(T8)比較Enum、EnumGetValue、EnumGetNames  
CourseGUID: 29f1196a-1950-41a4-b9c1-dd13a9e92d92  
=======================================================================  
(T8)比較Enum、EnumGetValue、EnumGetNames  
=======================================================================  
0. Summary

1. Create New Project

2. Program  
=======================================================================

0. Summary

1.

Enum

---------------------

1.1.

Using Enum keyword to create enumerations and it is strongly value typed constants.

The default underlying type of an enum is int.

You may use " : short " to set the underlying type of an enum is short.

The default value for first element is ZERO and gets incremented by 1.

---------------------

1.2.

Syntax :

//public enum EnumName  [ : underlyingType ]

//{

//    EnumValue1 [ = StarValue],

//    EnumValue2,

//    EnumValue3 [ = SpecificValue],

//    ....

//}

E.g.1.

//public enum MagicType   // : int

//{

//    Wood,

//    Fire,

//    Earth,

//    Metal,

//    Water

//}

E.g.2.

//public enum MagicType2 : short

//{

//    Wood = 5,

//    Fire,   //6

//    Earth   //7

//}

E.g.3.

//public enum MagicType4 : short

//{

//    Wood = 8,

//    Fire = 100,

//    Earth = 20

//}

---------------------

1.3.

//int woodInt = (int)MagicType.Wood;

Convert Enum to int

---------------------

1.4.

//MagicType magicType1 = (MagicType)1;

Convert int to Enum

---------------------

1.5.

Enum.GetValues list Enum underlying type values.

E.g.

int[] MagicTypeValues = (int[])Enum.GetValues(typeof(MagicType));

//MagicTypeValues == {0,1,2,3,4}

---------------------

1.6.

Enum.GetNames list Enum underlying type names.

string[] MagicTypeNames = Enum.GetNames(typeof(MagicType));

//MagicTypeNames == {"Wood","Fire","Earth","Metal","Water"}

=============================================

1. Create New Project

File --> New --> Project... -->

Visual C# -->  **Console App** **(.Net Framework)** -->

Name: **Sample**







=============================================

2. Program

using System;

using System.Collections.Generic;

using System.Linq;

using OnLineGame;

namespace Sample

{

    class Program

    {

        static void Main(string[] args)

        {

            Console.WriteLine("Gamer.GamerInfo(gamer) ========================================");

            List<Gamer> gamerList = new List<Gamer>();

            gamerList.Add(new Gamer { Name = "Name01", Gender = 0 });

            gamerList.Add(new Gamer { Name = "Name02", Gender = 2 });

            gamerList.Add(new Gamer { Name = "Name03", Gender = 1 });

            foreach (Gamer gamer in gamerList)

            {

                Console.WriteLine(Gamer.GamerInfo(gamer));

            }

            ////Name: Name01

            ////Gender: 0

            ////MagicList:

            ////Name: Name02

            ////Gender: 2

            ////MagicList:

            ////Name: Name03

            ////Gender: 1

            ////MagicList:

            //how do I know what does "Gender: 1" mean?

            Console.WriteLine("Gamer.GamerInfo2(gamer) ========================================");

            foreach (Gamer gamer in gamerList)

            {

                Console.WriteLine(Gamer.GamerInfo2(gamer));

            }

            ////Name: Name01

            ////Gender: Unknown

            ////MagicList :

            ////Name: Name02

            ////Gender: Female

            ////MagicList :

            ////Name: Name03

            ////Gender: Male

            ////MagicList :

            //I have to use Gamer.GetGender(int gender) method to know the meaning.

            //It is totally un-readable. Thus, we need enum.

            Console.WriteLine("firstGamerMagicList ========================================");

            List<Magic> firstGamerMagicList = new List<Magic>();

            firstGamerMagicList.Add(new Magic { MagicName = "WoodMagic", MpCost = 5, MagicType = MagicType.Wood });

            firstGamerMagicList.Add(new Magic { MagicName = "FireMagic", MpCost = 4, MagicType = MagicType.Fire });

            firstGamerMagicList.Add(new Magic { MagicName = "EarthMagic", MpCost = 3, MagicType = MagicType.Earth });

            firstGamerMagicList.Add(new Magic { MagicName = "MetalMagic", MpCost = 2, MagicType = MagicType.Metal });

            firstGamerMagicList.Add(new Magic { MagicName = "WaterMagic", MpCost = 1, MagicType = MagicType.Water });

            //MagicType = MagicType.Wood   is more readable.

            gamerList.First().MagicList = firstGamerMagicList;

            foreach (Magic magic in gamerList.First().MagicList)

            {

                Console.WriteLine(Magic.MagicInfo(magic));

            }

            ////MagicName: WoodMagic

            ////MpCost: 5

            ////MagicList: Wood

            ////MagicName : FireMagic

            ////MpCost: 4

            ////MagicList: Fire

            ////MagicName : EarthMagic

            ////MpCost: 3

            ////MagicList: Earth

            ////MagicName : MetalMagic

            ////MpCost: 2

            ////MagicList: Metal

            ////MagicName : WaterMagic

            ////MpCost: 1

            ////MagicList: Water

            // The code is readable, and the return is readable.

            Console.WriteLine("Enum to Int ========================================");

            int woodInt = (int)MagicType.Wood;

            int fireInt = (int)MagicType.Fire;

            int earthInt = (int)MagicType.Earth;

            int metalInt = (int)MagicType.Metal;

            int waterInt = (int)MagicType.Water;

            Console.WriteLine($"woodInt : {woodInt} \n" +

                              $"fireInt : {fireInt} \n" +

                              $"earthInt : {earthInt} \n" +

                              $"metalInt : {metalInt} \n" +

                              $"waterInt : {waterInt}");

            //woodInt: 0

            //fireInt: 1

            //earthInt: 2

            //metalInt: 3

            //waterInt: 4

            Console.WriteLine("Int to Enum ========================================");

            MagicType magicType0 = 0;   // When 0, you don't need cast keyword "(MagicType)"

            MagicType magicType1 = (MagicType)1;

            MagicType magicType2 = (MagicType)2;

            MagicType magicType3 = (MagicType)3;

            MagicType magicType4 = (MagicType)4;

            Console.WriteLine($"magicType0 : {magicType0} \n" +

                              $"magicType1 : {magicType1} \n" +

                              $"magicType2 : {magicType2} \n" +

                              $"magicType3 : {magicType3} \n" +

                              $"magicType4 : {magicType4}");

            //magicType0: Wood

            //magicType1 : Fire

            //magicType2 : Earth

            //magicType3 : Metal

            //magicType4 : Water

            Console.WriteLine("(int[])Enum.GetValues(typeof(MagicType)) ========================");

            int[] MagicTypeValues = (int[])Enum.GetValues(typeof(MagicType));

            Console.WriteLine("MagicType Enum Values");

            foreach (int value in MagicTypeValues)

            {

                Console.WriteLine(value);

            }

            ////MagicType Enum Values

            ////0

            ////1

            ////2

            ////3

            ////4

            //(int[])Enum.GetValues(typeof(MagicType))   list Enum underlying type values

            Console.WriteLine("Enum.GetNames(typeof(MagicType)) ========================");

            string[] MagicTypeNames = Enum.GetNames(typeof(MagicType));

            Console.WriteLine("MagicType Enum Names");

            foreach (string name in MagicTypeNames)

            {

                Console.WriteLine(name);

            }

            ////MagicType Enum Names

            ////Wood

            ////Fire

            ////Earth

            ////Metal

            ////Water

            //Enum.GetNames(typeof(MagicType))  list Enum underlying type names

            Console.ReadLine();

        }

    }

}

namespace OnLineGame

{

    public class Gamer

    {

        public string Name { get; set; }

        public int Gender { get; set; }

        public List<Magic> MagicList { get; set; }

        public static string GetGender(int gender)

        {

            switch (gender)

            {

                case 0:

                    return "Unknown";

                case 1:

                    return "Male";

                case 2:

                    return "Female";

                default:

                    //Run default only if the value does not match any of case.

                    return "Invalid";

            }

        }

        public static string GamerInfo(Gamer gamer)

        {

            return $"Name : {gamer.Name} \nGender: {gamer.Gender} \nMagicList : {gamer.MagicList} ";

        }

        public static string GamerInfo2(Gamer gamer)

        {

            return $"Name : {gamer.Name} \nGender: {GetGender(gamer.Gender)} \nMagicList : {gamer.MagicList} ";

        }

    }

    public class Magic

    {

        public string MagicName { get; set; }

        public int MpCost { get; set; }

        public MagicType MagicType { get; set; }

        public static string MagicInfo(Magic magic)

        {

            return $"MagicName : {magic.MagicName} \nMpCost: {magic.MpCost} \nMagicType : {magic.MagicType} ";

        }

    }

    public enum MagicType   // : int

    {

        Wood,

        Fire,

        Earth,

        Metal,

        Water

    }

    public enum MagicType2 : short

    {

        Wood = 5,

        Fire,   //6

        Earth   //7

    }

    public enum MagicType3 : short

    {

        Wood = 8,

        Fire = 9,

        Earth = 10

    }

    public enum MagicType4 : short

    {

        Wood = 8,

        Fire = 100,

        Earth = 20

    }

}

/\*

1.

Enum

---------------------

1.1.

Using Enum keyword to create enumerations and it is strongly value typed constants.

The default underlying type of an enum is int.

You may use " : short " to set the underlying type of an enum is short.

The default value for first element is ZERO and gets incremented by 1.

---------------------

1.2.

Syntax :

//public enum EnumName  [ : underlyingType ]

//{

//    EnumValue1 [ = StarValue],

//    EnumValue2,

//    EnumValue3 [ = SpecificValue],

//    ....

//}

E.g.1.

//public enum MagicType   // : int

//{

//    Wood,

//    Fire,

//    Earth,

//    Metal,

//    Water

//}

E.g.2.

//public enum MagicType2 : short

//{

//    Wood = 5,

//    Fire,   //6

//    Earth   //7

//}

E.g.3.

//public enum MagicType4 : short

//{

//    Wood = 8,

//    Fire = 100,

//    Earth = 20

//}

---------------------

1.3.

//int woodInt = (int)MagicType.Wood;

Convert Enum to int

---------------------

1.4.

//MagicType magicType1 = (MagicType)1;

Convert int to Enum

---------------------

1.5.

Enum.GetValues list Enum underlying type values.

E.g.

int[] MagicTypeValues = (int[])Enum.GetValues(typeof(MagicType));

//MagicTypeValues == {0,1,2,3,4}

---------------------

1.6.

Enum.GetNames list Enum underlying type names.

string[] MagicTypeNames = Enum.GetNames(typeof(MagicType));

//MagicTypeNames == {"Wood","Fire","Earth","Metal","Water"}

\*/

Text

Description automatically generated

Text

Description automatically generated