常量

在C#中，基元类型都可以用于定义常量。同时，非基元类型也允许被定义为常量变量，但前提是，把值设为null。

常量总被视为静态成员而非实例成员，定义常量将导致创建元数据。代码引用常量符号时，编译器咋定义常量的程序集的元数据中查找该符号，提取常量的值，将值嵌入IL代码中。由于常量的值直接嵌入代码，所以在运行时不需要为常量分配任何内存。

常量的缺点（以下缺点都来源于“常量的值直接嵌入代码”）：不能获取常量的地址、不能以传引用的方式传递常量（这使得常量不能很好地支持跨版本跨程序集的版本控制）。每次常量需要改变值，都需要重新编译。

字段

字段是一种数据成员，其中容纳了一个值类型的实例或则一个引用类型的引用。

字段有以下几种修饰符：

图片包含 文本

描述已自动生成

Ps：readonly除了可以在构造器方法写入，还可以使用反射修改。

字段存储在动态内存RAM中，它们的值在运行时才能获取。字段可以是任何数据类型。

静态字段在生成类型对象时分配空间，实例字段在生成实例时分配空间。

Readonly不可改变的是引用而非字段引用的对象。

举个例子：

文本

描述已自动生成

这个只读的数组的引用是不可以被改变的，但是这个数组中的每个下标的元素都是可以修改的。