**枚举类型（C# 编程指南）**

枚举类型（也称为枚举）为定义一组可以赋给变量的命名整数常量提供了一种有效的方法。例如，假设您必须定义一个变量，该变量的值表示一周中的一天。该变量只能存储七个有意义的值。若要定义这些值，可以使用枚举类型。枚举类型是使用 [enum](https://msdn.microsoft.com/zh-cn/library/sbbt4032.aspx)关键字声明的。

enum Days { Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday };

enum Months : byte { Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec };

默认情况下，枚举中每个元素的基础类型是 [int](https://msdn.microsoft.com/zh-cn/library/5kzh1b5w.aspx)。可以使用冒号指定另一种整数值类型，如前面的示例所示。有关可能的类型的完整列表，请参见 [enum（C# 参考）](https://msdn.microsoft.com/zh-cn/library/sbbt4032.aspx)。

可以通过转换验证基础数值与基础类型，如下例所示。

Days today = Days.Monday;

int dayNumber =(int)today;

Console.WriteLine("{0} is day number #{1}.", today, dayNumber);

Months thisMonth = Months.Dec;

byte monthNumber = (byte)thisMonth;

Console.WriteLine("{0} is month number #{1}.", thisMonth, monthNumber);

// Output:

// Monday is day number #1.

// Dec is month number #11.

以下是使用枚举而不使用数值类型的好处：

* 明确为客户端代码指定哪些值是变量的有效值。
* 在 Visual Studio 中，IntelliSense 列出定义的值。

如果未在枚举数列表中指定元素的值，则值将自动按 1 递增。在前面的示例中，Days.Sunday 的值为 0，Days.Monday 的值为 1，依此类推。创建新的 Days 对象时，如果不显式为其赋值，则它将具有默认值 Days.Sunday (0)。创建枚举时，应选择最合理的默认值并赋给它一个零值。这便使得只要在创建枚举时未为其显式赋值，则所创建的全部枚举都将具有该默认值。

如果变量 meetingDay 的类型为 Days，则只能将 Days 定义的某个值赋给它（无需显式强制转换）。如果会议日期更改，可以将 Days 中的新值赋给 meetingDay：

Days meetingDay = Days.Monday;

//...

meetingDay = Days.Friday;

| **https://wizardforcel.gitbooks.io/msdn-csharp/content/img/note.jpg 注意** |
| --- |
| 可以将任意整数值赋给 meetingDay。例如，代码行 meetingDay = (Days) 42 不会产生错误。但也不应该这样做，因为默认约定的是枚举变量只容纳枚举定义的值之一。将任意值赋给枚举类型的变量很有可能会导致错误。 |

可以将任意值赋给枚举类型的枚举数列表中的元素，也可以使用计算值：

enum MachineState

{

PowerOff = 0,

Running = 5,

Sleeping = 10,

Hibernating = Sleeping + 5

}

**枚举类型作为位标志**

可以使用枚举类型定义位标志，从而使该枚举类型的实例可以存储枚举数列表中定义的值的任意组合。（当然，某些组合在您的程序代码中可能没有意义或不允许使用。）

创建位标志枚举的方法是应用 [System.FlagsAttribute](https://msdn.microsoft.com/zh-cn/library/system.flagsattribute.aspx) 特性并适当定义一些值，以便可以对这些值执行 **AND**、**OR**、**NOT** 和 **XOR** 按位运算。在位标志枚举中包含一个值为零（表示“未设置任何标志”）的命名常量。如果零值不表示“未设置任何标志”，则请不要为标志指定零值。

在下面的示例中，定义了 Days 枚举的另一个版本，命名为 Days2。 Days2 具有 Flags 特性，且它的每个值都是 2 的若干次幂，指数依次递增。这样，您将能够创建值为 Days2.Tuesday 和 Days2.Thursday 的 Days2 变量。

[Flags]

enum Days2

{

None = 0x0,

Sunday = 0x1,

Monday = 0x2,

Tuesday = 0x4,

Wednesday = 0x8,

Thursday = 0x10,

Friday = 0x20,

Saturday = 0x40

}

class MyClass

{

Days2 meetingDays = Days2.Tuesday | Days2.Thursday;

}

若要在某个枚举上设置标志，请使用按位 **OR** 运算符，如下面的示例所示：

// Initialize with two flags using bitwise OR.

meetingDays = Days2.Tuesday | Days2.Thursday;

// Set an additional flag using bitwise OR.

meetingDays = meetingDays | Days2.Friday;

Console.WriteLine("Meeting days are {0}", meetingDays);

// Output: Meeting days are Tuesday, Thursday, Friday

// Remove a flag using bitwise XOR.

meetingDays = meetingDays ^ Days2.Tuesday;

Console.WriteLine("Meeting days are {0}", meetingDays);

// Output: Meeting days are Thursday, Friday

若要确定是否设置了特定标志，请使用按位 **AND** 运算，如下面的示例所示：

// Test value of flags using bitwise AND.

bool test = (meetingDays & Days2.Thursday) == Days2.Thursday;

Console.WriteLine("Thursday {0} a meeting day.", test == true ? "is" : "is not");

// Output: Thursday is a meeting day.

有关使用 [System.FlagsAttribute](https://msdn.microsoft.com/zh-cn/library/system.flagsattribute.aspx) 特性定义枚举类型时需要考虑的事项的更多信息，请参见 [System.Enum](https://msdn.microsoft.com/zh-cn/library/system.enum.aspx)。

**使用 System.Enum 方法发现和操作枚举值**

所有枚举都是 [System.Enum](https://msdn.microsoft.com/zh-cn/library/system.enum.aspx) 类型的实例。不能从 [System.Enum](https://msdn.microsoft.com/zh-cn/library/system.enum.aspx) 派生新类，但可以使用它的方法发现有关枚举实例中的值的信息以及操作这些值。

string s = Enum.GetName(typeof(Days), 4);

Console.WriteLine(s);

Console.WriteLine("The values of the Days Enum are:");

foreach (int i in Enum.GetValues(typeof(Days)))

Console.WriteLine(i);

Console.WriteLine("The names of the Days Enum are:");

foreach (string str in Enum.GetNames(typeof(Days)))

Console.WriteLine(str);

有关更多信息，请参见[AllMembers.T:System.Enum](https://msdn.microsoft.com/zh-cn/library/system.enum_members.aspx)。

此外，还可以使用扩展方法为枚举创建新方法。有关更多信息，请参见[如何：为枚举创建新方法（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/bb383974.aspx)。

**重要章节**

[更多有关变量](http://go.microsoft.com/fwlink/?LinkId=221230) 在 [开始 Visual c# 2010](http://go.microsoft.com/fwlink/?LinkId=221214)

**请参阅**

[System.Enum](https://msdn.microsoft.com/zh-cn/library/system.enum.aspx)

[C# 编程指南](https://msdn.microsoft.com/zh-cn/library/67ef8sbd.aspx)

[enum（C# 参考）](https://msdn.microsoft.com/zh-cn/library/sbbt4032.aspx)