**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П.О.СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

направление специальности 6-05-0611-01 Информационные системы

и технологии (в игровой индустрии)

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине: «Объектно-ориентированное проектирование и программирование»

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

Выполнил: студент гр. ИТИ-21

Богословский И.Н.

Принял: ассистент

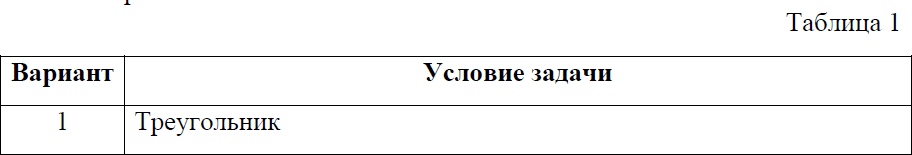
Панарин К.А.

Гомель 2024

**Цель работы**: изучить принципы объектно-ориентированного проектирования (ООП) и программирования на примере создания простых классов и взаимодействия между объектами. Реализовать основные концепции ООП, такие как инкапсуляция, наследование и полиморфизм, на практике.

# **Задание:**

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



**Ход работы и результаты выполнения:**

В ходе разработки программы была разработана библиотека классов – «*Figure*», содержащая два класса – «*Triangle*», согласно своему варианту.

Данная библиотека подключается к основному модулю программы, где реализована логика взаимодействия консольного приложения с пользователем, показанная на рисунке 1.

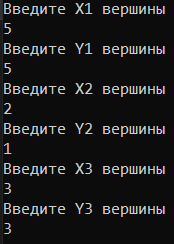


Рисунок 1 – Меню взаимодействия с пользователем

После ввода пользователем координат вершины треугольника, если он существует запускается основной модуль программы с функциями. Если треугольник не существует программа выводит сообщение о ошибке и просит ввести вершины заново. Представленный функционал представлен на рисунке 2.

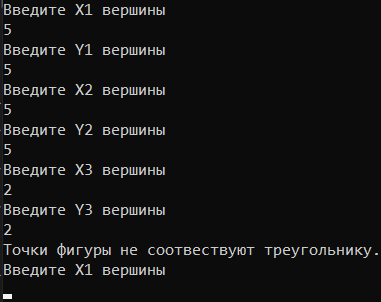


Рисунок 2 – Демонстрация работы программы

Меню с основным функционалом программы представлен на рисунке 3.

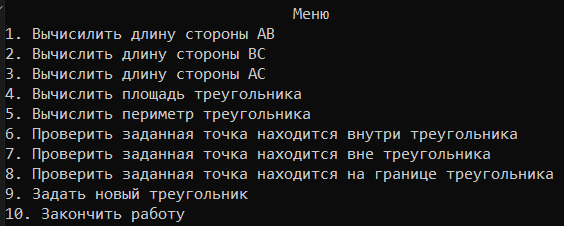


Рисунок 3 – Основной функционал программы

Так же был создан исполняемый файл, который можно импортировать на другие компьютеры.

Листинг разработанных классов и главного класса *Main* предоставлен в приложении А.

**Вывод:** в ходе выполнения лабораторной работы были освоены основные принципы объектно-ориентированного программирования, включая создание классов и объектов, инкапсуляцию данных, а также применение методов для вычисления характеристик объектов.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Листинг разработанных классов**

**«Triangle.cs»**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Security.Cryptography;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

namespace Figure

{

public class Triangle

{

int x1, x2, x3;

int y1, y2, y3;

public Triangle(int x1, int y1, int x2, int y2, int x3, int y3)

{

if (x1 \* (y2 - y3) + x2 \* (y3 - y1) + x3 \* (y1 - y2) == 0)

{

throw new ArgumentException("Треугольник из этих точек невозможно образовать.");

}

this.x1 = x1;

this.y1 = y1;

this.x2 = x2;

this.y2 = y2;

this.x3 = x3;

this.y3 = y3;

}

private double calculateDistance(int x1, int y1, int x2, int y2)

{

return Math.Sqrt(Math.Pow(x2 - x1, 2) + Math.Pow(y2 - y1, 2));

}

public double AB => calculateDistance(x1, y1, x2, y2);

public double AC => calculateDistance(x1, y1, x3, y3);

public double BC => calculateDistance(x2, y2, x3, y3);

public double getPerimeter()

{

return AB + AC + BC;

}

public double getS()

{

double s = (AB + AC + BC) / 2;

return Math.Sqrt(s \* (s - AB) \* (s - AC) \* (s - BC));

}

private double getD(int x, int y, int x1, int y1, int x2, int y2)

{

return (x - x1) \* (y2 - y1) - (y - y1) \* (x2 - x1);

}

public bool isPointInside(int x, int y)

{

double D1 = getD(x, y, x1, y1, x2, y2);

double D2 = getD(x, y, x2, y2, x3, y3);

double D3 = getD(x, y, x3, y3, x1, y1);

if (D1 > 0 && D2 > 0 && D3 > 0 || D1 < 0 && D2 < 0 && D3 < 0)

return true;

else

return false;

}

public bool isPountBoundary(int x, int y)

{

double D1 = getD(x, y, x1, y1, x2, y2);

double D2 = getD(x, y, x2, y2, x3, y3);

double D3 = getD(x, y, x3, y3, x1, y1);

if (D1 == 0 || D2 == 0 || D3 == 0)

return true;

else

return false;

}

public bool isPountOutside(int x, int y)

{

return !isPointInside(x, y);

}

}

}

**«Program.cs»**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Figure;

namespace lab

{

internal class Program

{

static void Main(string[] args)

{

startConsoleApp(getTriangle());

}

static Triangle getTriangle()

{

Console.WriteLine("Введите X1 вершины");

int x1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите Y1 вершины");

int y1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите X2 вершины");

int x2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите Y2 вершины");

int y2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите X3 вершины");

int x3 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите Y3 вершины");

int y3 = Convert.ToInt32(Console.ReadLine());

Triangle triangle;

try

{

triangle = new Triangle(x1, y1, x2, y2, x3, y3);

}

catch (ArgumentException ex)

{

Console.WriteLine("Точки фигуры не соотвествуют треугольнику.");

return getTriangle();

}

return triangle;

}

static void startConsoleApp(Triangle triangle)

{

bool isEnd = false;

while (!isEnd)

{

Console.WriteLine("Меню".PadRight(28, ' ').PadLeft(60, ' '));

Console.WriteLine("1. Вычисилить длину стороны AB");

Console.WriteLine("2. Вычислить длину стороны BC");

Console.WriteLine("3. Вычислить длину стороны AC");

Console.WriteLine("4. Вычислить площадь треугольника");

Console.WriteLine("5. Вычислить периметр треугольника");

Console.WriteLine("6. Проверить заданная точка находится внутри треугольника");

Console.WriteLine("7. Проверить заданная точка находится вне треугольника");

Console.WriteLine("8. Проверить заданная точка находится на границе треугольника");

Console.WriteLine("9. Задать новый треугольник");

Console.WriteLine("10. Закончить работу");

int answer = Convert.ToInt32(Console.ReadLine());

switch (answer)

{

case 1:

{

Console.WriteLine("Сторона AB = " + triangle.AB);

break;

}

case 2:

{

Console.WriteLine("Сторона ВС = " + triangle.BC);

break;

}

case 3:

{

Console.WriteLine("Сторона AC = " + triangle.AC);

break;

}

case 4:

{

Console.WriteLine("Площадь треугольника = " + triangle.getS());

break;

}

case 5:

{

Console.WriteLine($"Периметр треугольника = {triangle.getPerimeter()}");

break;

}

case 6:

{

Console.WriteLine("Введите точку x");

int x = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите точку y");

int y = Convert.ToInt32(Console.ReadLine());

if (triangle.isPointInside(x, y))

{

Console.WriteLine("Точка находится внутри треугольника");

}

else

{

Console.WriteLine("Точка находится вне треугольника или на его границе");

}

break;

}

case 7:

{

Console.WriteLine("Введите точку x");

int x = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите точку y");

int y = Convert.ToInt32(Console.ReadLine());

if (triangle.isPountOutside(x, y))

{

Console.WriteLine("Точка находится вне треугольника");

}

else

{

Console.WriteLine("Точка находится внутри треугольника или на его границе");

}

break;

}

case 8:

{

Console.WriteLine("Введите точку x");

int x = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите точку y");

int y = Convert.ToInt32(Console.ReadLine());

if (triangle.isPountBoundary(x, y))

{

Console.WriteLine("Точка находится на границе треугольника");

}

else

{

Console.WriteLine("Точка находится внутри треугольника или вне его");

}

break;

}

case 9:

{

isEnd = true;

startConsoleApp(getTriangle());

break;

}

case 10:

{

isEnd = true;

break;

}

}

}

}

}

}