Tristan Izlar

COP2362

TUTORIAL 2-3: Creating Value Types with Enumerations and Structures

I worked alone.

A picture containing graphical user interface

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Data.cs:

using System;

namespace StructsAndEnums

{

class Date

{

private int year;

private Month month;

private int day;

public Date(int ccyy, Month mm, int dd)

{

this.year = ccyy - 1900;

this.month = mm;

this.day = dd - 1;

}

public override string ToString()

{

string data = $"{this.month} {this.day + 1} {this.year + 1900}";

return data;

}

public void AdvanceMonth()

{

this.month++;

if(this.month == Month.December + 1)

{

this.month = Month.January;

this.year++;

}

}

}

}

Program.cs:

#region Using directives

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

#endregion

namespace StructsAndEnums

{

class Program

{

static void doWork()

{

//Month first = Month.December;

//Console.WriteLine(first);

//first++;

//Console.WriteLine(first);

//Date defaultDate = new Date();

//Console.WriteLine(defaultDate);

Date weddingAnniversary = new Date(2017, Month.July, 4);

Console.WriteLine(weddingAnniversary);

Date weddingAnniversaryCopy = weddingAnniversary;

Console.WriteLine($"Value of copy is {weddingAnniversaryCopy}");

weddingAnniversary.AdvanceMonth();

Console.WriteLine($"New value of weddingAnniversary is {weddingAnniversary}");

Console.WriteLine($"Value of copy is still {weddingAnniversaryCopy}");

}

static void Main()

{

try

{

doWork();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

}

}

Month.cs:

using System;

namespace StructsAndEnums

{

enum Month

{

January, February, March, April,

May, June, July, August,

September, October, November, December

}

}