

10 1. static readonly MyClass myins = new MyClass();

11 -- 不可以换成const。new操作符是需要执行构造函数的，所以无法在编译期间确定

12

13

14 2. static readonly MyClass myins = null;

15 -- 可以换成const。我们也看到，Reference类型的常量（除了String）只能是Null。

16

17

18 3. static readonly A = B * 20;

19 static readonly B = 10;

20 -- 可以换成const。我们可以在编译期间很明确的说，A等于200。

21

22

23 4. static readonly int [] constIntArray = new int[] {1, 2, 3};

24 -- 不可以换成const。道理和1是一样的，虽然看起来1,2,3的数组的确就是一个常量。

25

26

27 5. void SomeFunction()

28 {

29 const int a = 10;

30 ...

31 }

32 -- 不可以换成readonly，readonly只能用来修饰类的field，不能修饰局部变量，也不能修饰property等◆

◆他类成员。

33

34

35 在通俗一点说, `const` 类型赋值必须是脱离系统运行时才能初始化的值(

36 `const page p=null` 正确,

37 `const page p= new Page()` 错误, 因为 `new Page()` 需要运行时才初始化

38)