

Московский государственный технический университет
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”



Дисциплина «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №5

ГМ

Выполнил:
студент группы ИУ5-36Б
Германенко И. М.
Преподаватель:
Гапанюк Ю.Е.

Москва, 2025

Описание задания:

- Выбрать тип данных (например, Option/Maybe);
- Реализовать для него:
 - **функтор** (map);
 - **аппликативный функтор** (apply);
 - **монаду** (bind);
- Проверить выполнение соответствующих **законов** (там, где возможно в F#).

Листинг кода:

Program.cs:

```
using System;
using LevenshteinLibrary; // Подключаем нашу библиотеку

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("--- Лабораторная №5 (Адаптация для WSL) ---");

        Console.Write("Введите строку 1: ");
        string s1 = Console.ReadLine() ?? "";

        Console.Write("Введите строку 2: ");
        string s2 = Console.ReadLine() ?? "";

        Console.WriteLine("\nРезультаты:");

        int distLev = StringMetrics.LevenshteinDistance(s1, s2);
        Console.WriteLine($"Расстояние Левенштейна: {distLev}");

        int distDam = StringMetrics.DamerauLevenshteinDistance(s1, s2);
        Console.WriteLine($"Расстояние Дамерау-Левенштейна: {distDam}");
    }
}
```

StringMetrics.cs:

```
using System;

namespace LevenshteinLibrary
{
    public static class StringMetrics
    {
        // Обычное расстояние Левенштейна
        public static int LevenshteinDistance(string source, string target)
        {
            if (string.IsNullOrEmpty(source)) return string.IsNullOrEmpty(target)
                ? 0 : target.Length;
        }
    }
}
```

```

        if (string.IsNullOrEmpty(target)) return source.Length;

        int n = source.Length;
        int m = target.Length;
        int[,] matrix = new int[n + 1, m + 1];

        for (int i = 0; i <= n; i++) matrix[i, 0] = i;
        for (int j = 0; j <= m; j++) matrix[0, j] = j;

        for (int i = 1; i <= n; i++)
        {
            for (int j = 1; j <= m; j++)
            {
                int cost = (target[j - 1] == source[i - 1]) ? 0 : 1;
                matrix[i, j] = Math.Min(
                    Math.Min(matrix[i - 1, j] + 1, matrix[i, j - 1] + 1),
                    matrix[i - 1, j - 1] + cost);
            }
        }
        return matrix[n, m];
    }

    // Расстояние Дамерау-Левенштейна (с перестановками)
    public static int DamerauLevenshteinDistance(string source, string
target)
    {
        if (string.IsNullOrEmpty(source)) return string.IsNullOrEmpty(target)
? 0 : target.Length;
        if (string.IsNullOrEmpty(target)) return source.Length;

        int n = source.Length;
        int m = target.Length;
        int[,] matrix = new int[n + 1, m + 1];

        for (int i = 0; i <= n; i++) matrix[i, 0] = i;
        for (int j = 0; j <= m; j++) matrix[0, j] = j;

        for (int i = 1; i <= n; i++)
        {
            for (int j = 1; j <= m; j++)
            {
                int cost = (target[j - 1] == source[i - 1]) ? 0 : 1;
                matrix[i, j] = Math.Min(
                    Math.Min(matrix[i - 1, j] + 1, matrix[i, j - 1] + 1),
                    matrix[i - 1, j - 1] + cost);

                if (i > 1 && j > 1 && source[i - 1] == target[j - 2] &&
source[i - 2] == target[j - 1])
                {
                    matrix[i, j] = Math.Min(matrix[i, j], matrix[i - 2, j -
2] + cost);

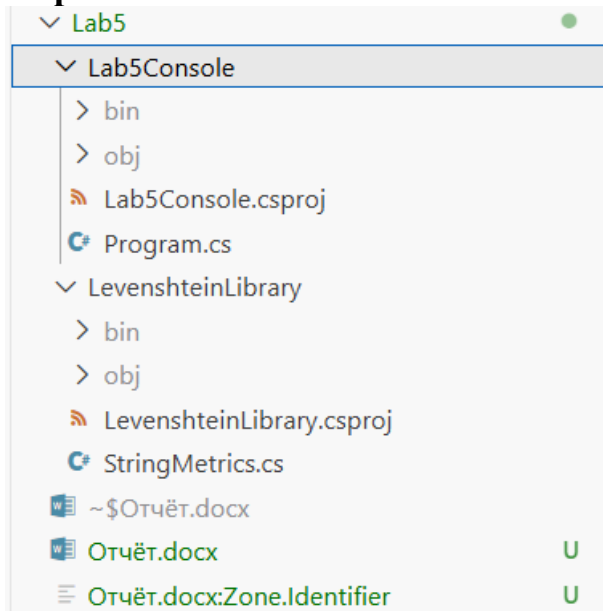
```

```

    }
    }
    }
    return matrix[n, m];
}
}
}

```

Скриншоты:



```

● root@ILGERPC:/home/PCPL/Lab5# cd ./Lab5Console/
● root@ILGERPC:/home/PCPL/Lab5/Lab5Console# dotnet run
--- Лабораторная №5 (Адаптация для WSL) ---
Введите строку 1: road
Введите строку 2: read

Результаты:
Расстояние Левенштейна: 1
Расстояние Дамерау-Левенштейна: 1
○ root@ILGERPC:/home/PCPL/Lab5/Lab5Console#

```