Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

Кафедра системного проектування

Звіт

**Про виконання лабораторної роботи № 3**

**Структури та перечислення**

Виконав:

Студент ФеП-11

Козуля Олексій

Перевірив:

Щербак С. С.

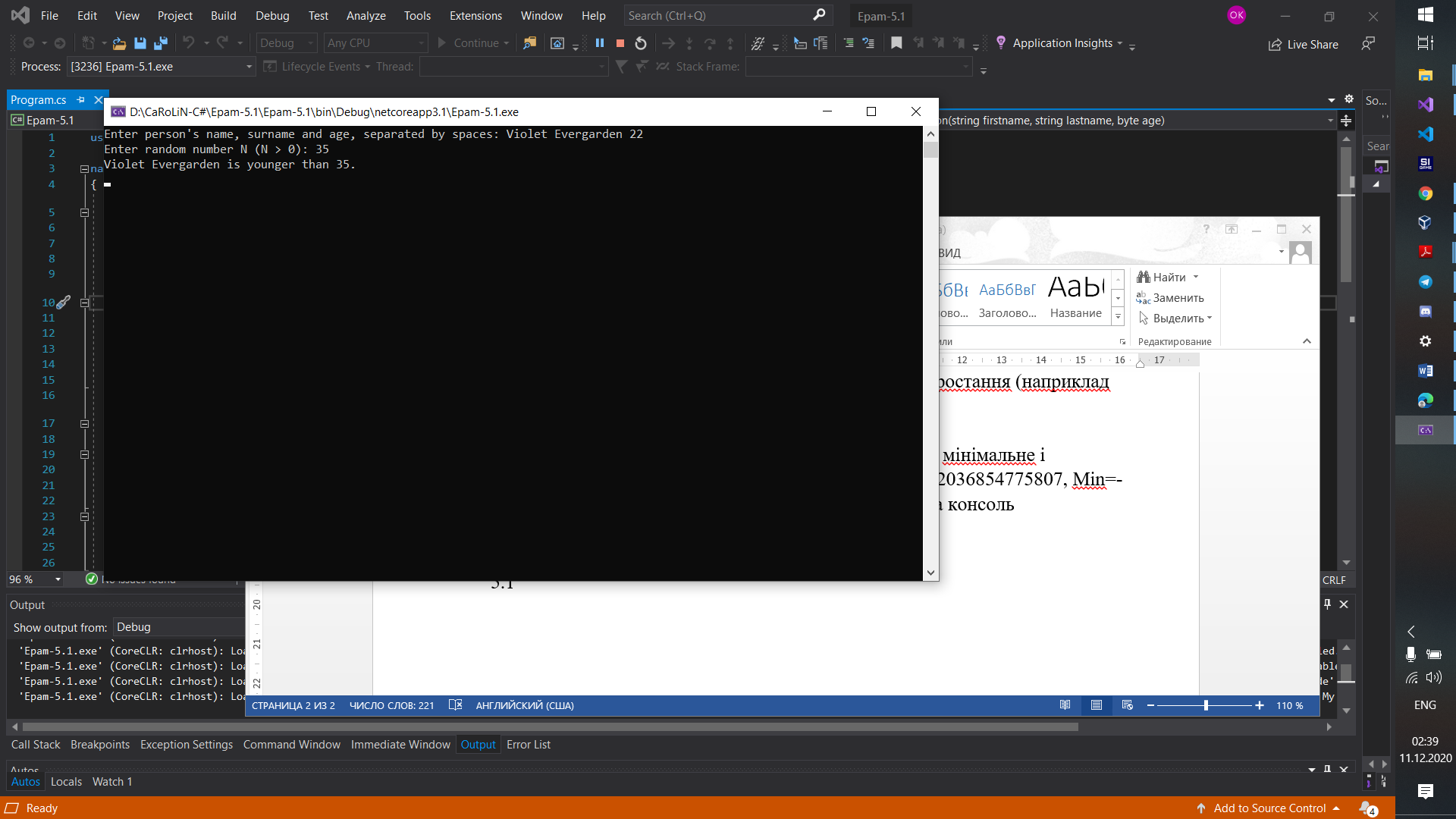
**Львів 2020**

**Мета:** дослідити структури та механізми роботи. Дослідити перерахування (enumerations).

**Хід роботи**

1. Реалізувати структуру Person в якій зберігається ім'я, прізвище та вік людини. Реалізувати метод для створеної структури, який приймає ціле значення “n” (n > 0) та повертає відформатований string: - "{Name} {Surname} older than {n}" якщо вік персони більший від заданого “n“ - "{Name} {Surname} younger than {n}" якщо вік персони менший від заданого “n”
2. Створити інтерфейси ISize (з властивостями Width та Height і методом Perimeter) та ICoordinates (з властивостями X, Y). Створити структуру Rectangle що реалізує дані інтерфейси.
3. Створити enum який включає в себе всі місяці року. Зчитати з консолі значення "n" (0 <= n < 12). Вивести на консоль місяць який відповідає значенню "n".
4. Створити enum який включає в себе декілька кольорів. Замінити стандартні значення на випадкові (наприклад "Red=4, Blue=15, Green=1..."). Написати розширювальний метод який виведе на консоль всі значення зі створеного переліку у порядку зростання (наприклад "Green=1, Red=4, Blue=15, ...").
5. Створити enum LongRange в якому знаходяться мінімальне і максимальне значення типу long ("Max=9223372036854775807, Min=-9223372036854775808"). Вивести ці значення на консоль використовуючи створений enum.

5.1



Code:

using System;

namespace lab\_5.\_1

{

struct Person

{

private string \_firstname, \_lastname;

private byte \_age;

public Person(string firstname, string lastname, byte age)

{

\_firstname = firstname;

\_lastname = lastname;

\_age = age;

}

public string IsOlderThan(byte n)

{

if (\_age > n)

{

return $"{\_firstname} {\_lastname} is older than {n}.";

}

if (\_age < n)

{

return $"{\_firstname} {\_lastname} is younger than {n}.";

}

return $"{\_firstname} {\_lastname} is exactly {n} years old.";

}

}

class Program

{

static void Main(string[] args)

{

Console.Write("Enter person's name, surname and age, separated by spaces: ");

var input = Console.ReadLine().Split(' ');

var person = new Person(input[0], input[1], byte.Parse(input[2]));

Console.Write("Enter random number N (N > 0): ");

var n = byte.Parse(Console.ReadLine());

Console.WriteLine(person.IsOlderThan(n));

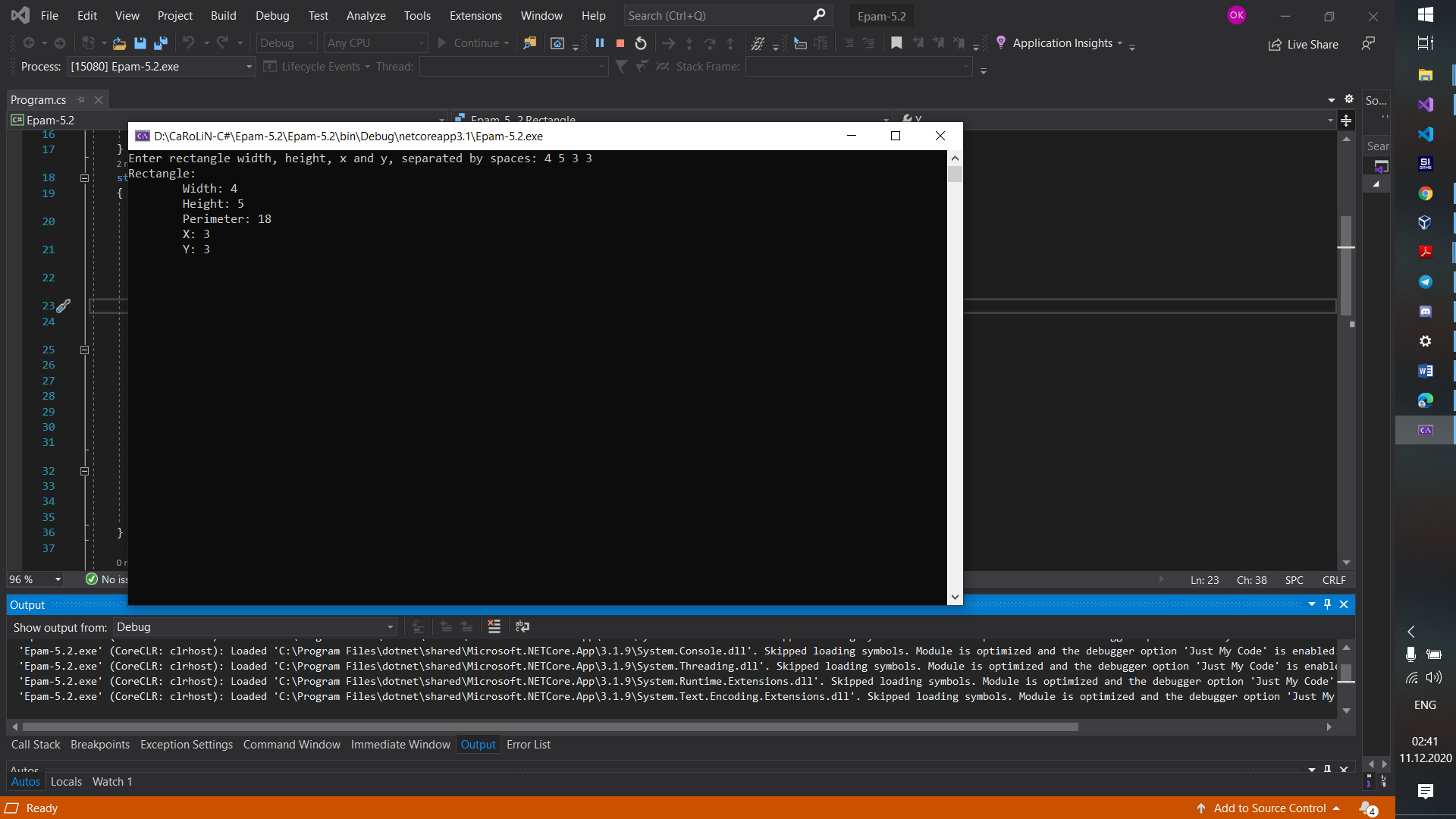
Console.ReadKey();

}

}

}

5.2



Code:

using System;

namespace Epam\_5.\_2

{

interface ICoordinates

{

double X { get; set; }

double Y { get; set; }

}

interface ISize

{

double Width { get; set; }

double Height { get; set; }

double Perimeter();

}

struct Rectangle : ISize, ICoordinates

{

public double Width { get; set; }

public double Height { get; set; }

public double X { get; set; }

public double Y { get; set; }

public Rectangle(double width, double height, double x, double y)

{

Width = width;

Height = height;

X = x;

Y = y;

}

public double Perimeter()

{

return 2 \* Width + 2 \* Height;

}

}

class Program

{

static void Main(string[] args)

{

Console.Write("Enter rectangle width, height, x and y, separated by spaces: ");

var input = Console.ReadLine().Split(' ');

var rect = new Rectangle(double.Parse(input[0]),

double.Parse(input[1]),

double.Parse(input[2]),

double.Parse(input[3]));

Console.WriteLine(

"Rectangle:\n\t" +

$"Width: {rect.Width}\n\t" +

$"Height: {rect.Height}\n\t" +

$"Perimeter: {rect.Perimeter()}\n\t" +

$"X: {rect.X}\n\t" +

$"Y: {rect.Y}");

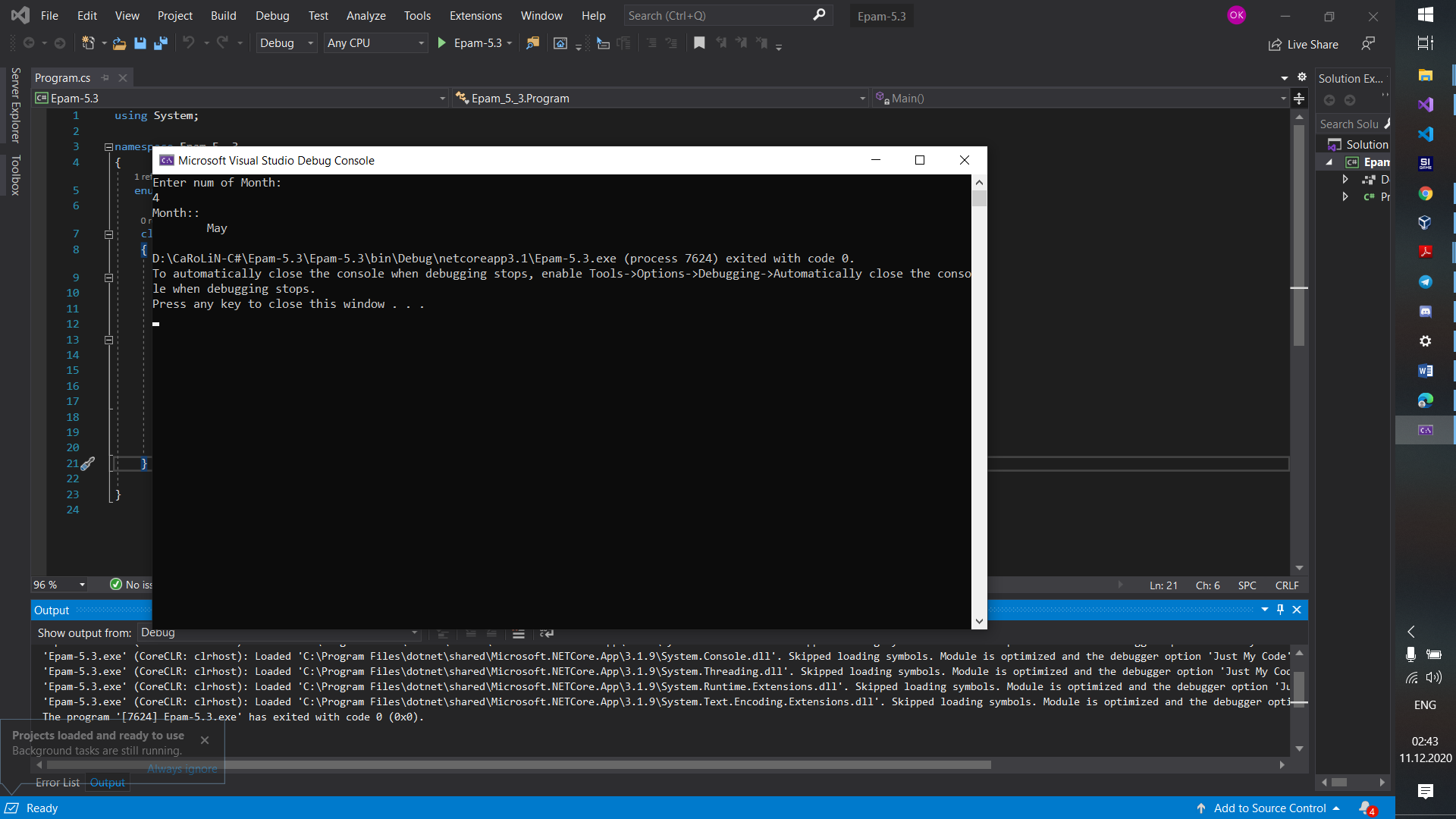
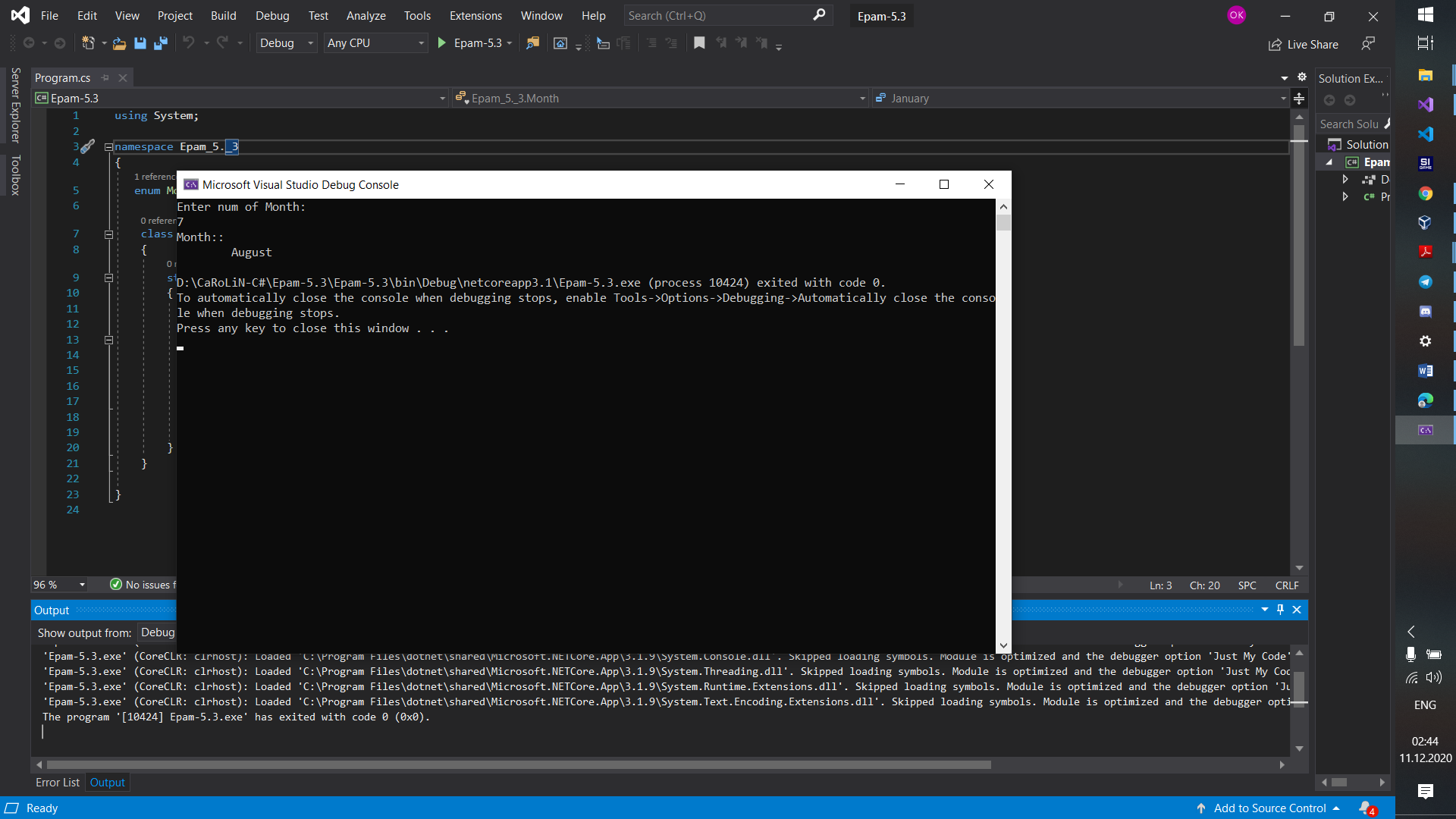
Console.ReadKey();

}

}

}

5.3

Code:

using System;

namespace Epam\_5.\_3

{

enum Month {January, Febuary, March, April, May, June, July, August, September, October, November, December }

class Program

{

static void Main()

{

Console.WriteLine("Enter num of Month: ");

int numMonth = int.Parse(Console.ReadLine());

if (numMonth > 12 || numMonth < 0)

{

Console.WriteLine("False num Enter 0-12");

return;

}

string month = Enum.GetName(typeof(Month), numMonth);

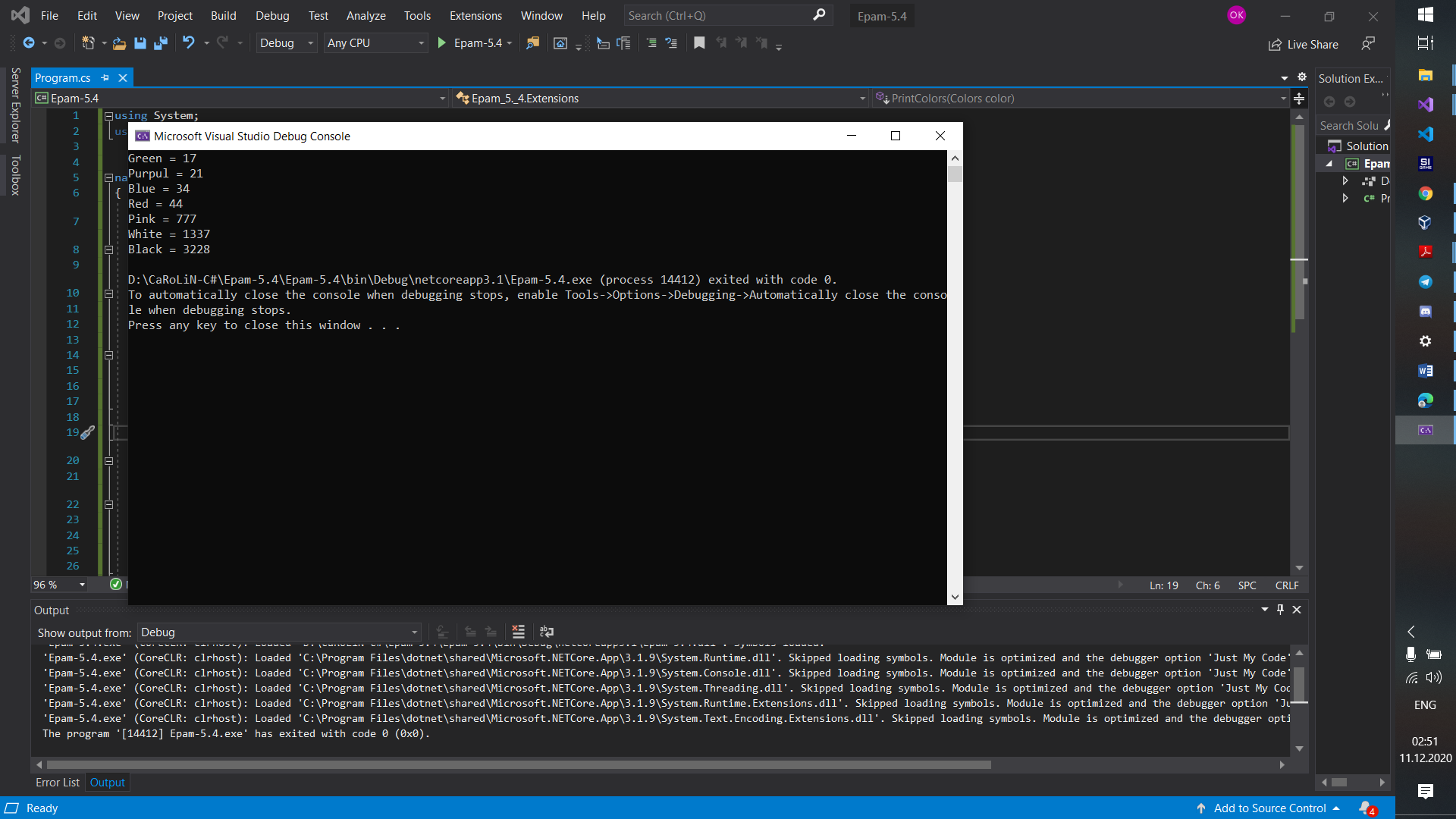
Console.WriteLine("Month:: \n" + "\t" + month);

}

}

}

5.4



Code:

using System;

using System.Collections.Generic;

namespace Epam\_5.\_4

{

public enum Colors { Red = 44, Blue = 34, Green = 17, Purpul = 21, Pink = 777, Black = 3228, White = 1337 }

public static class Extensions

{

public static void PrintColors(this Colors color)

{

int[] values = (int[])Enum.GetValues(color.GetType());

Array.Sort(values);

for (int i = 0; i < values.Length; i++)

{

Console.WriteLine("{0} = {1}", (Colors)values[i], values[i]);

}

}

}

class Program

{

static void Main(string[] args)

{

Colors someColor = Colors.Red;

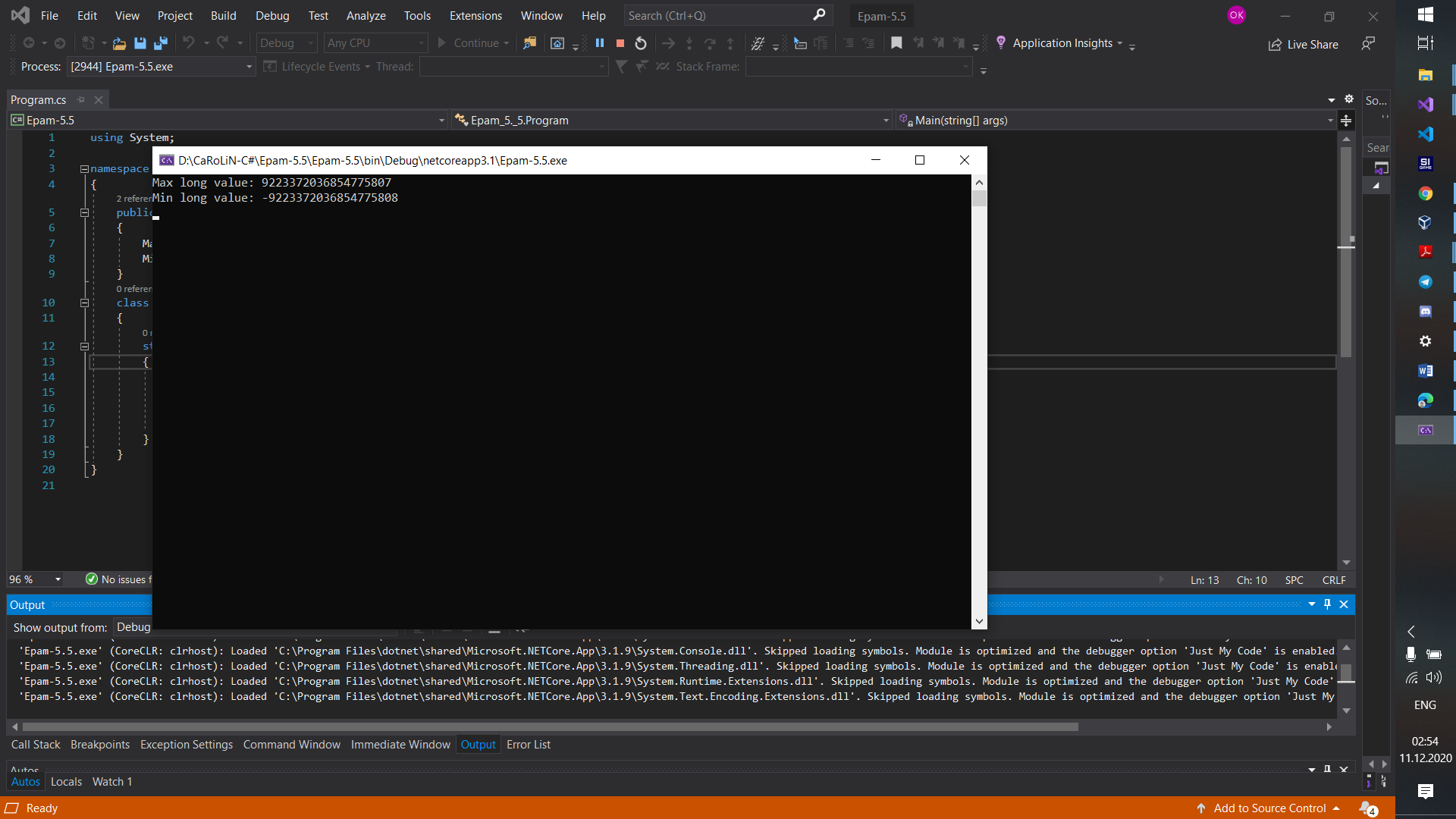
someColor.PrintColors();

}

}

}

5.5



using System;

namespace Epam\_5.\_5

{

public enum LongRange : long

{

Max = 9223372036854775807,

Min = -9223372036854775808

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine($"Max long value: {(long)LongRange.Max}");

Console.WriteLine($"Min long value: {(long)LongRange.Min}");

Console.ReadKey();

}

}

}

**Висновок:** ознайомився і вивчив структури та механізми роботи. Дослідив перерахування (enumerations).