**Міністерство освіти і науки України**

**Національний технічний університет**

**«Дніпровська політехніка»**



**ЗВІТ**

**про виконання лабораторних робіт**

**з дисципліни**

**«Об'єктно Орієнтоване Програмування»**

**Лабораторна робота № 3**

Виконав:

студент гр. 121-19-2

Дробний О.Ю.

Прийняв:

Приходченко С.Д.

**Дніпро**

**2020**

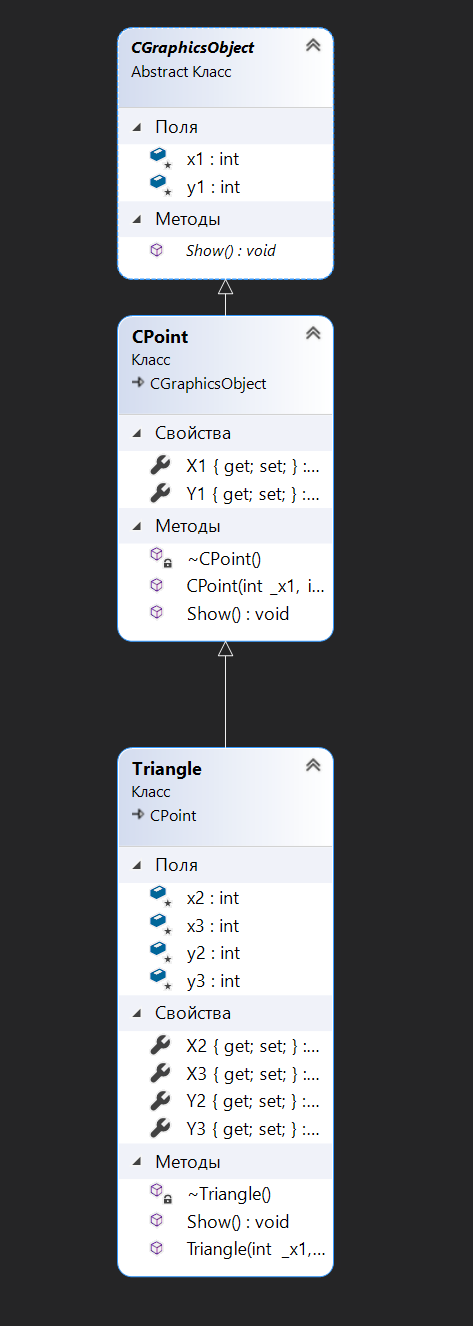
**Лабораторная работа № 3  
Вариант 6**

**Тема:** Изучение наследования как методики повторного использования кода в классах С#

**Цель:** научиться строить простейшие классовые модели;овладеть методикой автоматической генерации классов С# из диаграмм классов;создать определение функций-членов классов с использованием дополнительных средств отображения графики в консоли.

**Выполнение работы**

**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 ConsoleApp3

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter coordinates for x1,y1 for CPoint:" );

CPoint new\_point = new CPoint(Convert.ToInt32(Console.ReadLine()), Convert.ToInt32(Console.ReadLine()));

new\_point.Show();

Console.WriteLine("Enter coordinates for x1,y1,x2,y2,x3,y3 for Triangle:");

Triangle new\_triangle = new Triangle(Convert.ToInt32(Console.ReadLine()), Convert.ToInt32(Console.ReadLine()), Convert.ToInt32(Console.ReadLine()), Convert.ToInt32(Console.ReadLine()), Convert.ToInt32(Console.ReadLine()), Convert.ToInt32(Console.ReadLine()));

new\_triangle.Show();

Console.ReadKey();

}

}

}

**Файл CPoint.cs**

namespace ConsoleApp3

{

public class CPoint : CGraphicsObject

{

public int X1

{

get { return x1; }

set { x1 = value; }

}

public int Y1

{

get { return y1; }

set { y1 = value; }

}

public CPoint(int \_x1, int \_y1)

{

X1 = \_x1; Y1 = \_y1;

} //конструктор класса

public override void Show()

{

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

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

}

~CPoint()

{

Console.WriteLine("Destructor");

}// деструктор класса

}

}

**Файл CGraphicsObject.cs**

namespace ConsoleApp3

{

public abstract class CGraphicsObject

{

protected int x1; //поля класса

protected int y1;

public abstract void Show();

}

}

**Файл Triangle.cs**

namespace ConsoleApp3

{

public class Triangle : CPoint

{

protected int x2; //поля класса

protected int y2;

protected int x3; //поля класса

protected int y3;

public int X2 // setter&getter

{

get { return x2; }

set { x2 = value; }

}

public int Y2

{

get { return y2; }

set { y2 = value; }

}

public int X3

{

get { return x3; }

set { x3 = value; }

}

public int Y3

{

get { return y3; }

set { y3 = value; }

}

public Triangle(int \_x1, int \_y1, int \_x2, int \_y2, int \_x3, int \_y3) : base(\_x1, \_y1)

{

X1 = \_x1; Y1 = \_y1; X2 = \_x2; Y2 = \_y2; X3 = \_x3; Y3 = \_y3;

} //конструктор класса

public override void Show()

{

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

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

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

}

~Triangle()

{

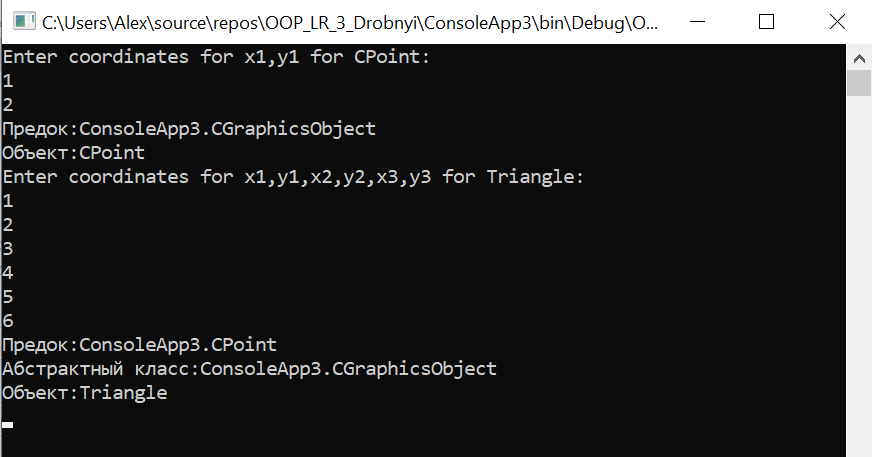
Console.WriteLine("Destructor");

}// деструктор класса

}

}

**Результат работы программы:**



**Вывод:**в результате выполнения этой лабораторной работы янаучился 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, его наследник - базовый класс CPoint, и класс-наследник от CPoint, представляющий собой графический объект.