

اگر با نوع داده Enum آشنایی ندارید [قسمت یکم این مطلب](#) را بخوانید.

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

**بازنویسی متد ToString():** امکان بازنویسی متد ToString() در نوع Enum وجود ندارد. بنابراین برای چاپ عبارت Very Good به جای VeryGood تکنیک زیر جالب به نظر می‌رسد. هر چند استفاده از آرایه و ترکیب اندیس آن با Enum و یا استفاده از HashTable راه‌هایی است که در ابتدا به ذهن ما خطور می‌کند اما لطفاً به ادامه مطلب توجه فرمایید! با در نظر گرفتن مثال قبل، یک Custom Attribute به نوع داده شمارشی اضافه می‌کنیم. برای این منظور بصورت زیر عمل می‌کنیم.

1. ایجاد کلاس Description که از کلاس Attribute مشتق شده است و تعریف خصوصیت Text:

```
class Description : Attribute
{
    public string Text;
    public Description(string text)
    {
        Text = text;
    }
}
```

2. به سراغ نوع Enum تعریف شده رفته و جهت استفاده از صفت جدید که در مرحله قبل پیاده سازی کردیم، تغییرات را به شکل زیر اعمال می‌کنیم:

```
public enum Grade
{
    [Description("Mardood")]
    Failing = 5,

    [Description("Ajab Shansi")]
    BelowAverage = 10,

    [Description("Bad Nabood")]
    Average = BelowAverage + 5,

    [Description("Khoob Bood")]
    VeryGood = 18,

    [Description("Gol Kashti")]
    Excellent = 20
}
```

تنها کاری که باقی مانده یاری گرفتن از متدهای الحاقی (Extension Methods) جهت خواندن مقدار Description است:

```
public static class ExtensionMethodCls
{
    public static string GetDescription(this Enum enu)
    {
        Type type = enu.GetType();
```

```

        MemberInfo[] memInfo = type.GetMember(enu.ToString());
        if (memInfo != null && memInfo.Length > 0)
        {
            object[] attrs = memInfo[0].GetCustomAttributes(typeof(Description), false);
            if (attrs != null && attrs.Length > 0)
                return ((Description)attrs[0]).Text;
        }
        return enu.ToString();
    }
}

```

حال نوع Enum ما کمی توسعه یافته است و توسط متد GetDescription می توان متن دلخواه و متناسب با مقدار را نمایش داد:

```
Console.WriteLine(grade.GetDescription()); // Print Bad Nabood
```

کد کامل مثال بررسی شده نیز بصورت زیر خواهد بود:

```

using System;
using System.Reflection;

namespace CSharpEnum
{
    class Description : Attribute
    {
        public string Text;
        public Description(string text)
        {
            Text = text;
        }
    }

    public enum Grade
    {
        [Description("Mardood")]
        Failing = 5,

        [Description("Ajab Shansi")]
        BelowAverage = 10,

        [Description("Bad Nabood")]
        Average = BelowAverage + 5,

        [Description("Khoob Bood")]
        VeryGood = 18,

        [Description("Gol Kashti")]
        Excellent = 20
    }

    public static class ExtensionMethodCls
    {
        public static string GetDescription(this Enum enu)
        {
            Type type = enu.GetType();
            MemberInfo[] memInfo = type.GetMember(enu.ToString());
            if (memInfo != null && memInfo.Length > 0)
            {
                object[] attrs = memInfo[0].GetCustomAttributes(typeof(Description), false);
                if (attrs != null && attrs.Length > 0)
                    return ((Description)attrs[0]).Text;
            }
            return enu.ToString();
        }
    }
}

```

```
}  
class Program  
{  
    static void Main(string[] args)  
    {  
        const Grade grade = Grade.Average;  
        Console.WriteLine("Underlying type: {0}", Enum.GetUnderlyingType(grade.GetType()));  
        Console.WriteLine("Type Code      : {0}", grade.GetTypeCode());  
        Console.WriteLine("Value          : {0}", (int)grade);  
        Console.WriteLine("-----");  
        Console.WriteLine(grade.ToString()); // name of the constant  
        Console.WriteLine(grade.ToString("G")); // name of the constant  
        Console.WriteLine(grade.ToString("F")); // name of the constant  
        Console.WriteLine(grade.ToString("x")); // value is hex  
        Console.WriteLine(grade.ToString("D")); // value in decimal  
        Console.WriteLine("-----");  
        Console.WriteLine(grade.GetDescription()); // Print Bad Nabood  
        Console.ReadKey();  
    }  
}
```

با استفاده از این تکنیک (مخصوصاً ما فارسی زبان ها) به راحتی می‌توانیم از مقادیر Enum استفاده بهتری ببریم. برای مثال اگر بخواهیم یک مقدار Enum را بصورت فارسی در یک Drop Down List نمایش دهیم این تکنیک بسیار مفید خواهد بود.

## نظرات خوانندگان

نویسنده: حسین  
تاریخ: ۱۳۹۱/۰۵/۰۲ ۱۲:۱۹

سلام  
کدهایی که به این صورت نوشته میشن چی بهشون گفته میشه؟ چی هستن؟  
خیلی جاها این کدها رو دیدم به صورتهای مختلف  
کمی تو گوگل سرچ کردم نتیجه مطلوبی نگرفتم یعنی نمیدونم دنبال چی بگردم  
لطفا راهنمایی کنید.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۰۵/۰۲ ۱۲:۲۳

به Attribute ای که در اینجا توسعه داده شده (یا از آن استفاده شده)، اصطلاحاً data annotation هم گفته می‌شود. یک سری از فریم ورک‌ها به صورت توکار قادر به استفاده از آن‌ها هستند مانند ASP.NET MVC برای نمایش توضیحات مرتبط یا نمایش برچسب‌ها به صورت خودکار.  
مطالب فوق رو می‌تونید پایه طراحی این نوع کتابخانه‌ها در نظر بگیرید.

نویسنده: حسین  
تاریخ: ۱۳۹۱/۰۵/۰۲ ۱۲:۳۰

کدهایی که تو linq هم استفاده میشن از همین دسته اند؟  
مثلا

```
[global::System.Data.Linq.Mapping.ColumnAttribute(Storage="_ID", AutoSync=AutoSync.Always, DbType="Int NOT NULL IDENTITY", IsDbGenerated=true)]
```

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۰۵/۰۲ ۱۲:۴۶

بله. به این‌ها [Attribute](#)، meta-data، یا data annotation گفته می‌شود.

نویسنده: ایمان محمدی  
تاریخ: ۱۳۹۱/۰۵/۰۶ ۱۴:۰۹

نیازی به تعریف کلاس Description نیست.

کافیه از فضای نام System.ComponentModel استفاده کنید و در مقدار بازگشتی متد GetDescription بجای

```
return ((Description)attrs[0]).Text;
```

بنوسید

```
return ((DescriptionAttribute)attrs[0]).Description;
```

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۰۵/۰۶ ۱۴:۲۰

هدف این مطلب بررسی زیر ساخت این نوع طراحی بوده نه صرفا مصرف کننده محض بودن. تشابه اسمی اهمیتی ندارد. روش انجام کار مهم بوده در اینجا.

نویسنده: ایمان محمدی  
تاریخ: ۱۶:۵۱۳۹۱/۰۵/۰۶

وقتی کلاس Description در فضای نام System.ComponentModel وجود دارد دلیلی نداره کلاس مشابه ای تعریف کنیم. بخاطر اینکه مصرف کننده محض نباشیم یک متد الحاقی به نام GetEnumerator() اضافه کردم که لیست اعضای یک Enum رو برای استفاده در کمبو باکس و ... بر می گردونه :  
ابتدا کلاس زیر به کلاس ExtensionMethodCls اضافه می کنیم :

```
public class EnumObject
{
    public Enum ValueMember { get; set; }
    public int intValueMember
    {
        get { return int.Parse(ValueMember.ToString("D")); }
    }
    public string stringValueMember
    {
        get { return ValueMember.ToString(""); }
    }
    public string DisplayMember
    {
        get { return ValueMember.GetDescription(); }
    }
}
```

و متد الحاقی زیر رو برای گرفتن لیست تعریف می کنیم:

```
public static List<EnumObject> GetEnumerator(this Enum enu)
{
    List<EnumObject> li = new List<EnumObject>();
    foreach (var item in enu.GetType().GetEnumValues())
    {
        li.Add(new EnumObject { ValueMember = (Enum)item });
    }
    return li;
}
```

نحوه استفاده :

```
comboBox1.DataSource = Grade.VeryGood.GetEnumerator();
comboBox1.DisplayMember = "DisplayMember";
comboBox1.ValueMember = "ValueMember";
```

همون طوری که در کد بالا می بینید برای گرفتن لیست مجبور شدیم یکی از اعضای enum رو انتخاب کنیم ( Grade.VeryGood )  
GetEnumerator() ( ) شاید انتخاب یکی از اعضا و بعد درخواست لیست اعضا رو کردن کار قشنگی نباشه به همین دلیل متد زیر رو تعریف کردیم :

```
public static List<EnumObject> EnumToList<T>()
{
    Type enumType = typeof(T);
    if (enumType.BaseType != typeof(Enum))
        throw new ArgumentException("T must be of type System.Enum");

    List<EnumObject> li = new List<EnumObject>();
    foreach (var item in enumType.GetEnumValues())
    {
        li.Add(new EnumObject { ValueMember = (Enum)item });
    }
}
```

```
        return li;
    }
}
```

نحوه استفاده :

```
comboBox1.DataSource = ExtensionMethodCls.EnumToList<Grade>();
comboBox1.DisplayMember = "DisplayMember";
comboBox1.ValueMember = "ValueMember";
```

کد کامل :

```
public static class ExtensionMethodCls
{
    public class EnumObject
    {
        public Enum ValueMember { get; set; }
        public int intValueMember
        {
            get { return int.Parse(ValueMember.ToString("D")); }
        }
        public string stringValueMember
        {
            get { return ValueMember.ToString(""); }
        }
        public string DisplayMember
        {
            get { return ValueMember.GetDescription(); }
        }
    }

    public static List<EnumObject> EnumToList<T>()
    {
        Type enumType = typeof(T);
        if (enumType.BaseType != typeof(Enum))
            throw new ArgumentException("T must be of type System.Enum");

        List<EnumObject> li = new List<EnumObject>();
        foreach (var item in enumType.GetEnumValues())
        {
            li.Add(new EnumObject { ValueMember = (Enum)item });
        }
        return li;
    }

    public static List<EnumObject> GetEnumList(this Enum enu)
    {
        List<EnumObject> li = new List<EnumObject>();
        foreach (var item in enu.GetType().GetEnumValues())
        {
            li.Add(new EnumObject { ValueMember = (Enum)item });
        }
        return li;
    }

    public static string GetDescription(this Enum enu)
    {
        Type type = enu.GetType();
        MemberInfo[] memInfo = type.GetMember(enu.ToString());
        if (memInfo != null && memInfo.Length > 0)
        {
            object[] attrs = memInfo[0].GetCustomAttributes(typeof(DescriptionAttribute), false);
            if (attrs != null && attrs.Length > 0)
                return ((DescriptionAttribute)attrs[0]).Description;
        }
        return enu.ToString();
    }
}
```

```
}
```

نویسنده: علیرضا اسم‌رام  
تاریخ: ۱۳۹۱/۰۵/۰۷ ۹:۴۵

با سلام خدمت شما دوست عزیز. همانطور که آقای نصیری اشاره کردند هدف این مطلب و البته این سری از مطالب آشنایی قدم به قدم با مفاهیم این نوع داده بوده است. انقیاد کنترل‌ها با اشیاء از نوع Enum موضوع بعدی این سری از مطالب بود. البته باز هم از شما تشکر می‌کنم بخاطر این نظر. اگر این قابلیت در سایت ایجاد شود که بتوان به نظرات لینک داد بهتر است.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۰۵/۰۷ ۱۰:۰۰

علامت آبی رنگ # کنار هر نظر، لینک مستقیم به همان نظر است.

نویسنده: علیرضا اسم‌رام  
تاریخ: ۱۳۹۱/۰۵/۰۷ ۲۲:۰۸

مرسی. خیلی خیلی خوب...

نویسنده: سعید یزدانی  
تاریخ: ۱۳۹۱/۱۱/۱۷ ۲۰:۵۴

باسلام؛ لطفاً به توضیحی در باره‌ی MemberInfo بدید.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۱۱/۱۸ ۹:۱۱

مراجعه کنید به [MSDN](#) و مثال‌های آن در همان صفحه.