МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

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 проектирование и программирование»

на тему: «Основы синтаксиса объектно-ориентированного языка программирования»

Выполнил студент гр. ИТИ-21:  
Буякевич Д.О.  
 Принял преподаватель:  
 Чернявская А.Э.

Гомель 2022

**Цель работы:** изучить основы синтаксиса языка c#, научиться объявлять поля, классы, методы. Научиться работать с переменными типа private, public и protected.

**Ход работы**

В таблице показан вариант лабораторной работы и фигура разработки класса, представляющего собой плоскую фигуру, согласно варианту 3.

Таблица 1.

|  |  |
| --- | --- |
| 3 | Трапеция |

На рисунке 1 представлены требования к классу которые нужно реализовать.

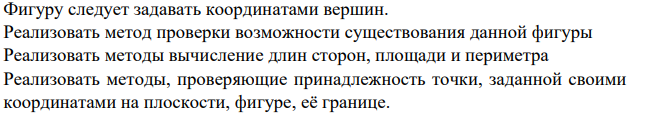


Рисунок 1 – Требования к классу

На рисунке 2 выведено интерфейс пользователя с выбором пунктов.

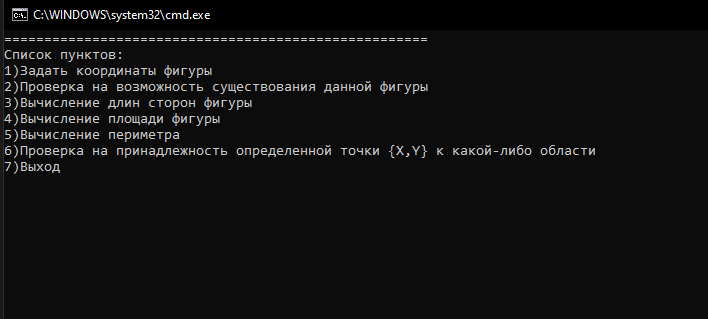


Рисунок 2 – Меню программы

Рисунок 3 показывает ввод координат вершин фигуры.

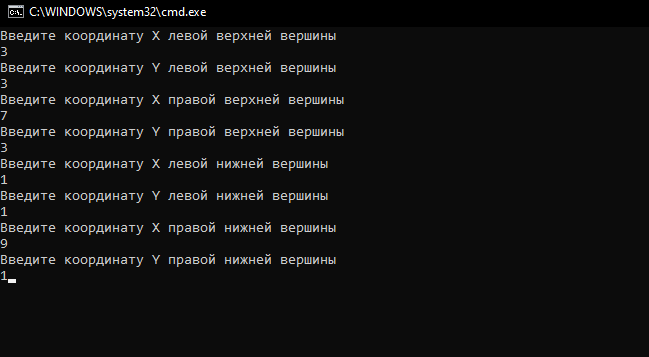


Рисунок 3 – Ввод координат фигуры

На рисунке 4 показан пункт существования фигуры. Существует несколько вариантов:

* Фигура существует
* Фигура не существует, т.к. имеет три вершины
* Фигура является прямоугольником, а не трапецией
* Фигура в принципе не трапеция, т.к. ее вершины образуют другую фигуру

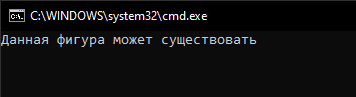


Рисунок 4 – Информация о существовании фигуры

На рисунке 5 выведены длины сторон трапеции.

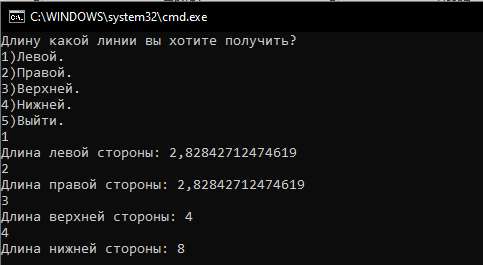


Рисунок 5 – Длины сторон фигуры

Рисунки 6 и 7 выводят информацию о площади и периметре трапеции.

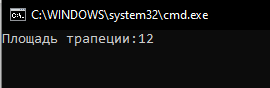


Рисунок 6 – Площадь трапеции

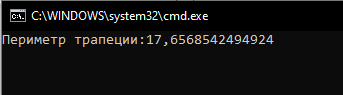


Рисунок 7 – Периметр трапеции

Рисунки 8, 9 и 10 показывают принадлежность (или ее отсутствие) введенной точки к фигуре. Точка может быть:

* Внутри фигуры (рис. 8)
* За пределами фигура (рис. 9)
* На границе фигуры (рис. 10)

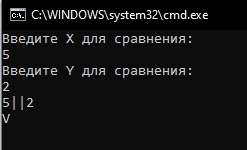


Рисунок 8 – Точка в фигуре

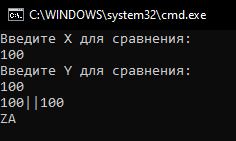


Рисунок 9 – Точка за фигурой

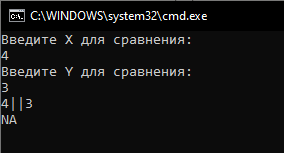


Рисунок 10 – Точка на фигуре

Вывод: познакомились с основами синтаксиса языка C#. Работали с полями, методами и классами для выполнения лабораторной работы. Использовали Coding-Convention для более чистого кода.

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

Файл Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_01

{

internal class Program

{

static void Main(string[] args)

{

Trapezoid trapezoid = new Trapezoid();

byte choiceMenuPunct = 0;

while (choiceMenuPunct != 7)

{

Console.WriteLine("=====================================================");

Console.WriteLine("Список пунктов:");

Console.WriteLine("1)Задать координаты фигуры");

Console.WriteLine("2)Проверка на возможность существования данной фигуры");

Console.WriteLine("3)Вычисление длин сторон фигуры");

Console.WriteLine("4)Вычисление площади фигуры");

Console.WriteLine("5)Вычисление периметра");

Console.WriteLine("6)Проверка на принадлежность определенной точки {X,Y} к какой-либо области");

Console.WriteLine("7)Выход");

choiceMenuPunct = Convert.ToByte(Console.ReadLine());

Console.Clear();

switch (choiceMenuPunct)

{

case 1:

Console.WriteLine("Введите координату X левой верхней вершины");

trapezoid.X\_lt = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите координату Y левой верхней вершины");

trapezoid.Y\_lt = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите координату X правой верхней вершины");

trapezoid.X\_rt = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите координату Y правой верхней вершины");

trapezoid.Y\_rt = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите координату X левой нижней вершины");

trapezoid.X\_lb = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите координату Y левой нижней вершины");

trapezoid.Y\_lb = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите координату X правой нижней вершины");

trapezoid.X\_rb = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите координату Y правой нижней вершины");

trapezoid.Y\_rb = Convert.ToInt32(Console.ReadLine());

double checkX;

double checkY;

Console.ReadKey();

Console.Clear();

break;

case 2:

trapezoid.isExists();

Console.ReadKey();

Console.Clear();

break;

case 3:

trapezoid.GetLine();

Console.ReadKey();

Console.Clear();

break;

case 4:

Console.WriteLine($"Площадь трапеции:{trapezoid.GetArea()}");

Console.ReadKey();

Console.Clear();

break;

case 5:

Console.WriteLine($"Периметр трапеции:{trapezoid.GetPerimetr()}");

Console.ReadKey();

Console.Clear();

break;

case 6:

Console.WriteLine("Введите X для сравнения:");

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

Console.WriteLine("Введите Y для сравнения:");

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

trapezoid.isBelongs(ref checkX, ref checkY);

Console.ReadKey();

Console.Clear();

break;

case 7:

//Выход

break;

default:

break;

}

}

}

}

}

Файл Trapezoid.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_01

{

class Trapezoid

{

// lt - left-top

// rt - rigth-top

// lb - left-bot

// rb - right-bot

public double X\_lt;

public double Y\_lt;

public double X\_rt;

public double Y\_rt;

public double X\_lb;

public double Y\_lb;

public double X\_rb;

public double Y\_rb;

public void isExists()

{

if (((X\_lt == X\_rt) && (Y\_lt == Y\_rt)) || ((X\_lt == X\_lb) && (Y\_lt == Y\_lb)) || ((X\_lt == X\_rb) && (Y\_lt == Y\_rb)) || ((X\_rt == X\_lb) && (Y\_rt == Y\_lb)) || ((X\_rt == X\_rb) && (Y\_rt == Y\_rb)) || ((X\_lb == X\_rb) && (Y\_lb == Y\_rb)))

{

Console.WriteLine("Данная фигура не может существовать, она имеет три вершины");

}

else if (((X\_lt == X\_lb) && (X\_rt == X\_rb)) /\*|| ((Y\_lt == Y\_rt) && (Y\_lb == Y\_rb))\*/)

{

Console.WriteLine("Данная фигура является прямоугольником, а не трапецией");

}

else if (((Y\_lt != Y\_rt) || (Y\_lb != Y\_rb)) && ((Y\_lt != Y\_lb) || (Y\_rt != Y\_rb)))

{

Console.WriteLine("Данная фигура не трапеция, так как никакие две её линии не параллельны");

}

else

{

Console.WriteLine("Данная фигура может существовать");

}

//прямоугольник не трапеция, x1 = x2 , но y1 != y2

}

public double GetArea()

{

// нахождение катета(нижнего катета)

double katet = (GetLineBot() - GetLineTop()) / 2;

// нахождение высоты(второго катета с помощью теоремы пифагора

double height = Math.Sqrt((GetLineLeft() \* GetLineLeft()) - (katet \* katet));

// площадь = (сторона\_нижняя\*сторона\_верхняя/2)\*высоту

double area = ((GetLineTop() + GetLineBot()) / 2) \* height;

return area;

}

public double GetPerimetr()

{

return GetLineLeft() + GetLineRight() + GetLineTop() + GetLineBot();

}

// Нахождение длины прямой с помощью формулы длины прямой

public double GetLineLeft()

{

double line\_left = Math.Sqrt(Math.Pow(X\_lt - X\_lb, 2) + Math.Pow(Y\_lt - Y\_lb, 2));

return line\_left;

}

public double GetLineRight()

{

double line\_right = Math.Sqrt(Math.Pow(X\_rt - X\_rb, 2) + Math.Pow(Y\_rt - Y\_rb, 2));

return line\_right;

}

public double GetLineTop()

{

double line\_top = Math.Sqrt(Math.Pow(X\_rt - X\_lt, 2) + Math.Pow(Y\_rt - Y\_lt, 2));

return line\_top;

}

public double GetLineBot()

{

double line\_bot = Math.Sqrt(Math.Pow(X\_rb - X\_lb, 2) + Math.Pow(Y\_rb - Y\_lb, 2));

return line\_bot;

}

public void GetLine()

{

Console.WriteLine("Длину какой линии вы хотите получить?");

Console.WriteLine("1)Левой.\n2)Правой.\n3)Верхней.\n4)Нижней.\n5)Выйти.");

byte choice = 0;

while (choice != 5)

{

choice = Convert.ToByte(Console.ReadLine());

switch (choice)

{

case 1:

Console.WriteLine($"Длина левой стороны: {GetLineLeft()}");

break;

case 2:

Console.WriteLine($"Длина правой стороны: {GetLineRight()}");

break;

case 3:

Console.WriteLine($"Длина верхней стороны: {GetLineTop()}");

break;

case 4:

Console.WriteLine($"Длина нижней стороны: {GetLineBot()}");

break;

case 5: break;

default:

Console.WriteLine("Пункт выбран неверно!");

break;

}

}

}

public void isBelongs(ref double checkX, ref double checkY)

{

Console.WriteLine(checkX + "||" + checkY);

if (X\_rt < checkX || X\_lt > checkX || Y\_lt < checkY || Y\_lb > checkY)

Console.WriteLine("ZA");

else if (((X\_rt > checkX && X\_lt < checkX) && (Y\_lt > checkY && Y\_lb < checkY)))

Console.WriteLine("V");

else if (X\_lt == checkX || X\_rt == checkX || Y\_lt == checkY || Y\_lb == checkY)

Console.WriteLine("NA");

}

}

}