**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 № 4

по дисциплине: «Объектно-ориентированное проектирование и программирование»

на тему: «Обработка строковых данных.

Регулярные выражения

»

Выполнил студент

группы ИТП-21

Лосев М. А.

Проверил ассистент

Карась О. В.

Гомель 2025

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

**Задание:** 1. Разработать приложения, решающие поставленную задачу (таблица 4.1 и таблица 4.2) 2. Разработать не менее 15 модульных тестов для тестирования



Рисунок 1 – Вариант задания

Ход выполнения задания

1. В приложении А указан код программы*.*
2. На рисунках 2 показан результат запуска программы.
3. На рисунке 3 представлены пройденные тесты

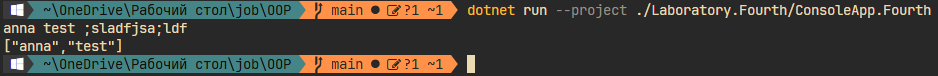


Рисунок 2 – Результат выполнения программы

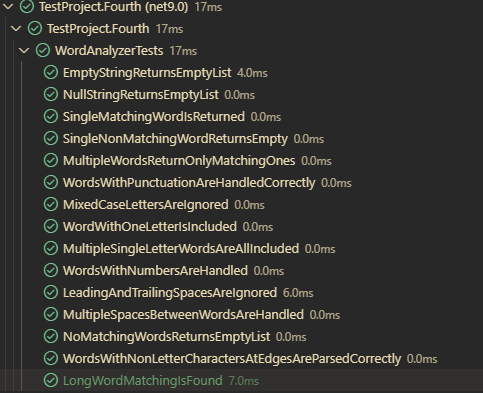


Рисунок 3 – Результат выполнения тестов

В приложении А представлен код программы. В приложении Б представлены полученные *xml* файлы.

**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 System.Text.Json;

using ClassLib.Fourth;

var result = WordAnalyzer.FindWordsWithSameStartAndEnd(Console.ReadLine() ?? "");

Console.WriteLine(JsonSerializer.Serialize(result));

***WordAnalyzer.cs*:**

using System.Text.RegularExpressions;

namespace ClassLib.Fourth;

public static class WordAnalyzer

{

/// <summary>

/// Finds all the words that start and end with the same character (case-insensitive).

/// </summary>

/// <param name="input">The input string.</param>

/// <returns>A list of words that start and end with the same character.</returns>

public static List<string> FindWordsWithSameStartAndEnd(string input)

{

if (string.IsNullOrWhiteSpace(input))

return [];

var words = Regex.Matches(input, @"\b\w+\b")

.Cast<Match>()

.Select(m => m.Value)

.ToList();

var result = new List<string>();

foreach (var word in words)

{

if (word.Length == 0)

continue;

char first = char.ToLower(word[0]);

char last = char.ToLower(word[^1]);

if (first == last)

{

result.Add(word);

}

}

return result;

}

}