|  |  |  |
| --- | --- | --- |
| Защищено:  Гапанюк Ю.Е.    "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |  | Демонстрация:  Гапанюк Ю.Е.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

**Отчет по лабораторной работе № 5 по курсу**

**Базовые компоненты интернет-технологий**

**ГУИМЦ**

#### Тема работы: " Библиотеки "

11

(количество листов)

Вариант № **4**

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы ИУ5Ц-52Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| Чиварзин А.Е. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

Москва, МГТУ - 2020

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

[1. Описание задания 3](#_Toc50574201)

[2. Диаграмма классов 3](#_Toc50574202)

[3. Текст программы 3](#_Toc50574203)

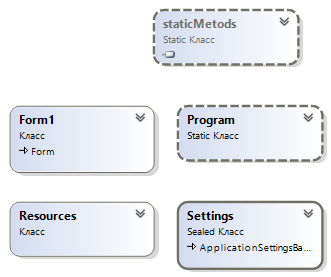
[4. Результаты выполнения программы 8](#_Toc50574204)

# Описание задания

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.

# Диаграмма классов



# Текст программы

**Form1.cs**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.IO;

using System.Windows.Forms;

using LAB\_5; //Библиотека

namespace LAB\_4

{

public partial class Form1 : Form

{

List<String> sList = new List<string>();

public Form1()

{

InitializeComponent();

}

private void openFileButton\_Click(object sender, EventArgs e)

{

OpenFileDialog fd = new OpenFileDialog();

fd.Filter = "текстовые файлы|\*.txt";

if (fd.ShowDialog() == DialogResult.OK)

{

Stopwatch sWatch = new Stopwatch();

sWatch.Start();

string fileContent = File.ReadAllText(fd.FileName);

//Разделительные символы для чтения из файла

char[] separators = new char[] { ' ', '.', ',', '!', '?', '/', '\t', '\n' };

string[] mas = fileContent.Split(separators);

foreach (string s in mas)

if (!sList.Contains(s))

sList.Add(s);

sWatch.Stop();

box\_readTime.Text = sWatch.Elapsed.ToString();

box\_wordsCount.Text = sList.Count.ToString();

}

else

{

MessageBox.Show("Необходито выбрать файл!");

}

}

private void SearchButton\_Click(object sender, EventArgs e)

{

string word = box\_search.Text.ToString();

//Если слово для поиска не пусто

if (!string.IsNullOrWhiteSpace(word) && sList.Count > 0)

{

string wordUpper = word.ToUpper(); //Перевести все буквы в верхний регистр

List<string> searchResult = new List<string>();

Stopwatch t = new Stopwatch();

t.Start();

foreach (string str in sList)

try

{

if (staticMetods.Distance(str.ToUpper(), wordUpper) <= Convert.ToInt32(box\_searchDistance.Text))

searchResult.Add(str);

} catch (FormatException)

{

MessageBox.Show("Максимальное расстояние Левинштейна введено неверно!");

break;

}

box\_resultSearch.BeginUpdate();

box\_resultSearch.Items.Clear();

foreach (string str in searchResult)

box\_resultSearch.Items.Add(str);

box\_resultSearch.EndUpdate();

t.Stop();

box\_searchTime.Text = t.Elapsed.ToString();

}

else

{

MessageBox.Show("Необходимо выбрать файл и ввести слово для поиска!");

}

}

private void exitButton\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

**Form1.Designer.cs**

namespace LAB\_4

{

partial class Form1

{

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

/// <param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе ложно.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Код, автоматически созданный конструктором форм Windows

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

private void InitializeComponent()

{

this.openFileButton = new System.Windows.Forms.Button();

this.label1 = new System.Windows.Forms.Label();

this.box\_readTime = new System.Windows.Forms.TextBox();

this.label2 = new System.Windows.Forms.Label();

this.box\_wordsCount = new System.Windows.Forms.TextBox();

this.SearchButton = new System.Windows.Forms.Button();

this.box\_search = new System.Windows.Forms.TextBox();

this.box\_resultSearch = new System.Windows.Forms.ListBox();

this.label3 = new System.Windows.Forms.Label();

this.box\_searchTime = new System.Windows.Forms.TextBox();

this.exitButton = new System.Windows.Forms.Button();

this.box\_searchDistance = new System.Windows.Forms.TextBox();

this.label4 = new System.Windows.Forms.Label();

this.SuspendLayout();

//

// openFileButton

//

this.openFileButton.Location = new System.Drawing.Point(12, 12);

this.openFileButton.Name = "openFileButton";

this.openFileButton.Size = new System.Drawing.Size(109, 23);

this.openFileButton.TabIndex = 0;

this.openFileButton.Text = "Открыть файл";

this.openFileButton.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

this.openFileButton.UseVisualStyleBackColor = true;

this.openFileButton.Click += new System.EventHandler(this.openFileButton\_Click);

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(149, 12);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(127, 13);

this.label1.TabIndex = 1;

this.label1.Text = "Время чтения из файла";

//

// box\_readTime

//

this.box\_readTime.Location = new System.Drawing.Point(282, 12);

this.box\_readTime.Name = "box\_readTime";

this.box\_readTime.ReadOnly = true;

this.box\_readTime.Size = new System.Drawing.Size(100, 20);

this.box\_readTime.TabIndex = 2;

//

// label2

//

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(12, 55);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(141, 13);

this.label2.TabIndex = 3;

this.label2.Text = "Уникальных слов в файле";

//

// box\_wordsCount

//

this.box\_wordsCount.Enabled = false;

this.box\_wordsCount.Location = new System.Drawing.Point(159, 55);

this.box\_wordsCount.Name = "box\_wordsCount";

this.box\_wordsCount.ReadOnly = true;

this.box\_wordsCount.Size = new System.Drawing.Size(100, 20);

this.box\_wordsCount.TabIndex = 4;

//

// SearchButton

//

this.SearchButton.Location = new System.Drawing.Point(292, 101);

this.SearchButton.Name = "SearchButton";

this.SearchButton.Size = new System.Drawing.Size(89, 23);

this.SearchButton.TabIndex = 5;

this.SearchButton.Text = "Поиск";

this.SearchButton.UseVisualStyleBackColor = true;

this.SearchButton.Click += new System.EventHandler(this.SearchButton\_Click);

//

// box\_search

//

this.box\_search.Location = new System.Drawing.Point(8, 104);

this.box\_search.Name = "box\_search";

this.box\_search.Size = new System.Drawing.Size(278, 20);

this.box\_search.TabIndex = 6;

//

// box\_resultSearch

//

this.box\_resultSearch.FormattingEnabled = true;

this.box\_resultSearch.Items.AddRange(new object[] {

"Здесь будут отображаться результаты поиска."});

this.box\_resultSearch.Location = new System.Drawing.Point(8, 130);

this.box\_resultSearch.Name = "box\_resultSearch";

this.box\_resultSearch.Size = new System.Drawing.Size(370, 212);

this.box\_resultSearch.TabIndex = 7;

//

// label3

//

this.label3.AutoSize = true;

this.label3.Location = new System.Drawing.Point(8, 345);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(79, 13);

this.label3.TabIndex = 8;

this.label3.Text = "Время поиска";

//

// box\_searchTime

//

this.box\_searchTime.Enabled = false;

this.box\_searchTime.Location = new System.Drawing.Point(93, 342);

this.box\_searchTime.Name = "box\_searchTime";

this.box\_searchTime.ReadOnly = true;

this.box\_searchTime.Size = new System.Drawing.Size(100, 20);

this.box\_searchTime.TabIndex = 9;

//

// exitButton

//

this.exitButton.Location = new System.Drawing.Point(8, 361);

this.exitButton.Name = "exitButton";

this.exitButton.Size = new System.Drawing.Size(75, 23);

this.exitButton.TabIndex = 10;

this.exitButton.Text = "Выход";

this.exitButton.UseVisualStyleBackColor = true;

this.exitButton.Click += new System.EventHandler(this.exitButton\_Click);

//

// box\_searchDistance

//

this.box\_searchDistance.Location = new System.Drawing.Point(224, 77);

this.box\_searchDistance.Name = "box\_searchDistance";

this.box\_searchDistance.Size = new System.Drawing.Size(100, 20);

this.box\_searchDistance.TabIndex = 11;

//

// label4

//

this.label4.AutoSize = true;

this.label4.Location = new System.Drawing.Point(12, 84);

this.label4.Name = "label4";

this.label4.Size = new System.Drawing.Size(206, 13);

this.label4.TabIndex = 12;

this.label4.Text = "Максимальное расстояние для поиска";

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(390, 389);

this.Controls.Add(this.label4);

this.Controls.Add(this.box\_searchDistance);

this.Controls.Add(this.exitButton);

this.Controls.Add(this.box\_searchTime);

this.Controls.Add(this.label3);

this.Controls.Add(this.box\_resultSearch);

this.Controls.Add(this.box\_search);

this.Controls.Add(this.SearchButton);

this.Controls.Add(this.box\_wordsCount);

this.Controls.Add(this.label2);

this.Controls.Add(this.box\_readTime);

this.Controls.Add(this.label1);

this.Controls.Add(this.openFileButton);

this.Name = "Form1";

this.Text = " Лабораторная работа №5";

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Button openFileButton;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.TextBox box\_readTime;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.TextBox box\_wordsCount;

private System.Windows.Forms.Button SearchButton;

private System.Windows.Forms.TextBox box\_search;

private System.Windows.Forms.ListBox box\_resultSearch;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.TextBox box\_searchTime;

private System.Windows.Forms.Button exitButton;

private System.Windows.Forms.TextBox box\_searchDistance;

private System.Windows.Forms.Label label4;

}

}

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace LAB\_4

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**MainPart.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB\_5

{

public static class staticMetods

{

/// <summary>

/// Вычисление расстояния Дамерау-Левенштейна

/// </summary>

public static int Distance(string str1Param, string str2Param)

{

if ((str1Param == null) || (str2Param == null))

return -1;

int str1Len = str1Param.Length;

int str2Len = str2Param.Length;

//Если хотя бы одна строка пустая, возвращается длина другой строки

if ((str1Len == 0) && (str2Len == 0))

return 0;

if (str1Len == 0)

return str2Len;

if (str2Len == 0)

return str1Len;

//Приведение строк к верхнему регистру

string str1 = str1Param.ToUpper();

string str2 = str2Param.ToUpper();

//Объявление матрицы

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;

//Вычисление расстояния Дамерау-Левенштейна

for (int i = 1; i <= str1Len; i++)

{

for (int j = 1; j <= str2Len; j++)

{

//Эквивалентность символов, переменная symbEqual //соответствует m(s1[i],s2[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

//Элемент матрицы вычисляется //как минимальный из трех случаев

matrix[i, j] = Math.Min(Math.Min(ins, del), subst);

//Дополнение Дамерау по перестановке соседних символов

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);

}

}

}

//Возвращается нижний правый элемент матрицы

return matrix[str1Len, str2Len];

}

/// <summary>

/// Вывод расстояния Дамерау-Левенштейна в консоль

/// </summary>

public static void WriteDistance(string str1Param, string str2Param)

{

int d = Distance(str1Param, str2Param);

Console.WriteLine("'" + str1Param + "','" + str2Param + "' -> " + d.ToString());

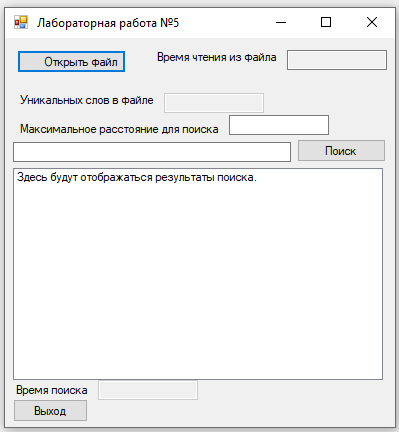
}

}

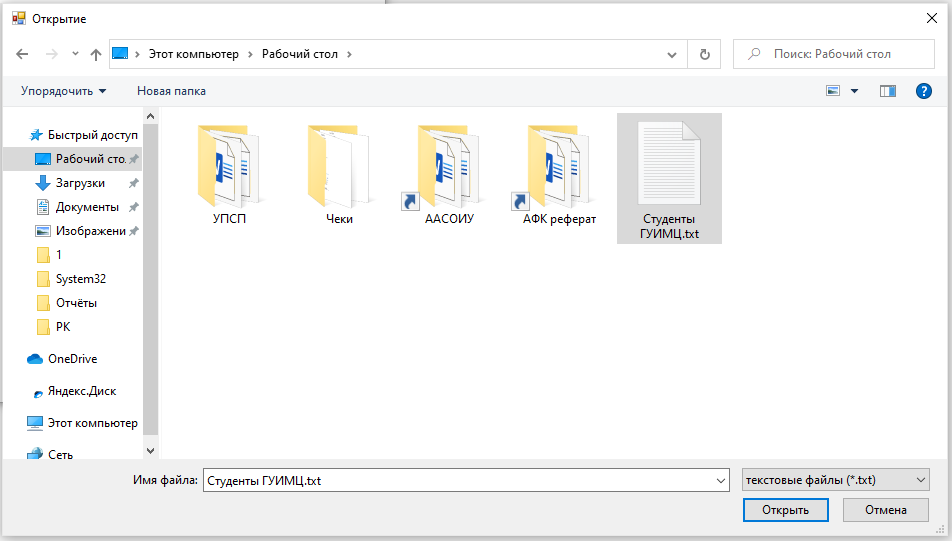
}

# Результаты выполнения программы

Окно приложения после запуска



Диалог открытия файла



Результаты поиска

