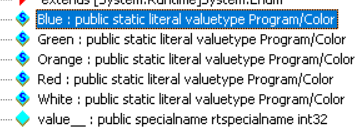
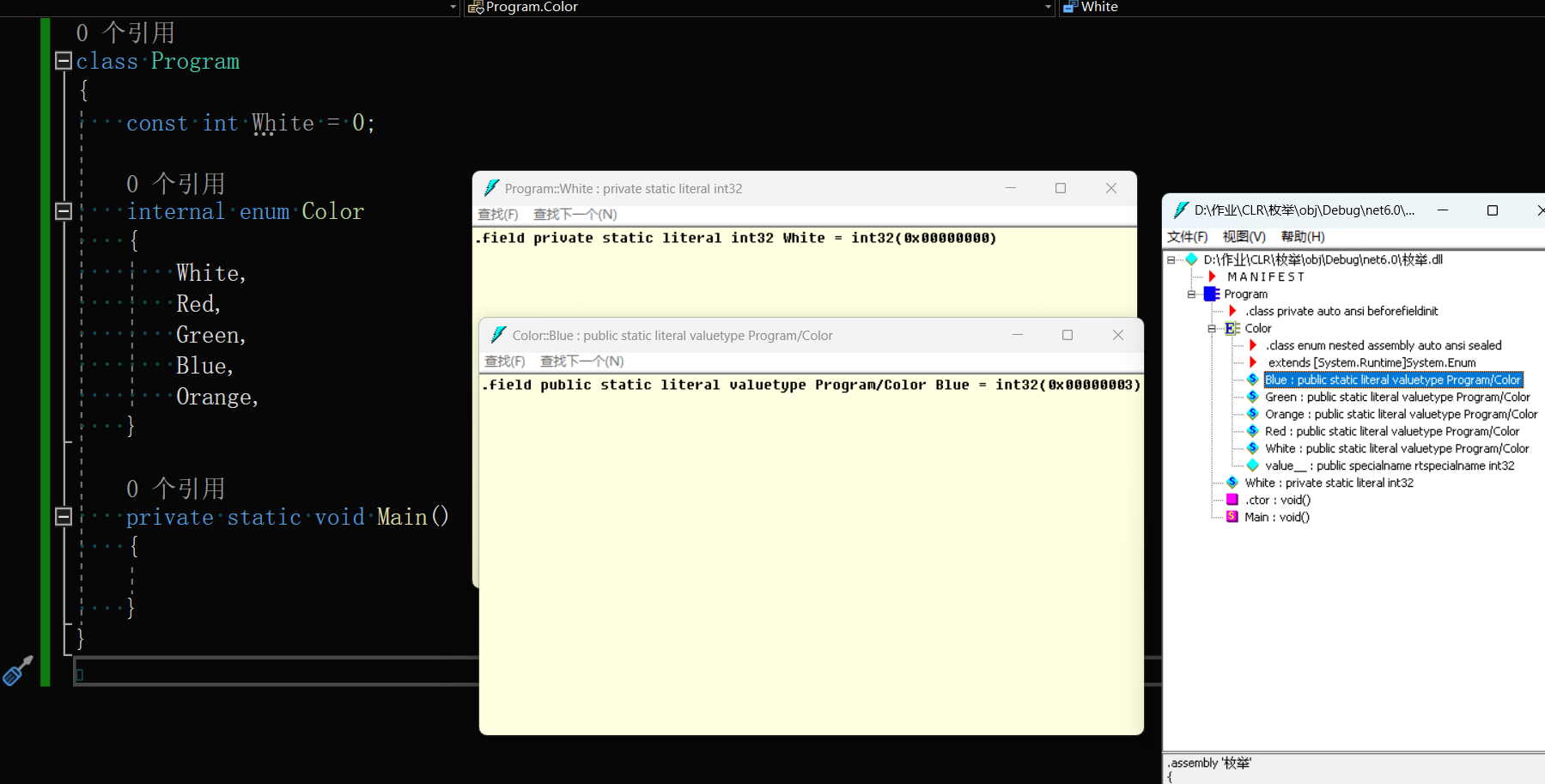
枚举类型：枚举类型是一种值类型，它定义了一组常量字段和一个实例字段，这些常量可以表示一组离散的值。枚举类型通常用于定义一组相关的常量，例如星期几、月份或颜色。（被视为常量即会内嵌入程序集，即编译时需要枚举类型的程序集而运行时不需要枚举类型的程序集）

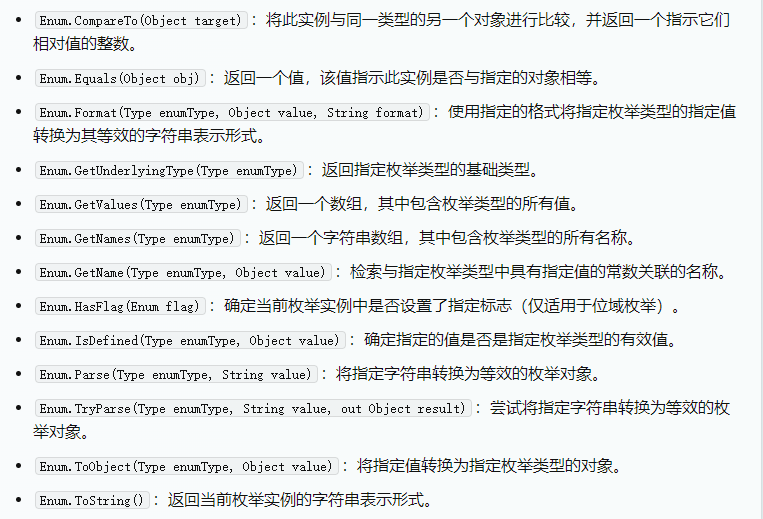


使用枚举的好处：使用枚举类型可以提高代码的可读性和可维护性。它允许使用描述性名称来表示整数值，从而使代码更清晰、更优雅。枚举类型有助于确保给定变量指定合法、期望的值，从而使代码更易于维护。

编译枚举类型时，C#把每个符号转换成类型的一个常量字段



可以在运行时通过反射相关的方法获取与枚举类型关联的所有符号及其值。但System.Enum类型提供了一些静态方法和实例方法操作枚举类型的实例，能够避免使用反射。



其中的GetUnderlyingType静态返回与枚举类型关联的基元类型（不可以是该基元类型对应的FCL类型，这是C#编译器简化实现的结果），System.Type类型也提供了一个GetEnumUnderlyingType实例方法，有一样的效果。

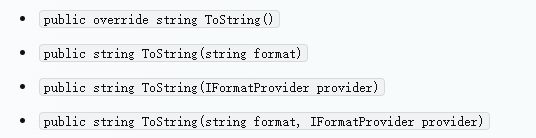
GetUnderlyingType方法例子：

手机屏幕截图

描述已自动生成

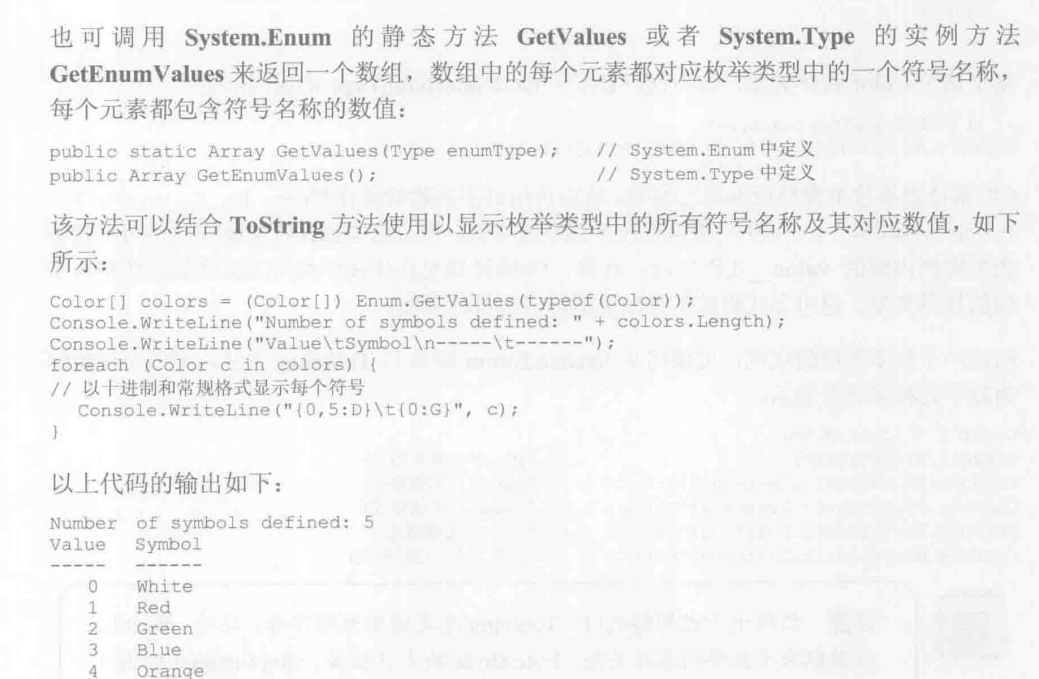
Ps：C#编译器将枚举类型视为基元类型（但枚举类型不是基元类型！！！），枚举类型可以使用可以操作基元类型的操作符来操作枚举类型，比如“==”、“！=”、“+”、“-”、“^”、“/”、“|”、“&”、“>=”、“++”、“--”等等操作符。

System.Enum.ToString的所有重载的签名：



System.Enum.GetValue和System.Type.GetEnumValues返回一个数组（Array类型），数组中的每个元素对应枚举类型中一个符号的名称。

例子：



除了使用GetValues方法，还可以使用GetName方法返回枚举类型的符号

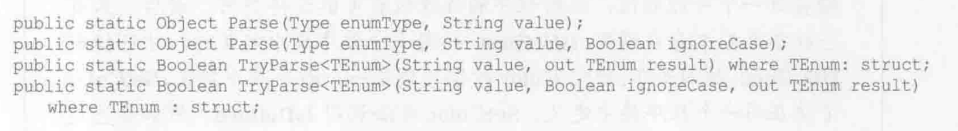
文本, 电子邮件

描述已自动生成

文本

描述已自动生成

System.Enum提供的全部Parse和TryParse方法的重载：



例子：

文本

描述已自动生成

IsDefined静态方法和IsEnumDefind的方法判断数值对于某枚举类型是否合法

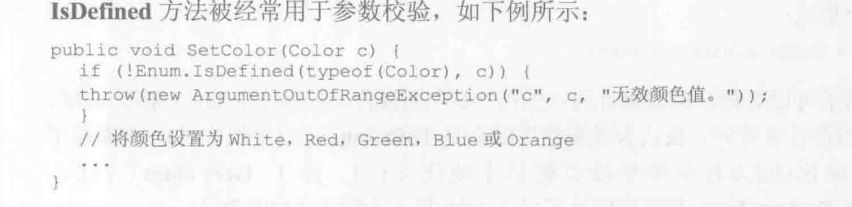
(是否合法即该枚举类型是否包含这个元素)

一些文字和图片的手机截图

描述已自动生成

Ps：IsDefined方法常用于参数校验

例如：



如果没有使用IsDefined方法，那么如果数值对于某枚举类型是不合法的，就会抛出异常，这影响了程序的健壮性。

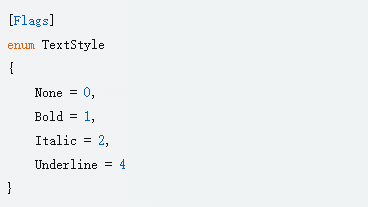
ToObject静态方法

文本

描述已自动生成

位标志

什么是位标志：位标志（Bit Flags）是一种编程技术，它允许使用单个整数值来存储多个布尔值（true/false）。这是通过将整数值视为一系列位来实现的，其中每个位都表示一个布尔值。位标志通常用于表示一组选项或状态。例如可以定义一个枚举类型，其中每个枚举成员都表示一个选项或状态，并且每个枚举成员都具有2的幂次方的值，可以使用按位或运算符（|）将多个枚举成员组合在一起，以创建一个表示多个选项或状态的位标志。

（2的4次方的值，即24=16）

使用或运算符组合起来这样使用：



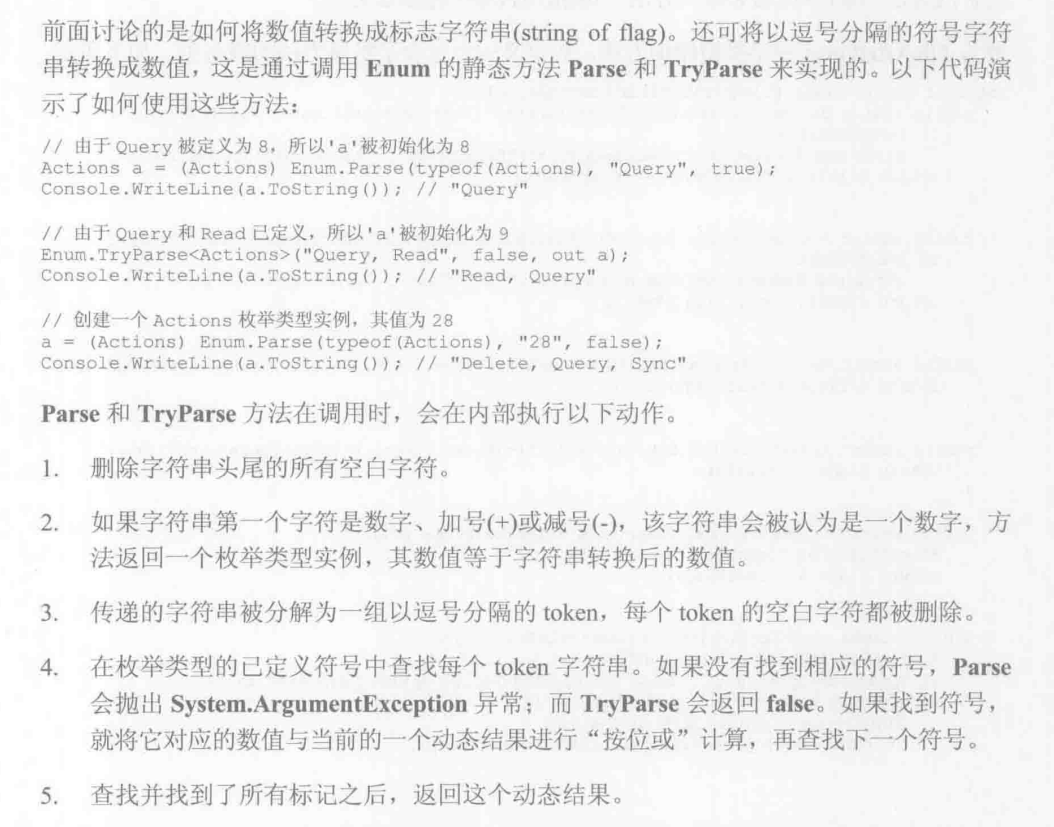
如果加上Console.WriteLine(style.ToString());

在使用[Flags]特性之后，这一行代码会输出Bold,Italic，因为[Flags]特性使得ToString把style视为一组位标识，以下是ToString如何工作的：

一些文字和图片的手机截图

描述已自动生成

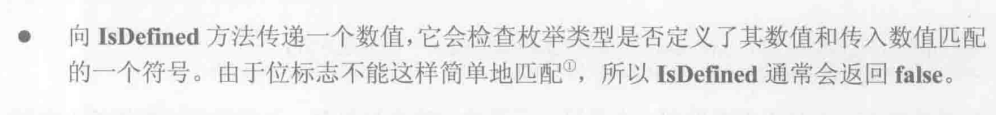
相应的上面的例子是将数值转换为标识字符串，下面的例子则恰恰相反，使用Parse方法与TryParse方法



Ps：不要对标志位使用IsDefine方法，原因有以下两点：

文本, 信件

描述已自动生成



可以为枚举类型提供扩展方法

下面是书P329为FileAttribute添加扩展方法的例子：

