**Міністерство освіти і науки України**

**Національний технічний університет**

**«Дніпровська політехніка»**



**ЗВІТ**

**про виконання лабораторних робіт**

**з дисципліни**

**«Об'єктно Орієнтоване Програмування»**

**Лабораторна робота № 6**

Виконав:

студент гр. 121-19-2

Дробний О.Ю.

Прийняв:

Приходченко С.Д.

**Дніпро**

**2020**

**Лабораторная работа № 6  
Вариант 6**

**Тема:** Изучение интерфейсов, делегатов и коллекций в классах C#

**Цель:** научиться создавать интерфейсы, делегаты , коллекции ознакомится с понятием наследования интерфейсов , их модификаторами доступов, применением.

**Выполнение работы**

**Задание:**

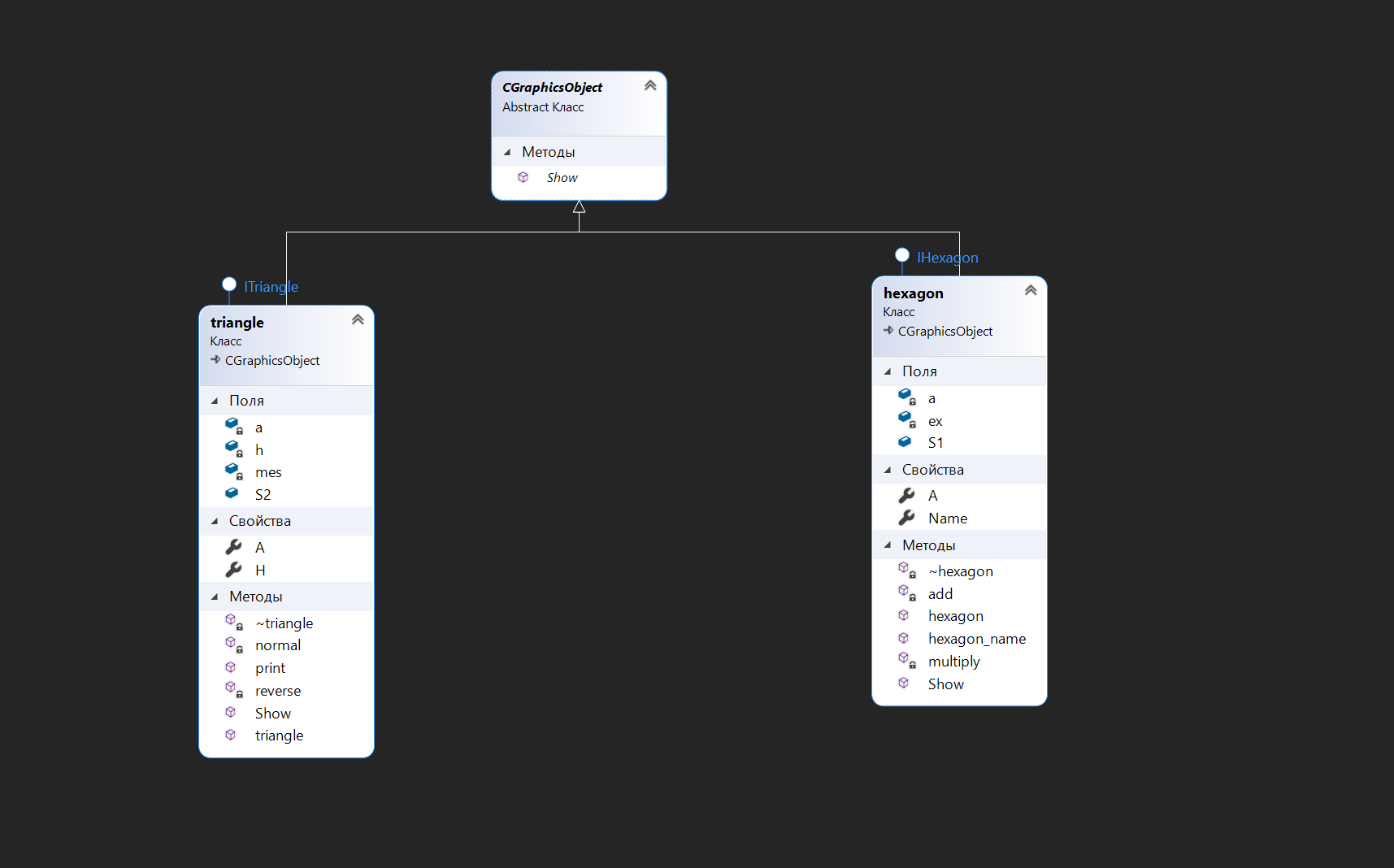
1. Создать классовую модель, которая включает в себя абстрактный класс CGraphicsObject, и его два наследника. Все классы, за исключением абстрактного, должны содержать конструкторы, деструкторы, соответствующие set- и get-методы, а также метод *Show*, который может иметь различную сигнатуру для разных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

2. Создать в классах-потомках по одному интерфейсу и делегату.

3. Создать простейшую коллекцию из экземпляров каждого из классов-потомков.



**Классовая модель:**



**Код программы:**

**Файл triangle.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApp3

{

delegate void Message(string one, string two); // Объявляем делегат

interface ITriangle

{

void print(); //метод интерфейса

}

public class triangle : CGraphicsObject, ITriangle

{

Message mes;//Создаем переменную делегата

private int a;

public int A

{

get { return a; }

set { a = value; }

} //поля класса + setter&getter

private int h;

public int H

{

get { return h; }

set { h = value; }

}

public double S2 = 0;

public triangle(int \_a, int \_h)

{

A = \_a; H = \_h;

if (\_a > 10){

mes = reverse;// Присваиваем этой переменной адрес метода

}

else

{

mes = normal;

}

mes("a<10", "a>10"); //Вызываем метод

} //конструктор класса

public override void Show()

{

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

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

}

public void print()

{

Console.WriteLine("Triangle in Triangle");

}

private static void normal(string one, string two)

{

Console.WriteLine(one);

}

private static void reverse(string one, string two)

{

Console.WriteLine(two);

}

~triangle()

{

Console.WriteLine("Destructor");

}// деструктор класса

}

}

**Файл hexagon.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApp3

{

delegate void Example(int one); // Объявляем делегат

interface IHexagon

{

string Name

{

get; set;

}

void hexagon\_name();//метод интерфейса

}

public class hexagon : CGraphicsObject , IHexagon

{

Example ex; //Создаем переменную делегата

public string Name { get; set; }

private int a;

public int A

{

get { return a; }

set { a = value; }

} //поля класса + setter&getter

public double S1;

public hexagon(int \_a, string name)

{

a = \_a;

Name = name;

if (\_a > 10)

{

ex = multiply; // Присваиваем этой переменной адрес метода

}

else

{

ex = add;

}

ex(a); //Вызываем метод

} //конструктор класса

public void hexagon\_name()

{

Console.WriteLine("Name:" + Name);

}

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());

}

private static void multiply(int a)

{

Console.WriteLine("a\*2=" + (a \* 2).ToString());

}

private static void add(int a)

{

Console.WriteLine("a+2=" + (a + 2).ToString());

}

~hexagon()

{

Console.WriteLine("Destructor");

}// деструктор класса

}

}

**Файл CGraphicsObject.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApp3

{

public abstract class CGraphicsObject

{

abstract public void Show();

}

}

**Файл Program.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections;

namespace ConsoleApp3

{

class Program

{

static void Main(string[] args)

{

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

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

new\_hexagon.Show();

new\_hexagon.hexagon\_name();

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

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

triangle.Show();

ArrayList objectList = new ArrayList() { new\_hexagon, triangle }; //создаем колекцию экземпляров классов наследников

foreach (object o in objectList) // выводим элементы коллекции

{

Console.WriteLine("\n" + o);

}

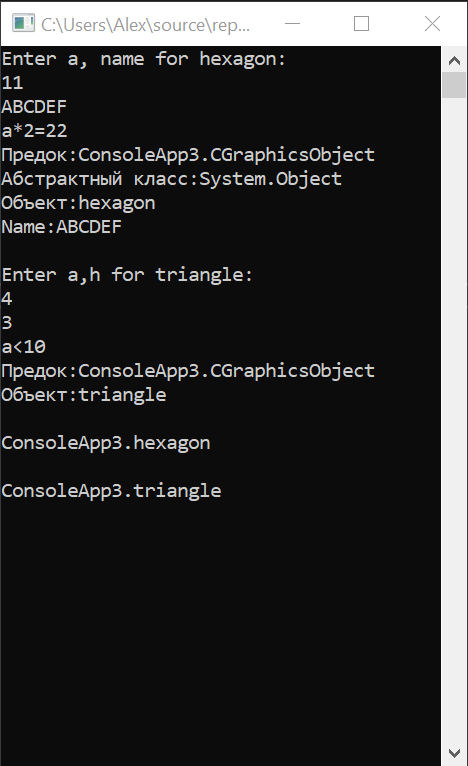
Console.ReadKey();

}

}

}

**Результат работы программы:**



**Вывод:** во время выполнения это лабораторной работы янаучился создавать интерфейсы, делегаты , коллекции ознакомился с понятием наследования интерфейсов , их модификаторами доступов, применением. Создал программу , которая содержит классовую модель, состоящую из абстрактного класса и двух его наследников , каждый наследник содержит по делегату и интерфейсу ,также программа содержит коллекцию из экземпляров классов потомков.