УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

**“ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО”**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

по дисциплине «Объектно-ориентированное программирование»

На тему **«**Основы синтаксиса объектно-ориентированного языка программирования**»**

Выполнил:

студент гр. ИТП-21

Козлов О. В.

Принял: ассистент

Гуменников Е.Д.

Гомель 2022

**Цель работы:** Освоить основы синтаксиса объектно-ориентированного языка программирования на языке *c#*.

**Задание:**

1. Необходимо разработать класс, представляющий собой плоскую фигуру, согласно варианта (таблица 1).
2. При именовании класса, полей и методов руководствоваться соглашением о наименовании кода
3. Фигуру следует задавать координатами вершин.
4. Реализовать метод проверки возможности существования данной фигуры
5. Реализовать методы вычисление длин сторон, площади и периметра
6. Реализовать методы, проверяющие принадлежность точки, заданной своими координатами на плоскости, фигуре, её границе.
7. Класс должен быть размещён в библиотеке классов.
8. Весь код должен быть снабжён элементами документирования
9. Сгенерировать XML-файл с документацией по проекту
10. Создать консольное приложение с интерфейсом пользователя
11. Подключить созданную библиотеку в консольное приложение
12. Верифицировать разработанное приложение
13. Сгенерировать исполняемый файл для последующего переноса на другие компьютеры

Таблица 1

|  |  |
| --- | --- |
| **Вариант** | **Условие Задачи** |
| 13 | Криволинейная трапеция на заданном интервале, образуемая показательной функцией и осью OX |

**Ход работы**

В ходе работы был разработан класс *Trapezoid,* который представляет плоскую фигуру, заданную координатами четырех точек. В классе параллелограмма хранятсякоординаты x,y двух точек и параметр a для построения трапеции. Так же класс содержит методы для: выяснения возможности существования данного параллелограмма, вычисления: площади, длины, периметра – и проверки точки на принадлежность к фигуре. Код программы будет представлен в приложении А.

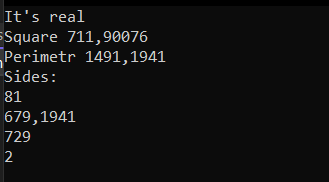


Рисунок 1 – Пример работы программы

**Вывод:** в ходе лабораторной работы были освоены основы синтаксиса объектно-ориентированного языка программирования на языке *c#*.

**Приложение А**

Листинг основной программы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Figure;

namespace lab1norm

{

internal class Program

{

static void Main(string[] args)

{

double[] x = new double[4], y = new double[4];

double pointX,pointY;

for(int i = 0; i < 4; i++)

{

Console.Write($"Введите x{i} и y{i} \n");

x[i] = Convert.ToDouble(Console.ReadLine());

y[i] = Convert.ToDouble(Console.ReadLine());

}

Console.Write("Введите координаты точки для проверки");

pointX = Convert.ToDouble(Console.ReadLine());

pointY = Convert.ToDouble(Console.ReadLine());

Figure.Parallel p1 = new Figure.Parallel(x,y);

double[] sides = p1.getSides();

if (p1.isReal())

{

Console.WriteLine("It's real");

Console.WriteLine("Square " + p1.getSquare());

Console.WriteLine("Perimetr " + p1.getPerimeter());

Console.WriteLine("Sides:");

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

{

Console.WriteLine(sides[i]);

}

Console.WriteLine(p1.doesInclude(0, 0));

}

else

{

Console.WriteLine("It's not real");

}

}

}

}

Листинг класса *Trapezoid*

internal class Trapezoid

{

public float x1, x2, y1 = 0, y2 = 0, a, leftSide, topSide, rightSide, bottomSide;

public Trapezoid(float x1, float x2,float a)

{

this.x1 = x1;

this.x2 = x2;

this.a = a;

}

public bool isReal()

{

return x1 != x2;

}

public float[] getSides()

{

leftSide = (float)(Math.Abs(Math.Pow(a,x1)));

topSide = (float)(Math.Abs(Math.Sqrt(1+Math.Pow((a\*a),x2)\*Math.Log(a))- (Math.Sqrt(1 + Math.Pow((a \* a), x1) \* Math.Log(a)))));

rightSide = (float)(Math.Abs(Math.Pow(a, x2)));

bottomSide = (float)(Math.Abs(x2-x1));

float[] sides = { leftSide, topSide, rightSide, bottomSide};

return sides;

}

public float getPerimeter()

{

return leftSide + topSide + rightSide + bottomSide;

}

public float getSquare()

{

return (float)(Math.Pow(a,x2)\*Math.Log(a)- Math.Pow(a, x1) \* Math.Log(a));

}

}Начало формы

Конец формы