**Міністерство освіти і науки України**

**Національний технічний університет**

**«Дніпровська політехніка»**



**ЗВІТ**

**про виконання лабораторних робіт**

**з дисципліни**

**«Об'єктно Орієнтоване Програмування»**

**Лабораторна робота № 4**

Виконав:

студент гр. 121-19-2

Дробний О.Ю.

Прийняв:

Приходченко С.Д.

**Дніпро**

**2020**

**Лабораторная работа № 4  
Вариант 6**

**Тема:** Изучение полиморфизма в классах C#

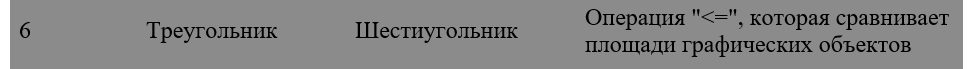
**Цель:** научиться создавать виртуальные методы; научиться перегружать арифметические операции и операции сравнения

**Выполнение работы**

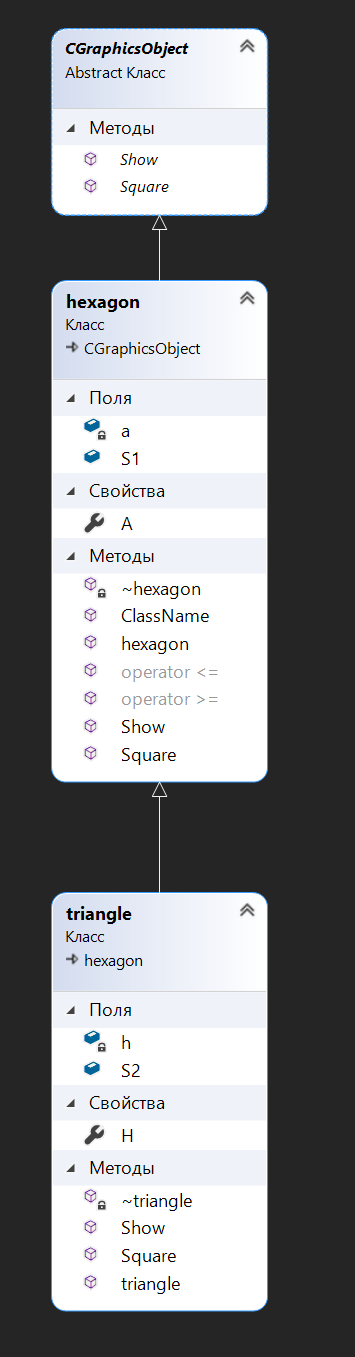
**Задание 1:**1. Используя навыки, полученные в результате выполнения лабораторной работы #3-OOP, выполнить моделирование и кодирование классов, представляющих собой простейшую классовую иерархию.

Классы должны содержать поля данных для описания соответствующих графических объектов, конструкторы, деструкторы, соответствующие set- и get-методы, виртуальный метод Show, выводящий на экран информацию о графическом объекте и его предках (если таковые имеются), и виртуальный метод GetClassName, возвращающий имя класса объекта.

2. Расширить возможности классов, входящих в классовую иерархию, за счёт перегрузки операций в соответствии с вариантом задания.



**1.Диаграмма классов**



**2.Код программы**

**Файл Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LR\_4.\_1\_Drobnyi

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a for hexagon:");

hexagon new\_hexagon = new hexagon(Convert.ToInt32(Console.ReadLine()));

new\_hexagon.Show();

new\_hexagon.ClassName();

new\_hexagon.Squre();

Console.WriteLine("Enter a,h for triangle:");

triangle triangle = new triangle(Convert.ToInt32(Console.ReadLine()), Convert.ToInt32(Console.ReadLine()));

triangle.Show();

triangle.ClassName();

triangle.Squre();

bool result = new\_hexagon.S1 >= triangle.S2;

Console.WriteLine("s1 >= s2:" + result.ToString());

Console.ReadKey();

}

}

}

**Файл hexagon.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace LR\_4.\_1\_Drobnyi

public class hexagon : CGraphicsObject

{

private int a;

public int A {

get { return a;}

set { a = value; }

} //поля класса + setter&getter

public double S1;

public hexagon(int \_a)

{

a = \_a;

} //конструктор класса

public override void Show() //метод вывода информации об объекте

{

Console.WriteLine("Предок:" + (this.GetType().BaseType).ToString());

Console.WriteLine("Абстрактный класс:" + (this.GetType().BaseType).BaseType.ToString());

Console.WriteLine("Объект:" + (this.GetType().Name).ToString());

}

public virtual void ClassName()

{

Console.WriteLine("Имя:" + (this.GetType().Name).ToString());

}

public override double Square() //метод вычисления площади

{

S1 = ((3 \* Math.Sqrt(3) \* Math.Pow(a, 2)) / 2);

Console.WriteLine("s1 = " + S1.ToString());

return S1;

}

public static bool operator >=(hexagon S1, triangle S2) //перегрузка оператора

{

return S1 >= S2;

}

public static bool operator <=(hexagon S1, triangle S2)

{

return S1 <= S2;

}

~hexagon()

{

Console.WriteLine("Destructor");

}// деструктор класса

}

}**Файл CGraphicsObject.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace LR\_4.\_1\_Drobnyi

{

public abstract class CGraphicsObject

{

abstract public void Show();

abstract public double Square();

}

}

**Файл triangle.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace LR\_4.\_1\_Drobnyi

{

public class triangle : hexagon

{

private int h;

public int H

{

get { return h; }

set { h = value; }

}

public double S2 = 0;

public triangle(int \_a, int \_h) : base(\_a)

{

A = \_a; H = \_h;

} //конструктор класса

public override void Show()

{

Console.WriteLine("Предок:" + (this.GetType().BaseType).ToString());

Console.WriteLine("Объект:" + (this.GetType().Name).ToString());

}

public override double Square()

{

S2 = (0.5 \* A \* H);

Console.WriteLine("s2 = " + S2.ToString());

return S2;

}

~triangle()

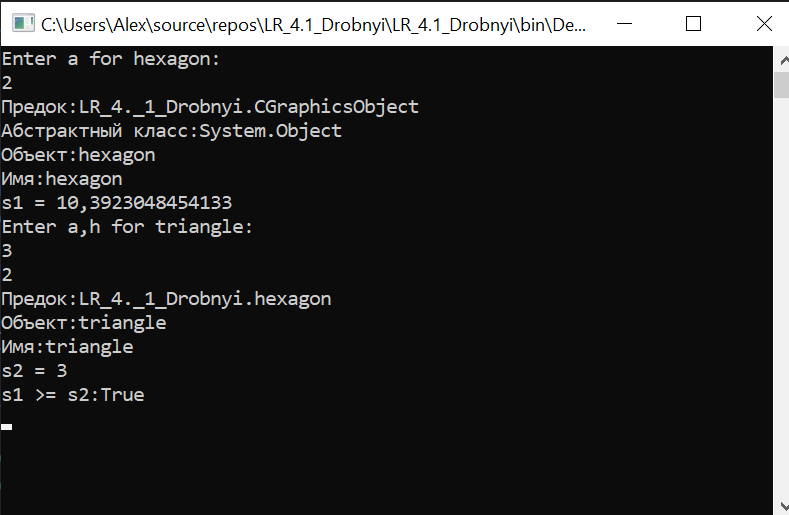
{

Console.WriteLine("Destructor");

}// деструктор класса

}

}**Результат работы програмы:**



**Задание 2:**

Проект создавать как проект Windows Forms. Модифицировать лабораторную №2 таким образом, чтобы на форме отображались объекты обоих классов (в простейшем случае – на pictureBox).

Создать классовую модель, которая включает в себя абстрактный класс CGraphicsObject, и его два наследника. Все классы, за исключением абстрактного, должны содержать конструкторы, деструкторы, соответствующие set- и get-методы, а также метод *Show*, который может иметь различную сигнатуру для разных классов и должен выводить на экран в текстовом виде информацию об объекте и его предках. Абстрактный класс должен содержать декларацию полей и методов.

Обязательно перегружать указанную в варианте операцию внутри класса.

На форме Windows  
  
*Примечание: абстрактный класс не нуждается в реализации методов.*

**Код программы**

**Файл Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace LR\_4.\_2\_Drobnyi

{

public partial class Form1 : Form

{

triangle new\_triangle = new triangle();

hexagon new\_hexagon = new hexagon();

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

pictureBox1.Refresh();

}

private void button2\_Click(object sender, EventArgs e)

{

pictureBox2.Refresh();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void pictureBox1\_Paint(object sender, PaintEventArgs e)

{

PointF[] shape = new PointF[new\_hexagon.n];

shape[0] = new PointF(0, 0);

shape[1] = new PointF(0, 0);

shape[2] = new PointF(0, 0);

shape[3] = new PointF(0, 0);

shape[4] = new PointF(0, 0);

shape[5] = new PointF(0, 0);

var graphics = e.Graphics;

var x = pictureBox1.Width / 2;

var y = pictureBox1.Height / 2;

var r = 70;

//создаем 6 точек

for (int a = 0; a < new\_hexagon.n; a++)

{

shape[a] = new PointF(

x + r \* (float)Math.Cos(a \* 60 \* Math.PI / 180f),

y + r \* (float)Math.Sin(a \* 60 \* Math.PI / 180f));

}

graphics.DrawPolygon(Pens.Red, shape);

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

}

private void pictureBox2\_Paint(object sender, PaintEventArgs e)

{

Point[] shape = new Point[new\_triangle.n];

shape[0] = new Point(200, 100);

shape[1] = new Point(300, 200);

shape[2] = new Point(100, 200);

var graphics = e.Graphics;

graphics.DrawPolygon(Pens.Red, shape);

}

}

}

**Файл hexagon.cs**

using System;

using System.Collections.Generic;

using System.Text;

using System.Drawing;

namespace LR\_4.\_2\_Drobnyi

{

public class hexagon : CGraphicsObject

{

public hexagon()

{

n = 6;

} //конструктор класса

}

}

**Файл triangle.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace LR\_4.\_2\_Drobnyi

{

public class triangle : CGraphicsObject

{

public triangle()

{

n = 3;

} //конструктор класса

}

}

**Файл CGraphicsObject.cs**

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace LR\_4.\_2\_Drobnyi

{

public abstract class CGraphicsObject

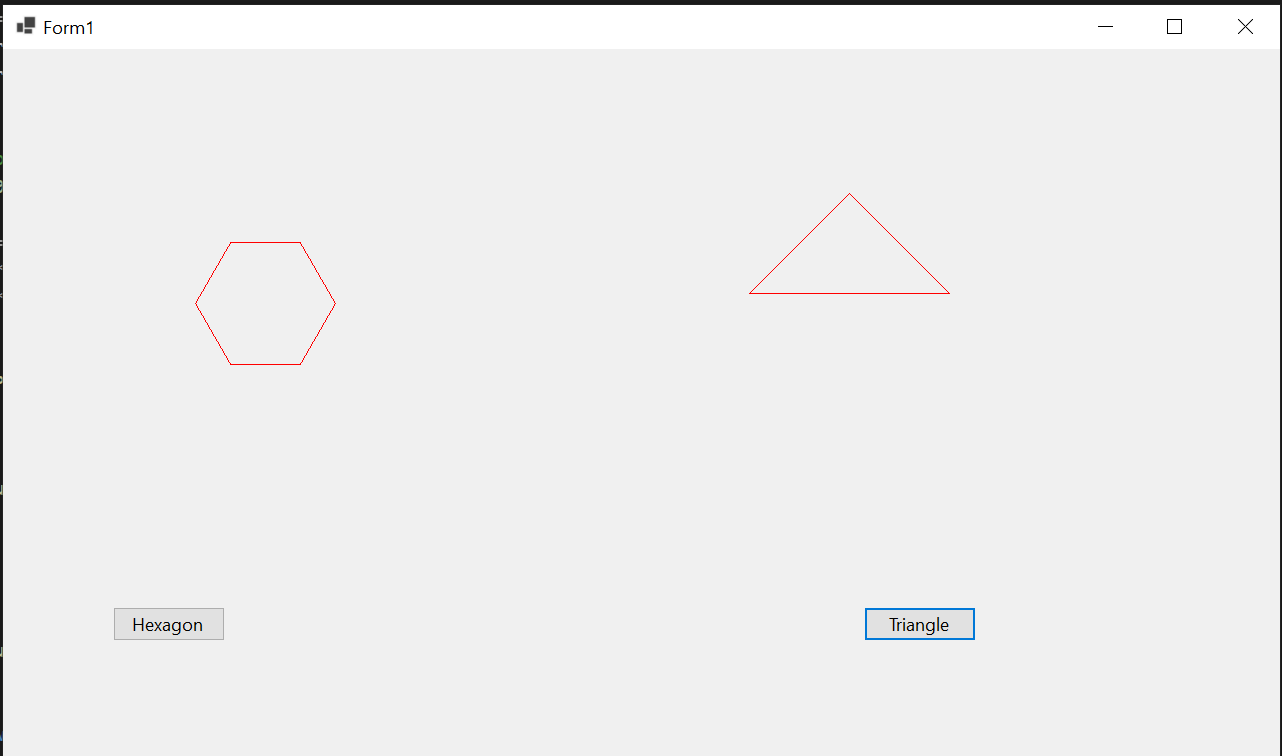
{

public int n { set; get; }

}

}

**Результат работы програмы:**



**Вывод:** научиться создавать виртуальные методы; научиться перегружать арифметические операции и операции сравнения; научился строить простейшие классовые модели;овладел методикой автоматической генерации классов С# из диаграмм классов;создал определение функций-членов классов с использованием дополнительных средств отображения графики в консоли.А также разработал програму для создания классовой модели, которая включает в себя абстрактный класс CGraphicsObject и его наследников, а так же научился отрисовывать графические объекты с помошью picturebox.