Московский государственный технический университет им. Н.Э. Баумана.

Факультет «Информатика и управление»

Кафедра ИУ5.

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №5

Выполнил:
студент группы ИУ5-31Б

Гришин Станислав
Подпись и дата:
21.12.2020

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:
21.12.2020

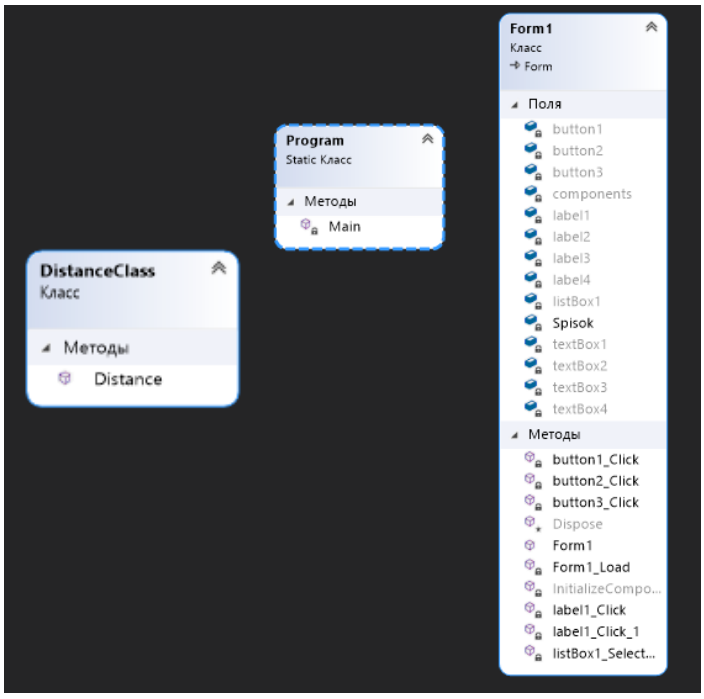
г. Москва, 2020 г.

Постановка задачи

Разработать программу, реализующую вычисление расстояния Левенштейна с использованием алгоритма Вагнера-Фишера.

1. Программа должна быть разработана в виде библиотеки классов на языке C#.
2. Использовать самый простой вариант алгоритма без оптимизации.
3. Дополнительно возможно реализовать вычисление расстояния Дамерау-Левенштейна (с учетом перестановок соседних символов).
4. Модифицировать предыдущую лабораторную работу, вместо поиска подстроки используется вычисление расстояния Левенштейна.
5. Предусмотреть отдельное поле ввода для максимального расстояния. Если расстояние Левенштейна между двумя строками больше максимального, то строки считаются несовпадающими и не выводятся в список результатов.

Разработка интерфейса класса



Листинг программы

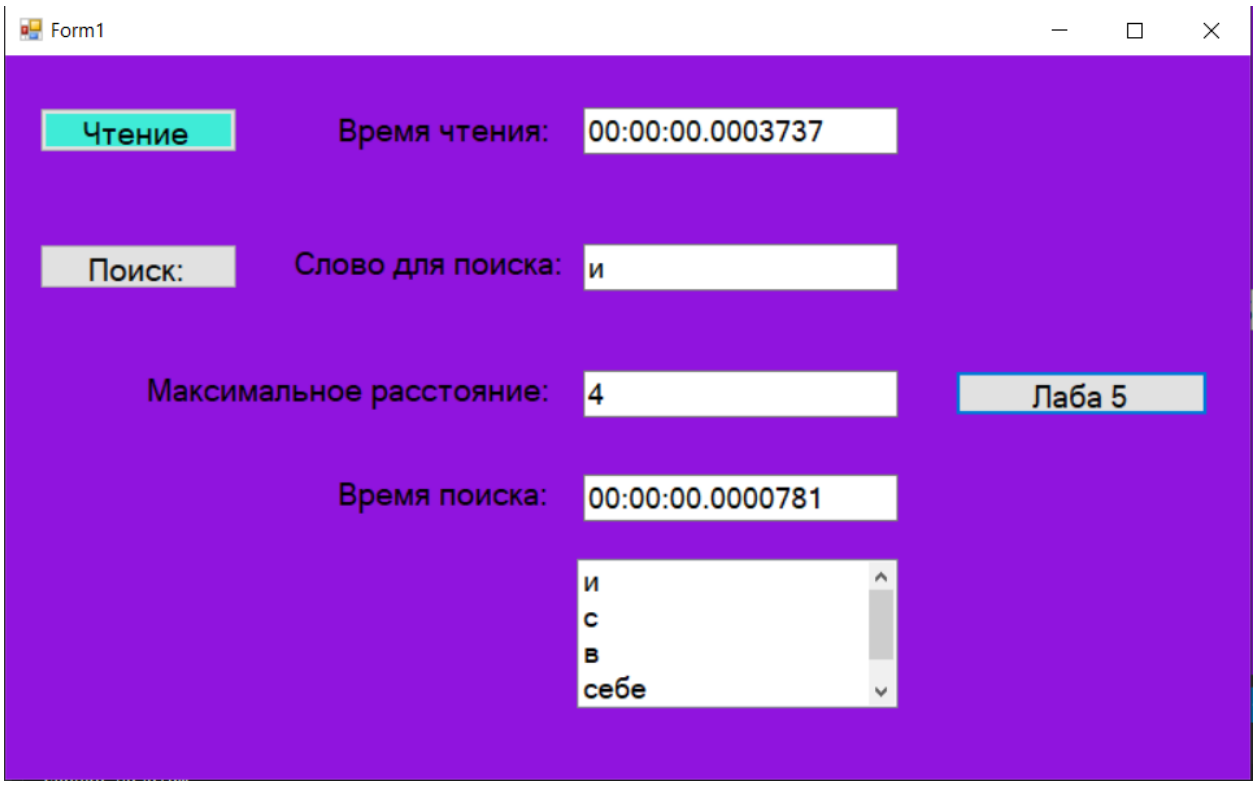
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
```

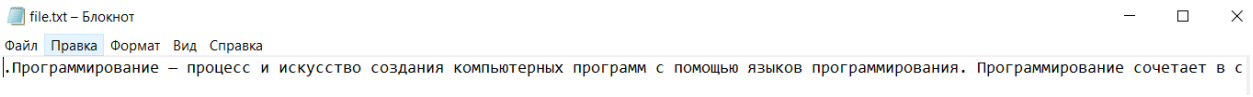
```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Diagnostics;
using LibraryLab5;

namespace Laba4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

        }
        List<string> Spisok = new List<string>();
        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {
            OpenFileDialog A = new OpenFileDialog();
            Stopwatch watch = new Stopwatch();
            A.Filter = "Текстовые файлы|*.txt";
            if (A.ShowDialog() != DialogResult.OK)
            {
                MessageBox.Show("Необходимо выбрать файл");
                return;
            }
            watch.Start();
            string text = File.ReadAllText(A.FileName);
            char[] razdeliteli = { '—', '-', '?', '!', ' ', ',', '.', ':', '\t', '\n' };
            //text=text.Trim(razdeliteli);
            string[] words = text.Split(razdeliteli);
            foreach (string word in words)
            {
                if (!String.IsNullOrEmpty(word))
                    if (!Spisok.Contains(word))
                        Spisok.Add(word);
            }
            watch.Stop();
            this.textBox1.Text = watch.Elapsed.ToString();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            if (!String.IsNullOrEmpty(this.textBox2.Text) && Spisok.Count > 0)
            {
                this.listBox1.Items.Clear();
                Stopwatch watch = new Stopwatch();
                string Word = this.textBox2.Text;
                this.listBox1.BeginUpdate();
                bool a = false;
                watch.Start();
                foreach (string word in Spisok)
                {
                    if (word.Contains(Word))
                    {
                        this.listBox1.Items.Add(word);
                        a = true;
                    }
                }
                watch.Stop();
```

```
                if (a == false)
                    this.listBox1.Items.Add("Не найдено совпадений");
                this.listBox1.EndUpdate();
                this.textBox3.Text = watch.Elapsed.ToString();
            }
            else if (String.IsNullOrEmpty(this.textBox2.Text) && Spisok.Count == 0)
            {
                MessageBox.Show("Необходимо отрыть файл и выбрать слово для поиска");
            }
            else if (String.IsNullOrEmpty(this.textBox2.Text))
            {
                MessageBox.Show("Необходимо выбрать слово для поиска");
            }
            else if (Spisok.Count == 0)
            {
                MessageBox.Show("Необходимо отрыть файл");
            }
        }
        private void button3_Click(object sender, EventArgs e)
        {
            if (!String.IsNullOrEmpty(this.textBox4.Text))
            {
                if (!String.IsNullOrEmpty(this.textBox2.Text) && Spisok.Count > 0)
                {
                    this.listBox1.Items.Clear();
                    Stopwatch watch = new Stopwatch();
                    string Word = this.textBox2.Text;
                    string Max = this.textBox4.Text;
                    int max = int.Parse(Max);
                    this.listBox1.BeginUpdate();
                    watch.Start();
                    foreach (string word in Spisok)
                    {
                        if (DistanceClass.Distance(word, Word) <= max)
                        {
                            this.listBox1.Items.Add(word);
                        }
                    }
                    watch.Stop();
                    this.listBox1.EndUpdate();
                    this.textBox3.Text = watch.Elapsed.ToString();
                }
                else if (String.IsNullOrEmpty(this.textBox2.Text) && Spisok.Count == 0)
                {
                    MessageBox.Show("Необходимо отрыть файл и выбрать слово для поиска");
                }
                else if (String.IsNullOrEmpty(this.textBox2.Text))
                {
                    MessageBox.Show("Необходимо выбрать слово для поиска");
                }
                else if (Spisok.Count == 0)
                {
                    MessageBox.Show("Необходимо отрыть файл");
                }

            }
            else
            {
                MessageBox.Show("Необходимо ввести максимальное расстояние");
            }
        }
    }
}

using System;
```

```
namespace LibraryLab5
{

    public class DistanceClass
    {
        public static int Distance(string str1Param, string str2Param)
        {
            if ((str1Param == null) || (str2Param == null)) return -1;
            int str1Len = str1Param.Length;
            int str2Len = str2Param.Length;
            if (str1Len == 0) return str2Len;
            if (str2Len == 0) return str1Len;

            string str1 = str1Param.ToUpper();
            string str2 = str2Param.ToUpper();

            int[,] matrix = new int[str1Len + 1, str2Len + 1];
            //Инициализация нулевой строки и нулевого столбца матрицы
            for (int i = 0; i <= str1Len; i++) matrix[i, 0] = i;
            for (int j = 0; j <= str2Len; j++) matrix[0, j] = j;
            for (int i = 1; i <= str1Len; i++)
            {
                for (int j = 1; j <= str2Len; j++)
                {
                    int symbEqual = (
                    (str1.Substring(i - 1, 1) ==
                    str2.Substring(j - 1, 1)) ? 0 : 1);
                    int ins = matrix[i, j - 1] + 1; //Добавление
                    int del = matrix[i - 1, j] + 1; //Удаление
                    int subst = matrix[i - 1, j - 1] + symbEqual; //Замена
                    matrix[i, j] = Math.Min(Math.Min(ins, del), subst);
                    if ((i > 1) && (j > 1) &&
                    (str1.Substring(i - 1, 1) == str2.Substring(j - 2, 1)) &&
                    (str1.Substring(i - 2, 1) == str2.Substring(j - 1, 1)))
                    {
                        matrix[i, j] = Math.Min(matrix[i, j],
                        matrix[i - 2, j - 2] + symbEqual);
                    }
                }
            }
            return matrix[str1Len, str2Len];
        }
    }
}
```

Анализ результатов


Form1
Чтение
Время чтения:
00:00:00.0003737
Поиск:
Слово для поиска:
и
Максимальное расстояние:
4
Лаба 5
Время поиска:
00:00:00.0000781
и
с
в
себе


file.txt – Блокнот
Файл Правка Формат Вид Справка
.Программирование — процесс и искусство создания компьютерных программ с помощью языков программирования. Программирование сочетает в с