C#程序设计及其应用 课堂作业06

1900094809 刘文琳；1900094810 尤俊浩；1900094813 汪蔚淓

**阅读dotnet 的referencesource源代码并简要回答以下问题。**

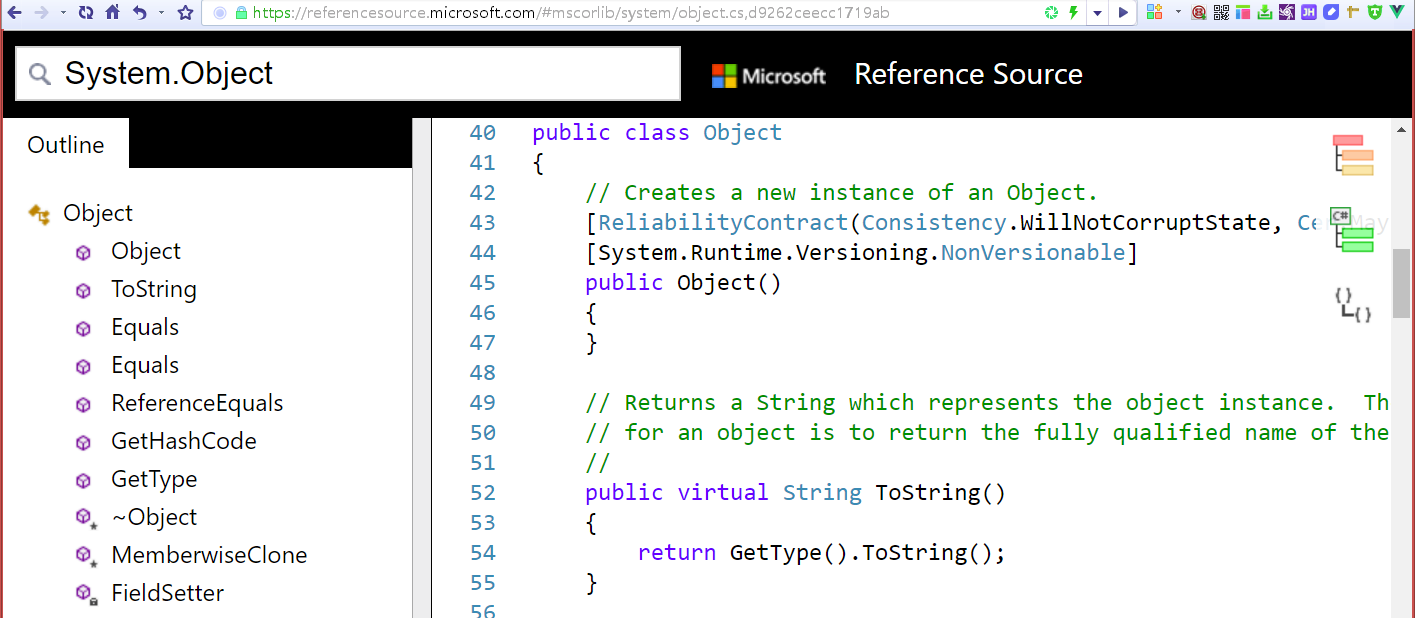
注：网址在（<https://referencesource.microsoft.com> ） 其右上角有下载download链接。

在其中mscorlib中有最基础的一些类，还有的在其他项目中(项目与命名空间有交叉关系)。

可以在其中输入关键词进行查找，如输入Convert或System.Convert

代码区右边有红色、绿色、灰色按钮可显示类/项目/命名空间的大纲。

用Ctrl+F 可以在页面内进行查找。



除此之外，你可以在这个网站上查看： <https://source.dot.net> 如：

<https://source.dot.net/#System.Private.CoreLib/String.cs>

-----System.Object类-----

1. 其ToString()的声明前的修饰符是什么？

public virtual

1. 其两个Equals()函数是什么关系，与ReferenceEquals()有没有差别？

重载关系。有差别，ReferenceEquals判断两个引用类型的对象是否是指向同一个地址，Equals则是判断两个object的值是否等价。

1. 其GetHashCode前面有个什么修饰语？

public virtual

1. 有没有Clone()函数？

MemberwiseClone()，是shallow copy

1. 其构造函数做了什么？有没有写析构函数？

构造函数内部什么也没做。但是有两个attributes [[ReliabilityContract](https://referencesource.microsoft.com/mscorlib/system/runtime/reliability/reliabilitycontractattribute.cs.html#937cd863ccbfbe47)([Consistency](https://referencesource.microsoft.com/mscorlib/system/runtime/reliability/reliabilitycontractattribute.cs.html#945a19cd67768190).[WillNotCorruptState](https://referencesource.microsoft.com/mscorlib/system/runtime/reliability/reliabilitycontractattribute.cs.html#e24b9a6439523417), [Cer](https://referencesource.microsoft.com/mscorlib/system/runtime/reliability/reliabilitycontractattribute.cs.html#0e851cd7ef098460).[MayFail](https://referencesource.microsoft.com/mscorlib/system/runtime/reliability/reliabilitycontractattribute.cs.html#197166b4cf47ad7e))]

[System.Runtime.Versioning.[NonVersionable](https://referencesource.microsoft.com/mscorlib/system/runtime/versioning/NonVersionableAttribute.cs.html#6a375d4b743be03e)]

有析构函数。

-----System.Type类-----

1. 它的构造方法前面有什么修饰符？

protected 受保护

1. 为什么说我们不能 new Type()?

Type的构造函数是受保护的，只限于本类和子类访问，因此不能在其他地方/实例直接new来进行访问

1. GetFields(), GetMethods(), GetProperties(), GetEvents()各有什么作用？

GetFields() 返回类定义的所有字段构成的数组。

GetMethods() 返回类实现的所有方法。

GetProperties() 返回Type定义的所有属性。

GetEvents() 返回 EventInfo的数组，如果没有定义过Events就返回一个空数组。

-----System.String类-----

1. String可以继承吗？

不能，前面有Sealed修饰符

1. 从哪里可以看出String是可以比较、可以克隆的？

String继承了IComparable以及ICloneable接口，因此必须在类内部实现。

System.String类里提供了不同参数类型的compare函数，同时也有System.StringComparer和Ststem.Comparison类协助进行。关于克隆，System.String类里也提供了Clone函数，返回类型为string对象。

1. Join()函数使用了什么样的保卫语句？(注：Contract.EndContractBlock(); // 此方法不做任何事情，只是让二进制编译器知道，此句以上的部分是契约。)

if (value==null)  
 throw new [ArgumentNullException](https://referencesource.microsoft.com/mscorlib/system/argumentnullexception.cs.html#8314938388254d5a)("value");

1. Join()方法中使用了StringBuilder对象，为什么不要在多次循环中使用+=？

使用+=来进行concat连接操作会创建新的string字符串，若要进行连接的字串过大，将导致性能严重下降，开销过大。因此使用StringBuilder可以有效解决这一问题，其不仅提供了可自由伸缩的缓冲区，同时也可有效获取已填充的大小。

1. 为什么说String是immutable的？其Replace、Append、Trim等方法是否改变了这个字符串自己？

因为String的对象一旦被创建好，它的状态将不会改变。没有改变，因为当使用这些方法试图“改变”一个String对象的时候，一个新的String对象副本将被创建。

1. SmallCharToUpper中如何将字符变大写，为何该函数要用unsafe的指针？

字符c -= 0x20，这是因为函数中需要使用指针来访问内存，但是C# 不保证指针操作的安全性，这方面需要程序员自己确保代码不会引起安全风险或者指针错误。

1. Equals方法是判断内容相等还是引用相等？为什么说字符串的= = 比较的是内容相等？compareTo是比较什么？

Equals方法判断的是内容相等。System.String里重载了==，因此字符串的==操作为取两个字符串对象的值来调用String.Equals()比较是否相等。CompareTo比较适用于排序或alphabetizing操作，返回值类型为int，返回值有三种类型，小于0、零或大于零，可以说比较的是大小。

1. 字符串中如何定义索引器的？

char [this](https://referencesource.microsoft.com/mscorlib/R/8307d03426b56fe1.html)[int index]，返回字符串中第index位的字符。

1. IsNullOrWhiteSpace是什么含义？

判断字符串是否为null或空字符或空白字符。

1. GetHashCode方法override是谁的方法，它使用指针有什么好处？

System.Object的方法。可以手动把循环展开，比如在32-bit 机器中将 char \* 指针转换成 int \* 指针，这样就能在 int ( 32 bits) 里存2个char ( 2 x 16 bits )了。这样可以提升效率。

1. ToDouble()调用了什么类的什么方法?

System.Convert类的ToDouble方法，将string类型强制转换为double类型。

-----System.Text.StringBuilder类-----

1. StringBuilder的初始内存是多少字符？

16 (internal const int DefaultCapacity =16)

1. Capacity属性的get和set方法有什么用，set是在原来的基础上扩展多少？

get是返回当前分配给StringBuilder对象所有Chunk的大小(capacity)。

也即m\_ChunkChars.Length + m\_ChunkOffset，也就是一个block有多少个字符加上在这个block前面的所有字符的数量，返回当前block最后一个字符的位置。

set是先检查当前值是不是超过MaxCapacity或者小于当前已分配的大小，是的话返回out of range exception。若没有异常，就定义新的数组，给它指定一个空间。char[] newArray = new char[newLen];

set扩展了newLen = value - m\_ChunkOffset，就是设置的 value 减去当前已存在的Chunk总共的字符数量，也即只扩展当前Chunk，并把扩展前Chunk的m\_ChunkChars个字符都Copy到新的空间。

1. Append()、Remove()等方法的返回值是什么，这有什么好处？

Append()和Remove()返回的是当前构建器本身的引用。

这样可以反复使用Append和Remove (比如 s.Append(...).Append(...)，而无需a.Append(...); a.Append(...);)。

-----System.Double-----

1. Double是struct还是class?它实现了哪些接口？

struct，实现了IComparable, IFormattable, IConvertible

public struct [Double](https://referencesource.microsoft.com/mscorlib/R/1a65cbdb09544ba1.html) : [IComparable](https://referencesource.microsoft.com/mscorlib/system/icomparable.cs.html#29a39323fbfa896b), [IFormattable](https://referencesource.microsoft.com/mscorlib/system/iformattable.cs.html#09da57c2c21a3a44), [IConvertible](https://referencesource.microsoft.com/mscorlib/system/iconvertible.cs.html#09e9f8f17849c7ce)

, [IComparable](https://referencesource.microsoft.com/mscorlib/system/icomparable.cs.html#3911354c40e0c30e)<[Double](https://referencesource.microsoft.com/mscorlib/system/double.cs.html#1a65cbdb09544ba1)>, [IEquatable](https://referencesource.microsoft.com/mscorlib/system/iequatable.cs.html#91a17479daaada86)<[Double](https://referencesource.microsoft.com/mscorlib/system/double.cs.html#1a65cbdb09544ba1)>

1. 它里面有几个字段?前面用了什么修饰词？

8个 (m\_value, MinValue, MaxValue, Epsilon, NegativeInfinity, PositiveInfinity, NaN, NegativeZero)

除了m\_value和NegativeZero，都是用public const double，m\_value使用了internal double的修饰符，NegativeZero用了internal static double的修饰词。

Double本身使用了public struct 修饰词。

1. 其常数MaxValue前面用了什么修饰词，其值是多少？

public const double，其值是1.7976931348623157E+308

1. CompareTo什么情况下返回-1，0及1?

当 value == null， 返回1。

当 m\_value < value 返回 -1

当m\_value > value 返回1

当m\_value == d 返回 0

用例子解释：

double d1 = a, d2 = b;

d1.CompareTo(d2) 会返回0当 a= b，返回1当a>b，返回-1当a<b。

1. Equals是override谁的方法？两个NAN是否相等？

System.ValueType的方法。Equals方法会判定相等，但是double类型的==运算符强制NaN不等于其他任何值。经调试，用Double.NaN调用Equals方法测试会返回true，但如果调用相等运算符==来测试则会返回false。

1. 如何实现==的运算符重载的？

[System.Runtime.Versioning.[NonVersionable](https://referencesource.microsoft.com/mscorlib/system/runtime/versioning/NonVersionableAttribute.cs.html#6a375d4b743be03e)]

public static bool operator ==([Double](https://referencesource.microsoft.com/mscorlib/system/double.cs.html#1a65cbdb09544ba1) left, [Double](https://referencesource.microsoft.com/mscorlib/system/double.cs.html#1a65cbdb09544ba1) right) {

return left == right;

}

1. GetHashCode是如何计算的？

double类型的hashcode是其bits在int中的表示，另需保证0和-0有同样的hashcode。

double d = [m\_value](https://referencesource.microsoft.com/mscorlib/system/double.cs.html#9aa58276ebdf906a);

if (d == 0) {

// Ensure that 0 and -0 have the same hash code

return 0;

}

long value = \*(long\*)(&d);

return unchecked((int)value) ^ ((int)(value >> 32));

-----System.Convert-----

1. Convert类前面为什么加个static？

因为Convert类是要让别的对象调用以转换类型的，不必有实例，所以修饰为静态类。

1. ToDouble(int value) 及 ToDouble(string value)内部分别调用了什么方法？

ToDouble(int value)直接返回value；在ToDouble(string value)里，若value不为空，则使用System.Double里的Parse解析函数解析该value后返回。

-----System.Math类-----

1. Abs(short)什么情况下会抛出导常？

value等于Int16.MinValue，由于Abs以后没有对应的最大值，所以会溢出并抛出异常。

1. 举几个方法是extern的方法。

Acos, Asin, Atan, Atan2, Ceiling, Cos, Cosh, Floor, Sin, Tan, Sinh, Tanh, Round, Sqrt, Log, Log10, Exp, Pow, Abs

-----System.Random类-----

1. 不带参数的构造函数，它用什么进行了初始化？

调用系统时钟来生成种子值。

1. 从Next（）来看为什么说它是伪随机?

Next()调用了InternalSample()，随机性是人工模拟的，函数里的逻辑依赖于SeedArray，这点可能造成其“随机”是有迹可循的，因此是伪随机的。

----- System.Array类-----

1. 它使用了一个什么内部类？

SZArrayEnumerator 和 FunctorComparer

1. SorterGenericArray()来看它使用哪两种主要的算法？

快速排序 DepthLimitedQuickSort 和内省排序 IntrospectiveSort

----- List<T>类-----

1. 从EnsureCapacity()可以看出每次容量增加多少倍？

若当前的容量小于min时，EnsureCapacity会增加其容量至现有容量的两倍(若当前容量是0则赋初值\_defaultCapacity）或min，取决于哪个更大。

----- Stack<T>类-----

1. push()与pop()进行了什么操作？

push()用来将元素排入栈的末尾，pop()用来将栈末尾的元素弹出。实际上，stack用数组来存放元素，在push()里时，若当前数组已满，则需要创建新的数组并将先前的所有元素拷贝至新数组后，再在末尾处存放新元素。而pop()里，若数组已为空，则报错，否则是取出数组的最后一个元素并返回该元素。

----- Hashtable类-----

1. Hashtable是如何定义索引器的？

this[key]与一个value一一对应。get 会返回 key 对应的 value，而 set 则会设置 key 对应的value。

具体来说，get要读出hashtable就先先算出hash，找出位置。先比较哈希码，如果不等于就结束。然后，对key进行比较，如果不等于就结束，否则就返回桶里的值。

----- WebClient类-----

1. 从GetWebRequest()来看，其WebClient与WebRequest是什么关系？

WebClient 包装了 WebRequest，因此可以说WebClient是WebRequest的高度抽象。这样我们可以省去中间很多繁琐的步骤，使得编程更为简便。

1. DownloadString()中调用了GetStringUsingEncoding()，后者是如何猜测网页的字符编码的？

后者取得WebRequest里的内容后判断其ContentType后对其解析并取得该内容的编码。如果取编码失败，则从 UTF8,、UTF32、Unicode、BigEndianUnicode中猜一个，再猜不中则将编码设置为系统的编码。

-----其他阅读中学习到的，或阅读心得 -----

感觉专业程序员的源代码有几处非常值得我们学习：

1）注释：许多地方即使还没细看，单看注释就明白这个函数是具体想做什么、怎么实现、思想细节、边界条件等。

2）命名法：源代码里的变量基本都是有意义的，十分方便阅读与理解代码。很多函数只需要看名字就知道是做什么的，这样阅读代码的时候也不用那么费劲。

3）多使用函数：每个函数的代码都不是太长，一眼看上去就能看到整体脉络，这都有赖于其把功能性代码都包装成函数。同时，也使得代码的维护变得更加简单。

4）保卫语句：源代码使用了保卫语句，使得代码整体不会因为多层if else 而变得不可读。

5）面向对象：面向对象的编程方式，使得类通过继承有了可拓展性。

6）包装：源代码把许多思想都不断的包装，使得其高度抽象化，这样使得编程变得简单。

7）效率：源代码会使用许多小的trick，提升代码的效率，比如一些unsafe的指针操作等。