**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П.О.СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине: «Объектно-ориентированное проектирование и программирование»

на тему: «Особенности разработки и

верификации в объектно-ориентированном языке»

Выполнил: студент гр. ИТИ-21

Беловский С.С.

Принял: -------

----- -.-.

Гомель 2025

**Цель работы:** Создание консольного приложения с интерфейсом пользователя.

**Задание:** Разработать класс, представляющий собой плоскую фигуру, согласно варианту (рисунок 1).

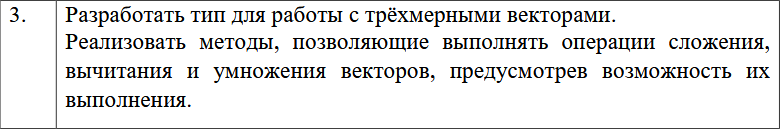


Рисунок 1 – Вариант задания

**Ход выполнения задания**

На рисунке 2 представлен результат выполнения программы с произвольными введенными значениями. Код программы приведен в приложении А.

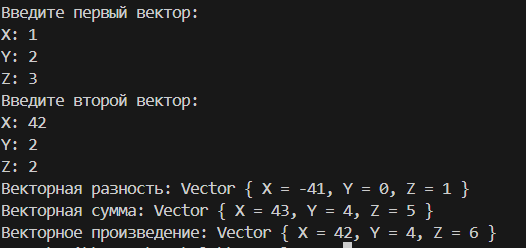


Рисунок 2 – Внешний вид меню и ввод трапеции

На рисунке 3 представлен результат выполнения тестов задания.

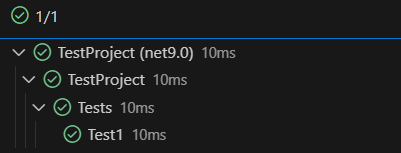


Рисунок 3 – Результат выполнения тестов

В приложении Б представлены используемые для проверки тесты.

**Вывод:** Были изучены методы создания ООП приложений на базе языка *C#*, изучены методы создания библиотек классов и их подключение к основному проекту.

**ПРИЛОЖЕНИЕ А**

(Обязательное)

**Текст программы**

**Program.cs:**

using ClassLib.Second;

Console.WriteLine("Введите первый вектор: ");

var firstVector = EnterVector();

Console.WriteLine("Введите второй вектор: ");

var secondVector = EnterVector();

Console.WriteLine($"Векторная разность: { firstVector - secondVector }");

Console.WriteLine($"Векторная сумма: { firstVector + secondVector }");

Console.WriteLine($"Векторное произведение: { firstVector \* secondVector }");

Vector EnterVector()

{

Console.Write("X: ");

double x = double.Parse(Console.ReadLine()!);

Console.Write("Y: ");

double y = double.Parse(Console.ReadLine()!);

Console.Write("Z: ");

double z = double.Parse(Console.ReadLine()!);

return new Vector(x, y, z);

}

**Vector.cs:**

namespace ClassLib.Second;

/// <summary>

/// 3 dimensioned vector

/// </summary>

/// <param name="X"></param>

/// <param name="Y"></param>

/// <param name="Z"></param>

public record struct Vector(double X, double Y, double Z)

{

/// <summary>

/// vector subtraction first minus second

/// </summary>

/// <param name="first"></param>

/// <param name="second"></param>

/// <returns></returns>

public static Vector operator -(Vector first, Vector second)

{

var (fx, fy, fz) = first;

var (sx, sy, sz) = second;

return new Vector(fx - sx, fy - sy, fz - sz);

}

/// <summary>

/// vector addition

/// </summary>

/// <param name="first"></param>

/// <param name="second"></param>

/// <returns></returns>

public static Vector operator +(Vector first, Vector second)

{

var (fx, fy, fz) = first;

var (sx, sy, sz) = second;

return new Vector(fx + sx, fy + sy, fz + sz);

}

/// <summary>

/// vector multiplication

/// </summary>

/// <param name="first"></param>

/// <param name="second"></param>

/// <returns></returns>

public static Vector operator \*(Vector first, Vector second)

{

var (fx, fy, fz) = first;

var (sx, sy, sz) = second;

return new Vector(fx \* sx, fy \* sy, fz \* sz);

}

}

**ПРИЛОЖЕНИЕ Б**

(Обязательное)

**Структура программы**

**VectorTests.cs**

using ClassLib.Second;

namespace TestProject;

public class VectorTests

{

[Test]

public void Subtraction()

{

var first = new Vector(1, 1, 1);

var second = new Vector(1, 1, 1);

Assert.That(first - second == new Vector(0, 0, 0));

}

[Test]

public void Addition()

{

var first = new Vector(1, 1, 1);

var second = new Vector(1, 1, 1);

Assert.That(first + second == new Vector(2, 2, 2));

}

[Test]

public void Multiplication()

{

var first = new Vector(1, 1, 1);

var second = new Vector(3, 3, 3);

Assert.That(first \* second == new Vector(3, 3, 3));

}

}