

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

enumeration

مدرس : مهندس افشین رفوآ

تصور کنید، هنگام برنامه نویسی برای شرکتی که معاملات املاک انجام می دهد، می خواهید برنامه ای که نوشته اید از مشتری نوع خانه ای را که قصد خرید آن را دارند بپرسد.

مثال

```
using System;
public class Exercise
{
    static int Main()
    {
        int typeOfHouse = 0;
        int typeOfGarage = 0;
        Console.WriteLine("Enter the type of house you
want to purchase");
        Console.WriteLine("1 - Single Family");
        Console.WriteLine("2 - Townhouse");
        Console.WriteLine("3 - Condominium");
        Console.Write("Your Choice: ");
        typeOfHouse = int.Parse(Console.ReadLine());
        Console.WriteLine("Enter the type of garage you
want");

        Console.WriteLine("0 - Doesn't matter");
        Console.WriteLine("1 - Interior");
        Console.WriteLine("2 - Exterior");
        Console.Write("Your Choice: ");
        typeOfGarage = int.Parse(Console.ReadLine());
        Console.Clear();
        Console.WriteLine("\nHouse Type: {0}",
typeOfHouse);
        Console.WriteLine("Garage Type: {0}",
typeOfGarage);
        return 0;
    }
}
```

مثالی از اجرای برنامه

```
Enter the type of house you want to purchase
1 - Single Family
```

```

2 - Townhouse
3 - Condominium
Your Choice: 3
Enter the type of garage you want
0 - Doesn't matter
1 - Interior
2 - Exterior
Your Choice: 1
-----
House Type: 3
Garage Type: 1
Press any key to continue...

```

برای چنین برنامه ای، ارقام ممکن است گنگ یا مبهم باشد. عدد 1، رقمی کلی است ولی در برنامه ی مورد نظر ممکن است نشانگر خانه ای با ظرفیت گنجایش تنها یک خانوار باشد. حال برنامه ی نام برده از ثابت 1 در راه های معنی دار مختلفی بهره می برد. برای ایجاد امکان دادن بیش از یک معنی یا کاربرد به یک عدد ثابت، به خصوص هنگامی که قرار است عدد مربوطه جزئی از یک سری باشد، زبان برنامه نویسی C# به شما اجازه می دهد یک نوع لیست ایجاد کنید.

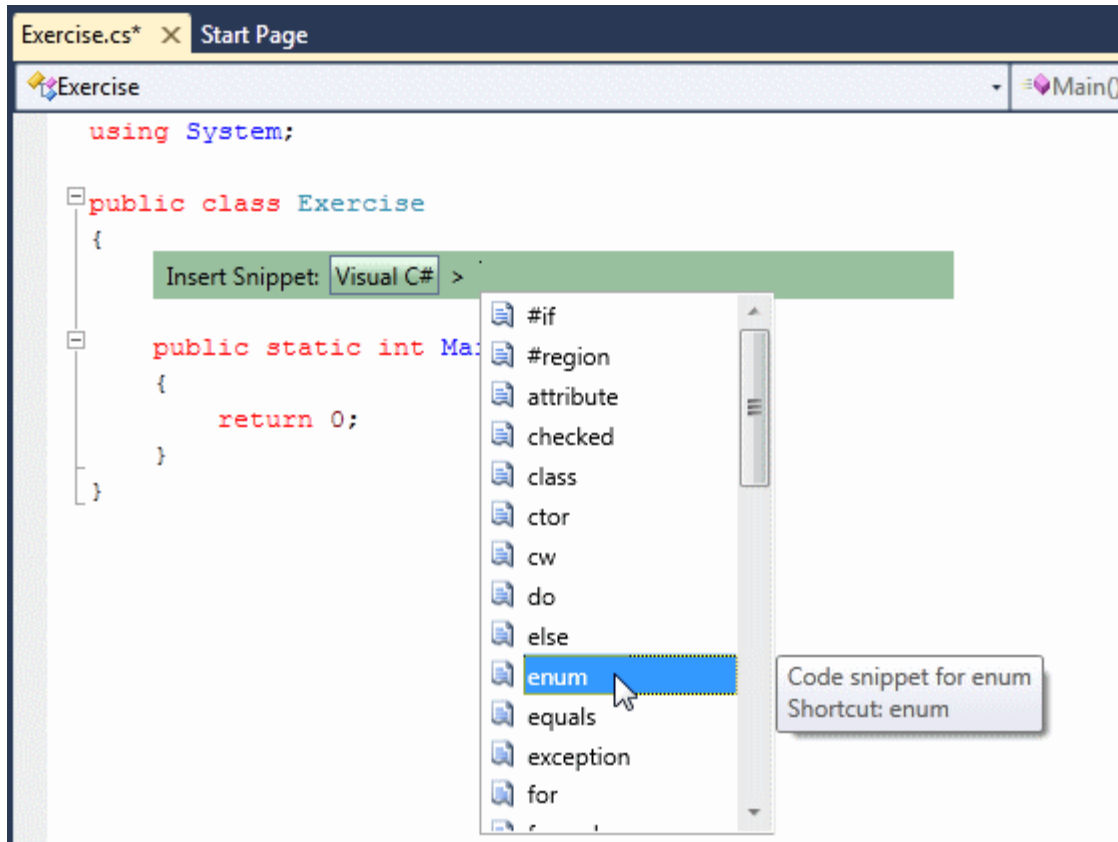
enumeration در واقع یک سری عدد صحیح ثابت (**constant integer**) است که هر یک موقعیت خاصی در لیست دارد و با اسم مشخصی شناسایی می شود. بر این اساس، به جای به خاطر داشتن این که ثابت 1 نشانگر تک خانوار است، می توان لیستی ایجاد کرد که نمونه ی خانه ی مزبور را شامل می شود. در لیست دیگر، به جای استفاده ی دوباره از 1، می توان اسم برای آن انتخاب کرد. در نتیجه، در هر لیست، اگرچه ثابت 1 باز هم در نظر گرفته می شود، حداقل معنای آن دقیق خواهد بود.

ایجاد enumeration

برای این منظور، کلیدواژه ی **enum** و به دنبال آن اسم **enumeration** و همچنین اسمی مختص هر یک از آیتم های لیست را تایپ کنید. اسم شمارنده (**enumerator**) و همچنین اسم تک تک آیتم های لیست از همان قوانینی که برای انتخاب اسم (ها) نام بردیم پیروی می کند. فرمول ایجاد **enumeration** به شرح زیر است.

```
enum Enumeration_Name {Item1, Item2, Item_n};
```

برای ایجاد **enumeration**، می توانید کد را به صورت دستی وارد کنید. برای استفاده از **code snippet**، روی بخش مورد نظر راست کلیک کرده، سپس **Insert Snippet...** را انتخاب کنید. اکنون روی **Visual C#** دوبار کلیک کنید. در لیستی که نمایان می شود، روی **enum** دوبار کلیک کنید.



مثال

```
using System;
public class Exercise
{
    enum HouseType { Unknown, SingleFamily, TownHouse,
    Condominium }
    static int Main()
    {
        return 0;
    }
}
```

تعریف متغیر enumeration

پس از ایجاد **enumeration**، هر یک از اعضای آن، مقدار عدد طبیعی مثل 0، 4، 12، 25 خواهند داشت.

پس از ایجاد **enumeration**، حتی می توان از آن متغیر تعریف کرد. به مثال زیر توجه کنید.

```
using System;
public class Exercise
{
    enum HouseType { Unknown, SingleFamily, TownHouse,
Condominium }
    static int Main()
    {
        HouseType propType;
        return 0;
    }
}
```

لازم به ذکر است که می توان متغیر (از) نوع **enumeration** را با کلید واژه **var** نیز تعریف کرد.

مقداردهی اولیه ی متغیر **enumeration**

پس از تعریف متغیر برای نوع **enumeration**، به منظور مقداردهی اولیه ی آن، مشخص کنید کدام عضو **enumeration** به متغیر اختصاص داده شود. شایان توجه است که تنها عضو شناس **enumeration** باید به متغیر تخصیص یابد. برای این منظور، در سمت راست عملگر جایگزین، اسم **enumeration** و به دنبال آن عملگر نقطه، عضوی که می خواهید مقدار آن را اختصاص دهید، تایپ کنید.

مثال

```
using System;
public class Exercise
{
    enum HouseType { Unknown, SingleFamily, TownHouse,
Condominium }
    static int Main()
    {
        var propType = HouseType.SingleFamily;
        return 0;
    }
}
```

همچنین می توان مقداری را که متغیر تعریف شده در حال حاضر دارد، به دست آورد. برای مثال، می توان آن را در کنسول و با استفاده از متدهای `Write()` و `WriteLine()` نمایش داد. به مثال زیر توجه کنید.

```
using System;
public class Exercise
{
    enum HouseType { Unknown, SingleFamily, TownHouse,
    Condominium }
    static int Main()
    {
        var propType = HouseType.SingleFamily;
        Console.WriteLine("House Type: {0}", propType);
        return 0;
    }
}
```

نتیجه ی زیر را به دست می دهد.

```
House Type: SingleFamily
Press any key to continue...
```

enumeration در واقع لیستی از اعداد است که هر یک توسط اسم معینی شناسایی می شود. به صورت پیش فرض، اولین آیتم لیست مقدار 0 را دارد، دومی مقدار 1 و غیره... برای مثال، در **enumeration** `HouseType`، مقدار 0 را دارد، در حالی که عضو `Townhouse` مقدار 2. مقادیر گفته شده پیش فرض هستند. چنانچه، با مقادیر بالا موافق نیستید می توانید مقدار هر یک از اعضای لیست را خود تعیین کنید. برای مثال، اگر بخواهیم عضو `Unknown` در **enumeration** فوق مقدار 5 داشته باشد، کافی است عملگر جایگزین " = " را به کار ببرید. شماره ده به این ترتیب خواهد بود.

```
using System;
public class Exercise
{
    enum HouseType { Unknown = 5, SingleFamily,
    TownHouse, Condominium }
    static int Main()
    {
        return 0;
    }
}
```

در این مورد عضو `Unknown`، مقدار 5 را خواهد داشت، `SingleFamily` مقدار 6، به این خاطر که پس از عضوی قرار گرفته که مقدار آن 1 می باشد (بنابراین $5 + 1 = 6$). عضو `Townhouse` مقدار 7 و

Condominium مقدار 8 را خواهد داشت. همچنین می توان مقداری را به چندین عضو **enumeration** اختصاص داد.

مثال

```
using System;
public class Exercise
{
    enum HouseType { Unknown = 3, SingleFamily = 12,
TownHouse, Condominium = 8 }
    static int Main()
    {
        return 0;
    }
}
```

در این مثال، **Townhouse** مقدار 13 را خواهد داشت زیرا پس از عضو **SingleFamily** قرار گرفته که مقدار آن 12 می باشد.

قابلیت رویت، دسترسی به enumeration

به صورت پیش فرض، پس از ایجاد **enumeration**، دسترسی به آن تنها از طریق پروژه ای امکان پذیر می باشد که در آن، **enumeration** ایجاد شده است. درست مثل کلاس، می توان (سطح دسترسی) دسترسی به **enumeration** را بیرون از پروژه اش مدیریت و تنظیم کرد. به عبارت دیگر، می توان **enumeration** را پنهان کرد یا برعکس بیرون از پروژه اش آن را قابل رویت ساخت. برای این منظور، می توانید کلیدواژه ی های **public** و **private** را پیش از **enumeration** قرار دهید. نظر خود را به مثال زیر جلب کنید.

```
using System;
public class Exercise
{
    public enum HouseType
    {
        Unknown,
        SingleFamily,
        TownHouse,
        Condominium
    }
    static int Main()
```

```

    {
        HouseType propType = HouseType.SingleFamily;
        Console.WriteLine("House Type: {0}", propType);
        return 0;
    }
}

```

enumeration به عنوان متغیر عضو

پس از ایجاد **enumeration**، می توان از آن به عنوان یک نوع داده برای تعریف متغیر استفاده کرد. برای ایجاد فیلد (از) نوع **enumeration**، باید همان قوانینی را که برای داده های نوع اولیه رعایت می کردیم پیاده کنیم :
اسم **enumeration**، به دنبال آن اسم متغیر و نقطه ویرگول ";" . به مثال زیر توجه کنید :

```

public enum HouseType
{
    Unknown,
    SingleFamily,
    TownHouse,
    Condominium
}
public class House
{
    HouseType propertyType;
}

```

به همین ترتیب، می توان هر تعداد متغیر عضو که لازم است ایجاد کرد. پس از تعریف متغیر، برای مقداردهی اولیه ی آن، عضو دلخواه **enumeration** (**enumeration member**) را به متغیر اختصاص دهید.

مثال

```

public enum HouseType
{
    Unknown,
    SingleFamily,
    TownHouse,
    Condominium
}
public class House
{
    HouseType propertyType;
    public House()
    {
        propertyType = HouseType.Unknown;
    }
}

```



```
    }  
}
```

پس از مقداردهی اولیه متغیر عضو، می توانید آن را هر طور که مایلید مورد استفاده قرار دهید. برای مثال، به منظور نمایش مقدار متغیر عضو می توان آن را به متدهای **Write()** یا **Writeline()** ارسال کرد. به مثال زیر توجه کنید.

```
using System;  
public enum HouseType  
{  
    Unknown,  
    SingleFamily,  
    TownHouse,  
    Condominium  
}  
public class House  
{  
    public HouseType propertyType;  
    public House()  
    {  
        propertyType = HouseType.Unknown;  
    }  
    public void Display()  
    {  
        Console.WriteLine("Property Type: {0}",  
propertyType);  
    }  
}  
public class Exercise  
{  
    static int Main()  
    {  
        var propType = new House();  
        propType.Display();  
        Console.WriteLine();  
        propType.propertyType = HouseType.SingleFamily;  
        propType.Display();  
        Console.WriteLine();  
        return 0;  
    }  
}
```

نتیجه ی زیر حاصل می شود.

Property Type: Unknown
Property Type: SingleFamily
Press any key to **continue...**

چنانچه از آن به عنوان یک نوع داده ی ساده استفاده کنید، می توانید متدی ایجاد کنید که **enumeration** بازمی گرداند. همچنین می توانید **enumeration** را به عنوان یک آرگومان به متد ارسال کنید.