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 № 4**

**Наслідування та інтерфейси**

Виконав:

Студент ФеП-11

Козуля Олексій

Перевірив:

Щербак С. С.

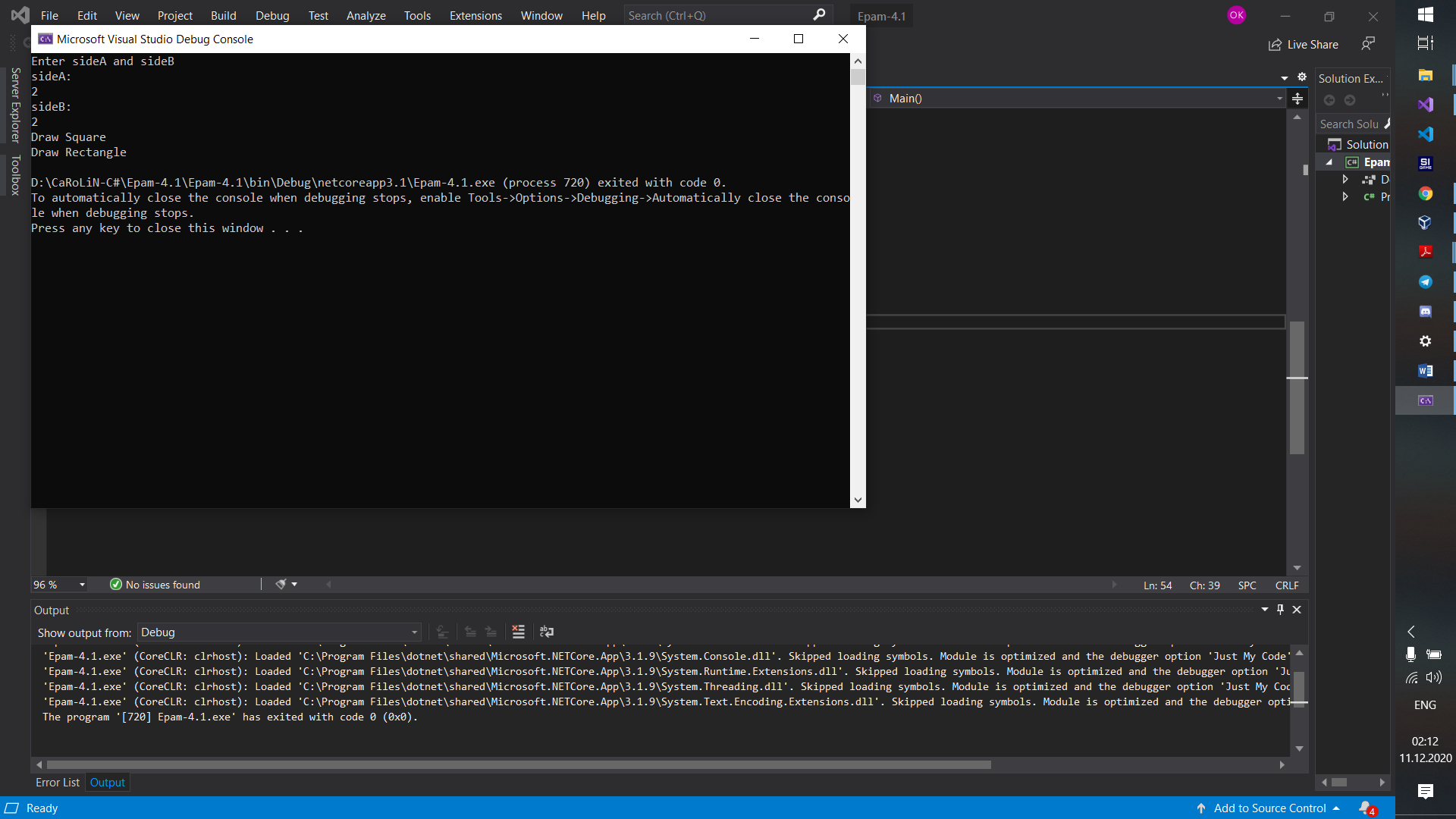
**Львів 2020**

**Мета:** вивчити поняття та механізми роботи наслідуання C#. Ознайомитись з пиняттями абстрактних класів та інтерфейсів

**Хід роботи**

1. Реалізувати базовий клас Figure з абстрактним методом Draw(). Створити класи Square та Rectangle (унаслідовані від Figure) в яких перегружена імплементація методу Draw() виводить на консоль назву класу в якому даний метод реалізований.
2. Використовуючи попереднє завдання добавити в клас Figure readonly проперті X та Y. Також, в клас Figure добавити параметризований конструктор що установляє значення в цих пропертях. Поправити унаслідовані класи щоб вони коректно працювали викликаючи базовий конструктор.
3. Використовуючи попереднє завдання змінити абстрактний метод Draw() на віртуальний і реалізувати його аналогічно як у класах Square та Rectangle. Переконатись що об'єкт класу Figure тепер може бути ініціалізований і виклики методу Draw() на об'єктах створених класів приведуть до виклику трьох різних реалізацій.
4. Використовуючи попереднє завдання винести метод Draw() у інтерфейс IDrawable. Створити метод DrawAll(params IDrawable[] array) в який передати об'єкти створених класів і в циклі викликати метод Draw() для кожного з них. Переконатись що були викликані коректні реалізації методу Draw() для кожного з класів.

4.1



Code:

using System;

using System.Reflection;

namespace Epam\_4.\_1

{

abstract class Figure

{

readonly int xxx;

readonly int yyy;

public Figure(int xxx, int yyy)

{

this.xxx = xxx;

this.yyy = yyy;

}

public abstract void Draw();

}

class Square : Figure

{

public Square(int aaa) : base(aaa, aaa) { }

public override void Draw()

{

Console.WriteLine("Draw Square");

}

}

class Rectangle : Figure

{

public Rectangle(int xxx, int yyy) : base(xxx, yyy) { }

public override void Draw()

{

Console.WriteLine("Draw Rectangle");

}

}

public static class Program

{

public static void Main()

{

Console.WriteLine("Enter sideA and sideB");

Console.WriteLine("sideA: ");

int \_x = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("sideB: ");

int \_y = Convert.ToInt32(Console.ReadLine());

Square a = new Square(\_x);

a.Draw();

Rectangle b = new Rectangle(\_x, \_y);

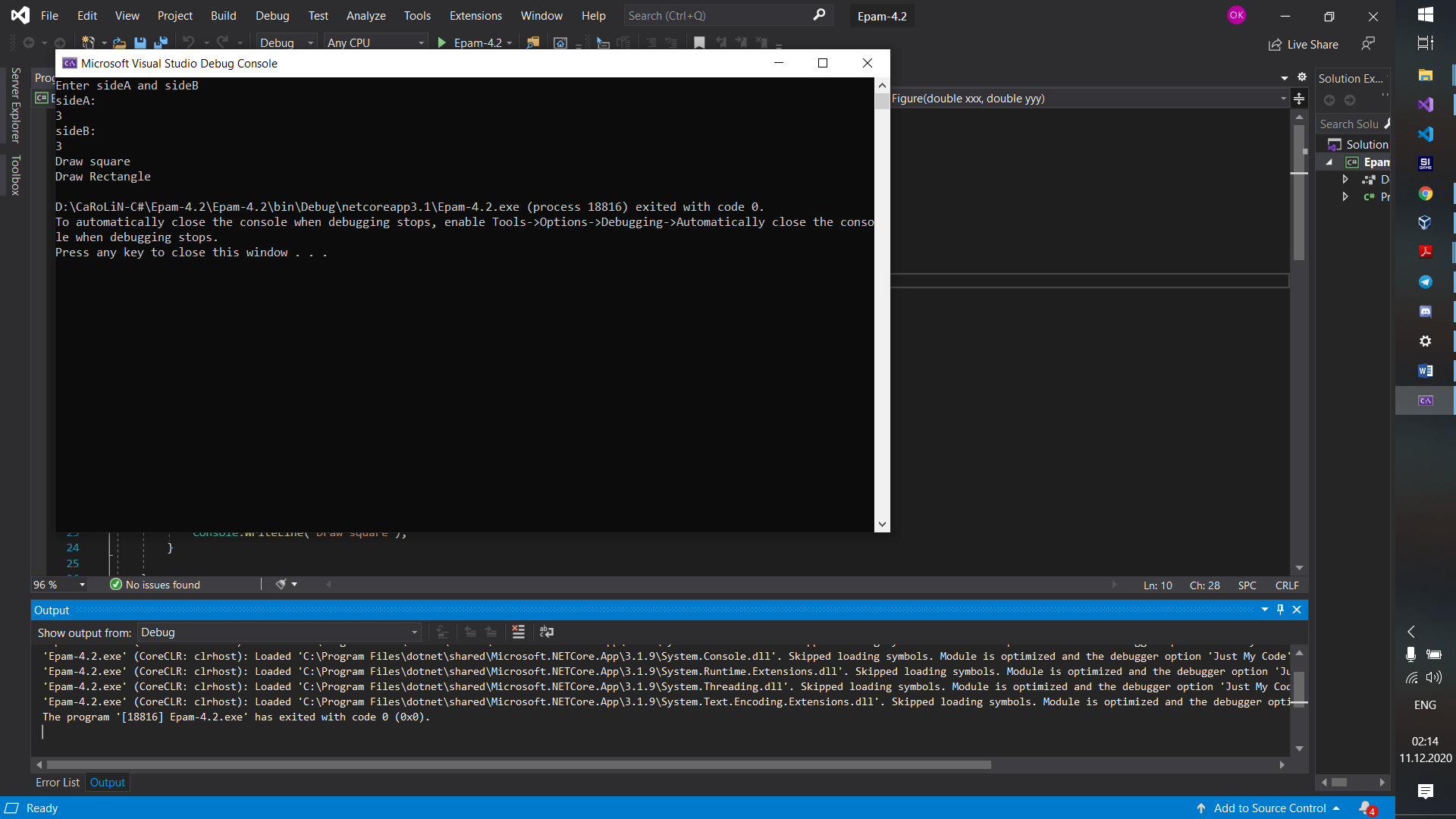
b.Draw();

}

}

}

4.2



Code:

using System;

namespace Epam\_4.\_2

{

public class Figure

{

readonly double xxx, yyy;

public Figure(double xxx, double yyy)

{

this.xxx = xxx;

this.yyy = yyy;

}

public virtual void Draw() { }

}

class Square : Figure

{

public Square(double aaa) : base(aaa, aaa) { }

public override void Draw()

{

base.Draw();

Console.WriteLine("Draw square");

}

}

class Rectangle : Figure

{

public Rectangle(double xxx, double yyy) : base(xxx, yyy) { }

public override void Draw()

{

base.Draw();

Console.WriteLine("Draw Rectangle");

}

}

public static class Program

{

static void Main()

{

Console.WriteLine("Enter sideA and sideB");

Console.WriteLine("sideA: ");

double \_x = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("sideB: ");

double \_y = Convert.ToDouble(Console.ReadLine());

Square a = new Square(\_x);

a.Draw();

Rectangle b = new Rectangle(\_x, \_y);

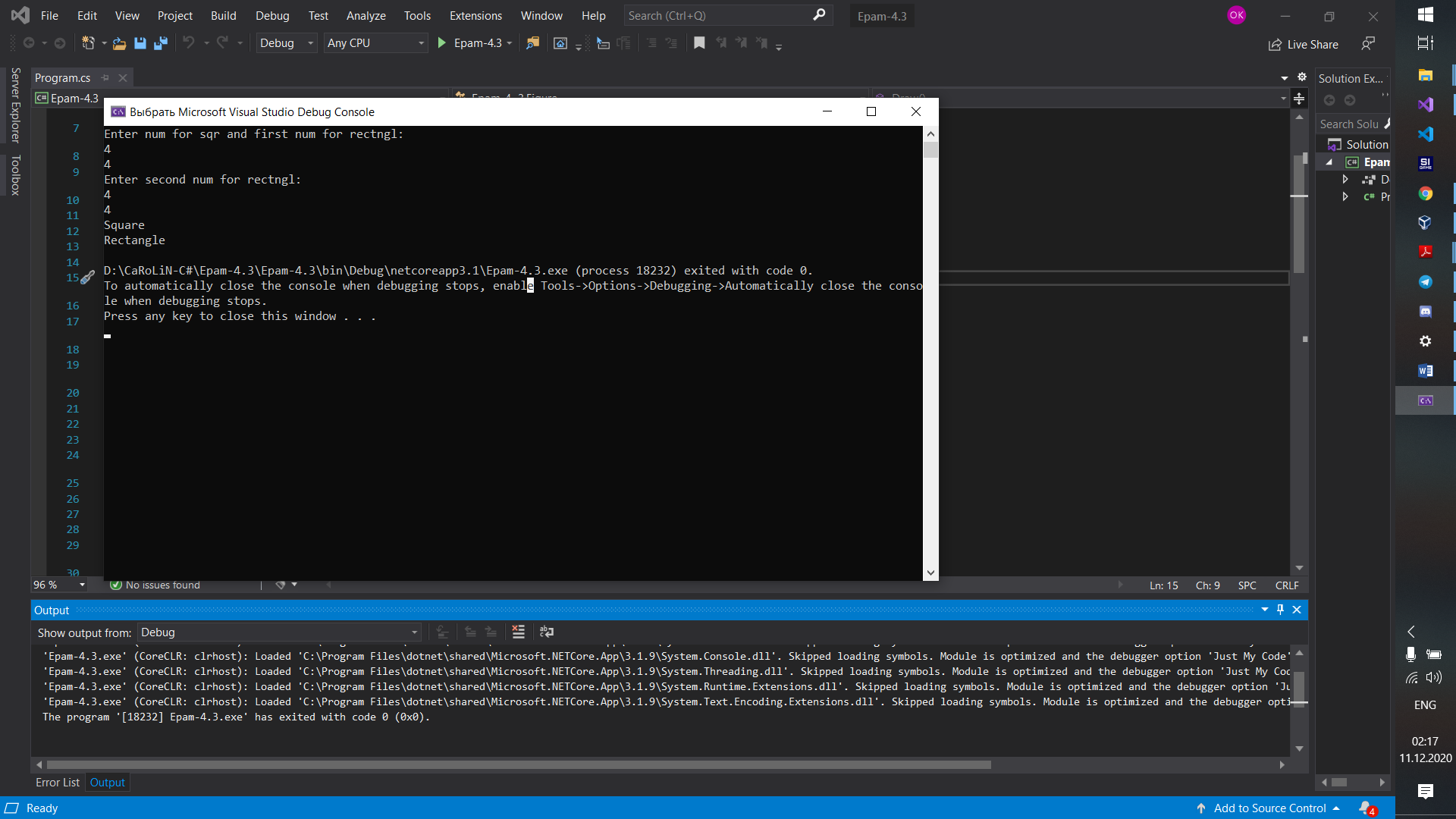
b.Draw();

}

}

}

4.3



Code:

using System;

namespace Epam\_4.\_3

{

abstract class Figure

{

public float xxx { get; private set; }

public float yyy { get; private set; }

protected Figure(float xxx, float yyy)

{

this.xxx = xxx;

this.yyy = yyy;

}

public abstract void Draw();

}

class Rectangle : Figure

{

public Rectangle(float xxx, float yyy) : base(xxx, yyy)

{

}

public override void Draw()

{

Console.WriteLine("Rectangle");

}

}

class Square : Figure

{

public Square(float xxx, float yyy) : base(xxx, yyy)

{

}

public override void Draw()

{

Console.WriteLine("Square");

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter num for sqr and first num for rectngl: ");

Rectangle rect = new Rectangle(float.Parse(Console.ReadLine()), float.Parse(Console.ReadLine()));

Console.WriteLine("Enter second num for rectngl: ");

Square sq = new Square(float.Parse(Console.ReadLine()), float.Parse(Console.ReadLine()));

sq.Draw();

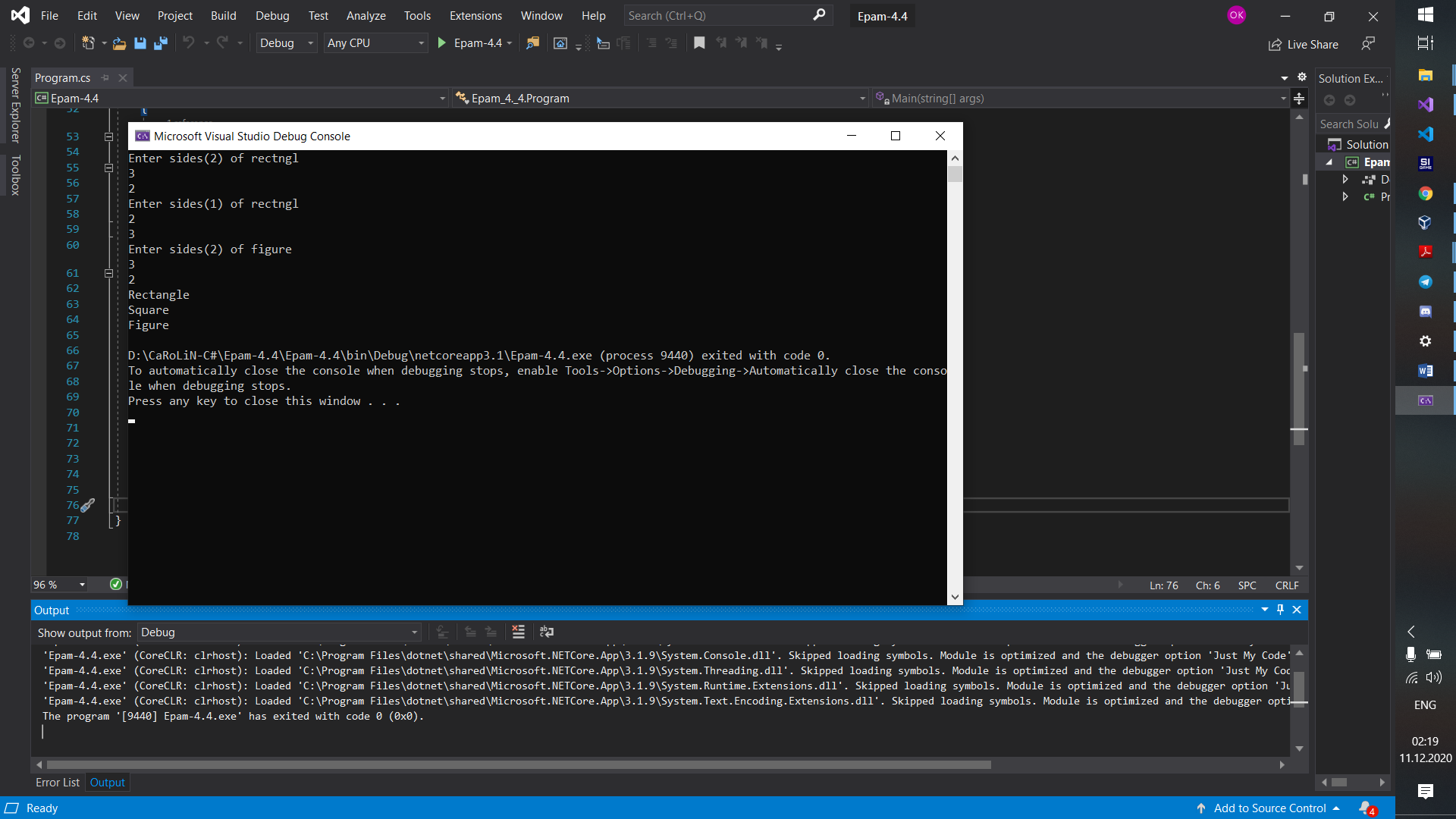
rect.Draw();

}

}

}

4.4



Code:

using System;

using System.Collections.Generic;

using System.Text;

namespace Epam\_4.\_4

{

interface IDrawable

{

void Draw();

}

class Figure : IDrawable

{

public float xxx { get; private set; }

public float yyy { get; private set; }

public Figure(float xxx, float yyy)

{

this.xxx = xxx;

this.yyy = yyy;

}

void IDrawable.Draw()

{

Console.WriteLine("Figure");

}

}

class Rectangle : Figure, IDrawable

{

public Rectangle(float xxx, float yyy) : base(xxx, yyy)

{

}

void IDrawable.Draw()

{

Console.WriteLine("Rectangle");

}

}

class Square : Figure, IDrawable

{

public Square(float xxx, float yyy) : base(xxx, yyy)

{

}

void IDrawable.Draw()

{

Console.WriteLine("Square");

}

}

class Program

{

static void DrawAll(params IDrawable[] array)

{

foreach (IDrawable drawable in array)

{

drawable.Draw();

}

}

static void Main(string[] args)

{

Console.WriteLine("Enter sides(2) of rectngl");

Rectangle rect = new Rectangle(float.Parse(Console.ReadLine()), float.Parse(Console.ReadLine()));

Console.WriteLine("Enter sides(1) of rectngl");

Square sq = new Square(float.Parse(Console.ReadLine()), float.Parse(Console.ReadLine()));

Console.WriteLine("Enter sides(2) of figure");

Figure fig = new Figure(float.Parse(Console.ReadLine()), float.Parse(Console.ReadLine()));

IDrawable rectDrawable = (IDrawable)rect;

IDrawable sqDrawable = (IDrawable)sq;

IDrawable figDrawable = (IDrawable)fig;

DrawAll(rectDrawable, sqDrawable, figDrawable);

}

}

}

**Висновки:** ознайомився та вивчив поняття та механізми роботи наслідуання C#. Ознайомивсь з пиняттями абстрактних класів та інтерфейсів.