**Módulo: Características del Lenguaje C#**

**Tutorial: Tipo de Datos Enumerados**

**Ejemplo 1**

1. Definición de enumerados (lista de Constantes). Definir la siguiente lista de enumerados:

namespace Enumeracion

{

//1.Definir la lista de enumerados

enum Colores

{

Rojo,

Verde,

Azul

}

1. Declarar variables que usa la lista de enumerados

static void Main(string[] args)

{

// 1. Declaracion de variables de enumerados

Colores MiColorA = Colores.Verde;

Colores MiColorB = Colores.Rojo

Console.WriteLine(MiColorA);

1. Agregar la sentencia para mostrar el valor numérico del enumerado.

Console.WriteLine((int)MiColorB);

Observar el valor de salida

1. Se puede asignar valores numéricos a cada componente de la lista de enumerados. Modificar la definición de enumerados:

enum Colores

{

Rojo = -1,

Verde = 5,

Azul = 12

}

1. Compilar y observar la salida.
2. Los componentes de tipo enumerado por defecto son te tipo integer. Modificar la definición para tipo, short y byte; compilar cada vez y observar el resultado.
3. Modificación de la declaración de enumerados mediante el uso de atributos. Definir un grupo de enumerados de la siguiente manera:

[Flags()]

enum Colores : sbyte

{

Rojo = 0,

Verde = 1,

Azul = 2,

Amarillo = 3

}

1. Se declaran las variables de Colores:

// 2. Operaciones con Enumerados

Colores MiColorA = Colores.Verde;

Colores MiColorB = Colores.Azul;

1. Se realiza la operación con las variables (Operación OR). Compilar y ejecutar.

Colores MiColorBitWihise = MiColorA | MiColorB;

Console.WriteLine(MiColorBitWihise);

Console.ReadLine();

1. Modificar los valores y usar la operación AND &
2. Mostrar el valor numérico.
3. Sentencia correspondientes:

Colores MiColorA = Colores.Verde;

Colores MiColorB = Colores.Azul;

Colores MiColorBitWihise = MiColorA | MiColorB;

Console.WriteLine("Operacion OR: {0} - {1}", MiColorBitWihise,(sbyte)MiColorBitWihise);

Colores MiColorBitWihiseB = MiColorA & MiColorB;

Console.WriteLine("Operacion AND: {0} - {1}", MiColorBitWihiseB, (sbyte)MiColorBitWihiseB);

Console.ReadLine();

1. Realizar un nueva declaración de variable, compilar y ver el resultado.

Colores MicolorC = 5;

1. Convertirlo a tipo de dato enumerado y mostrar la salida. Observar el resultado

Colores MiColorC = (Colores)5;

Console.WriteLine(MiColorC);

1. Modificar la asignación con el valor 3 y ver el resultado obtenido.

Ejemplo 2

1. Como tratar lo valores de una lista de enumerados no contemplado
2. Crear un método que tratará la evaluación del contenido del tipo enumerado

public Program() { }

public void Trabajar(Colores color)

{

}

1. Agregar el código del método para evaluar la variable color

public void Trabajar(Colores color)

{

if (Enum.IsDefined(typeof(Colores), color)) //

Console.WriteLine(color);

else

Console.WriteLine("Tipo no definido, ERROR!");

}

1. En el main agregar la llamada al método que evalúa la definición de la variable enumerada.

static void Main(string[] args)

{

//3. Asignaciones de valores no contemplados en la lista de enumerados

Colores MiColorA = Colores.Verde;

Colores MiColorB = Colores.Azul;

Colores MiColorC = (Colores)5;

Console.WriteLine(MiColorC);

//4. Evaluación de contenido de la variable enumerada

Program p = new Program();

p.Trabajar(MiColorC);

Console.ReadLine();

}

1. Ejecutar y observar el resultado, con valores 5 y 3.