

نوع داده شمارشی یا Enum، جهت تعاریف مقادیر ثابت و قابل شمارش در برنامه، بسیار کاربرد دارد. مقادیری که در این نوع داده تعریف می‌شوند بطور خودکار از عدد 0 شماره گذاری می‌شوند و به ترتیب یکی به آن‌ها اضافه می‌شود. برای مثال حالت زیر را در نظر بگیرید:

```
public enum Grade
{
    Failing,           // = 0
    BelowAverage,      // = 1
    Average,           // = 2
    VeryGood,          // = 3
    Excellent          // = 4
}
```

در این حالت متد ToString() نوع داده Enum عنوان مقادیر ثابت را بر می‌گرداند.

جهت برگشت مقدار عددی و شماره مقادیر ثابت‌های تعریف شده از متد ToString() با فرمت D (شماره مقدار را بصورت Decimal نشان می‌دهد) و فرمت X جهت نمایش بصورت هگزا می‌توان استفاده کرد.

روش عرف برای نمایش مقدار عددی استفاده از تبدیل نوع صریح به int است.

به منظور درک بهتر موضوع، از یک برنامه کنسول استفاده می‌کنیم تا این نوع داده شمارشی را در آن استفاده کنیم.

```
static void Main(string[] args)
{
    Grade grade = Grade.Average;
    Console.WriteLine(grade.ToString()); // Print Average
    Console.WriteLine(grade.ToString("D")); // Print 2
    Console.WriteLine(grade.ToString("X")); // Print 00000002
    Console.WriteLine((int) grade); //Print 2
    Console.ReadKey();
}
```

### تغییر شماره (اندیس) مقادیر ثابت تعریف شده:

جهت تغییر شماره مقادیر کافیت بصورت زیر عمل کنیم:

```
public enum Grade
{
    Failing = 5,
    BelowAverage = 10,
    Average = BelowAverage + 5, // = 15
    VeryGood = 18,
    Excellent = 20
}
```

همانطور که در بالا می‌بینید برای مقدار Average بصورت ترکیبی عمل شده است.

بصورت پیش فرض کامپایلر سی شارپ از Int32 جهت نگهداری اعضای یک Enum استفاده می‌کند. هر چند غیر معقول به نظر می‌رسد اما شما می‌توانید این نوع را به long - uint - ushort - short - sbyte - byte تغییر دهید.

```
public enum Grade : byte
{
    Failing = 5,
    BelowAverage = 10,
    Average = BelowAverage + 5, // = 15
    VeryGood = 18,
    Excellent = 20
}
```

آشنایی با مفاهیم نوع داده Enum و توسعه آن - قسمت یکم

---

```
}
```

بدیهی است در این حالت خروجی دستور زیر 0F خواهد بود:

```
Console.WriteLine(grade.ToString("X")); // Print 0F
```

همچنین به خروجی دستورات زیر در حالت فوق توجه کنید:

```
Console.WriteLine("Underlying type: {0}", Enum.GetUnderlyingType(grade.GetType())); // Print  
System.Byte
```

```
Console.WriteLine("Type Code      : {0}", grade.GetTypeCode()); // Print Byte
```

و البته این:

```
Console.WriteLine("Value : {0}", (int)grade); // Print 15
```

در قسمت دوم این مطلب با استفاده از فضای نام System.Reflection و Extension Method ها و Custom Attribute کمی مقادیر Enum را توسعه خواهیم داد.

## نظرات خوانندگان

نویسنده: احمد احمدی  
تاریخ: ۱۳۹۱/۰۵/۰۲ ۲:۴۶

سلام - بحث Enum خیلی جالبی است .  
فقط فراموش کردید برچسب Enum را هم به این قسمت از مقالاتون اضافه کنید .  
با تشکر ...

نویسنده: علیرضا اسم‌رام  
تاریخ: ۱۳۹۱/۰۵/۰۲ ۹:۳۲

سلام. با تشکر از شما. این موضوع را در قسمت سوم و باز هم با کمک متدهای الحاقی اجرا می‌کنیم. در نهایت امیدوارم یک کلاس از متدهای الحاقی جهت کار با Enumها داشته باشیم.  
ممنون از یادآوری شما.

نویسنده: KishIsland  
تاریخ: ۱۳۹۱/۱۲/۲۷ ۱۵:۲۰

سپاس.مفید بود