**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»

Отчет по лабораторной работе №4

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б |  | преподаватель каф. ИУ5 |
| Грызин Алексей |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

# Описание задания

Разработать программу, реализующую работу с файлами.

1. Программа должна быть разработана в виде приложения Windows Forms на языке C#. По желанию вместо Windows Forms возможно использование WPF.
2. Добавить кнопку, реализующую функцию чтения файла в список слов List<string>.
3. Для выбора имени файла используется класс OpenFileDialog, который открывает диалоговое окно с выбором файла. Ограничить выбор только файлами с расширением «.txt».
4. Для чтения из файла рекомендуется использовать статический метод ReadAllText() класса File (пространство имен System.IO). Содержимое файла считывается методом ReadAllText() в виде одной строки, далее делится на слова с использованием метода Split() класса string. Слова сохраняются в список List<string>.
5. При сохранении слов в список List<string> дубликаты слов не записываются. Для проверки наличия слова в списке используется метод Contains().
6. Вычислить время загрузки и сохранения в список с использованием класса Stopwatch (пространство имен System.Diagnostics). Вычисленное время вывести на форму в поле ввода (TextBox) или надпись (Label).
7. Добавить на форму поле ввода для поиска слова и кнопку поиска. При нажатии на кнопку поиска осуществлять поиск введенного слова в списке. Слово считается найденным, если оно входит в элемент списка как подстрока (метод Contains() класса string).
8. Добавить на форму список (ListBox). Найденные слова выводить в список с использованием метода «название\_списка.Items.Add()». Вызовы метода «название\_списка.Items.Add()» должны находится между вызовами методов «название\_списка.BeginUpdate()» и «название\_списка. EndUpdate()».
9. Вычислить время поиска с использованием класса Stopwatch. Вычисленное время вывести на форму в поле ввода (TextBox) или надпись (Label).

# Диаграмма классов

# 

# Текст программы

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.LoadTextFile = new System.Windows.Forms.Button();

this.ReadTime = new System.Windows.Forms.TextBox();

this.WordsCount = new System.Windows.Forms.TextBox();

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

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

this.InputWord = new System.Windows.Forms.TextBox();

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

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

this.FindTime = new System.Windows.Forms.TextBox();

this.ListFindWords = new System.Windows.Forms.ListBox();

this.label5 = new System.Windows.Forms.Label();

this.MaxDistance = new System.Windows.Forms.TextBox();

this.CheckWord = new System.Windows.Forms.TextBox();

this.button1 = new System.Windows.Forms.Button();

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

this.label6 = new System.Windows.Forms.Label();

this.ListSimilarWords = new System.Windows.Forms.ListBox();

this.SuspendLayout();

//

// LoadTextFile

//

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

this.LoadTextFile.Name = "LoadTextFile";

this.LoadTextFile.Size = new System.Drawing.Size(75, 61);

this.LoadTextFile.TabIndex = 0;

this.LoadTextFile.Text = "Чтение из файла";

this.LoadTextFile.UseVisualStyleBackColor = true;

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

//

// ReadTime

//

this.ReadTime.Location = new System.Drawing.Point(230, 12);

this.ReadTime.Name = "ReadTime";

this.ReadTime.Size = new System.Drawing.Size(123, 20);

this.ReadTime.TabIndex = 1;

//

// WordsCount

//

this.WordsCount.Location = new System.Drawing.Point(230, 53);

this.WordsCount.Name = "WordsCount";

this.WordsCount.Size = new System.Drawing.Size(123, 20);

this.WordsCount.TabIndex = 2;

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(128, 15);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(80, 13);

this.label1.TabIndex = 3;

this.label1.Text = "Время чтения:";

//

// label2

//

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(128, 56);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(96, 13);

this.label2.TabIndex = 4;

this.label2.Text = "Количество слов:";

//

// InputWord

//

this.InputWord.Location = new System.Drawing.Point(131, 133);

this.InputWord.Name = "InputWord";

this.InputWord.Size = new System.Drawing.Size(123, 20);

this.InputWord.TabIndex = 5;

//

// FindWord

//

this.FindWord.Location = new System.Drawing.Point(278, 133);

this.FindWord.Name = "FindWord";

this.FindWord.Size = new System.Drawing.Size(75, 61);

this.FindWord.TabIndex = 6;

this.FindWord.Text = "Найти слово";

this.FindWord.UseVisualStyleBackColor = true;

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

//

// label4

//

this.label4.AutoSize = true;

this.label4.Location = new System.Drawing.Point(12, 177);

this.label4.Name = "label4";

this.label4.Size = new System.Drawing.Size(82, 13);

this.label4.TabIndex = 9;

this.label4.Text = "Время поиска:";

//

// FindTime

//

this.FindTime.Location = new System.Drawing.Point(131, 174);

this.FindTime.Name = "FindTime";

this.FindTime.Size = new System.Drawing.Size(123, 20);

this.FindTime.TabIndex = 8;

//

// ListFindWords

//

this.ListFindWords.FormattingEnabled = true;

this.ListFindWords.Location = new System.Drawing.Point(12, 228);

this.ListFindWords.Name = "ListFindWords";

this.ListFindWords.Size = new System.Drawing.Size(341, 173);

this.ListFindWords.TabIndex = 10;

//

// label5

//

this.label5.AutoSize = true;

this.label5.Location = new System.Drawing.Point(125, 476);

this.label5.Name = "label5";

this.label5.Size = new System.Drawing.Size(99, 13);

this.label5.TabIndex = 15;

this.label5.Text = "Макс расстояние:";

//

// MaxDistance

//

this.MaxDistance.Location = new System.Drawing.Point(230, 473);

this.MaxDistance.Name = "MaxDistance";

this.MaxDistance.Size = new System.Drawing.Size(123, 20);

this.MaxDistance.TabIndex = 13;

this.MaxDistance.Text = "0";

//

// CheckWord

//

this.CheckWord.Location = new System.Drawing.Point(230, 435);

this.CheckWord.Name = "CheckWord";

this.CheckWord.Size = new System.Drawing.Size(123, 20);

this.CheckWord.TabIndex = 12;

//

// button1

//

this.button1.Location = new System.Drawing.Point(12, 432);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(75, 61);

this.button1.TabIndex = 11;

this.button1.Text = "Поиск похожих слов";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.Levenshtein\_Click);

//

// label3

//

this.label3.AutoSize = true;

this.label3.Location = new System.Drawing.Point(12, 136);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(101, 13);

this.label3.TabIndex = 7;

this.label3.Text = "Слово для поиска:";

//

// label6

//

this.label6.AutoSize = true;

this.label6.Location = new System.Drawing.Point(123, 438);

this.label6.Name = "label6";

this.label6.Size = new System.Drawing.Size(101, 13);

this.label6.TabIndex = 16;

this.label6.Text = "Слово для поиска:";

//

// ListSimilarWords

//

this.ListSimilarWords.FormattingEnabled = true;

this.ListSimilarWords.Location = new System.Drawing.Point(12, 523);

this.ListSimilarWords.Name = "ListSimilarWords";

this.ListSimilarWords.Size = new System.Drawing.Size(341, 173);

this.ListSimilarWords.TabIndex = 17;

//

// Form1

//

this.ClientSize = new System.Drawing.Size(366, 771);

this.Controls.Add(this.ListSimilarWords);

this.Controls.Add(this.label6);

this.Controls.Add(this.label5);

this.Controls.Add(this.MaxDistance);

this.Controls.Add(this.CheckWord);

this.Controls.Add(this.button1);

this.Controls.Add(this.ListFindWords);

this.Controls.Add(this.label4);

this.Controls.Add(this.FindTime);

this.Controls.Add(this.label3);

this.Controls.Add(this.FindWord);

this.Controls.Add(this.InputWord);

this.Controls.Add(this.label2);

this.Controls.Add(this.label1);

this.Controls.Add(this.WordsCount);

this.Controls.Add(this.ReadTime);

this.Controls.Add(this.LoadTextFile);

this.Name = "Form1";

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Button LoadTextFile;

private System.Windows.Forms.TextBox ReadTime;

private System.Windows.Forms.TextBox WordsCount;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.TextBox InputWord;

private System.Windows.Forms.Button FindWord;

private System.Windows.Forms.Label label4;

private System.Windows.Forms.TextBox FindTime;

private System.Windows.Forms.ListBox ListFindWords;

private System.Windows.Forms.Label label5;

private System.Windows.Forms.TextBox MaxDistance;

private System.Windows.Forms.TextBox CheckWord;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.Label label6;

private System.Windows.Forms.ListBox ListSimilarWords;

}

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private List<string> ListWords = new List<string>();

//Загрузка текстового файла

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

{

ListWords.Clear();

OpenFileDialog dialog = new OpenFileDialog();

dialog.Filter = "Текстовый файл (.txt)|\*.txt";

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

{

char[] separators = new char[] { ' ', ',', '.', '?', '!', '/', '|', '"', '\n', '\t', '\_', '-', '(', ')', '\*', '{', '}', '[', ']' };

Stopwatch timer = new Stopwatch();

timer.Start();

string filetext = File.ReadAllText(dialog.FileName);

string[] ListWordsBuf = filetext.Split(separators);

foreach (string word in ListWordsBuf)

{

if (!ListWords.Contains(word)) ListWords.Add(word);

}

this.ReadTime.Text = timer.Elapsed.ToString();

this.WordsCount.Text = ListWords.Count.ToString();

}

else

{

MessageBox.Show("Ошибка! Файл не выбран!");

}

}

//Поиск одинаковых слов

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

{

string findword = this.InputWord.Text.Trim();

if (!string.IsNullOrWhiteSpace(findword) && ListWords.Count > 0)

{

string findwordup = findword.ToUpper();

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

Stopwatch timer = new Stopwatch();

timer.Start();

foreach (string word in ListWords)

{

if (word.ToUpper().Contains(findwordup))

{

FindWords.Add(word);

}

}

timer.Stop();

this.FindTime.Text = timer.Elapsed.ToString();

this.ListFindWords.BeginUpdate();

this.ListFindWords.Items.Clear();

foreach (string word in FindWords)

{

this.ListFindWords.Items.Add(word);

}

this.ListFindWords.EndUpdate();

}

}

//Нахождения расстояния Левенштейна

public static int LevenshteinDistance(string str1, string str2)

{

//Проверка на исключительные случаи

if ((str1 == null && str2 == null) || (str1 == str2)) return 0;

if (str1 == null || str2 == null) throw new ArgumentNullException("Одна из строк пустая!\n");

//Алгоритм Вагнера — Фишера

int[,] matrix = new int[str1.Length + 1, str2.Length + 1];

for (int i = 0; i <= str1.Length; i++)

{

matrix[i, 0] = i;

}

for (int j = 0; j <= str2.Length; j++)

{

matrix[0, j] = j;

}

for (int i = 1; i <= str1.Length; i++)

{

for (int j = 1; j <= str2.Length; j++)

{

int d = 1;

if (str1[i - 1] == str2[j - 1]) d = 0;

matrix[i, j] = Math.Min(Math.Min(matrix[i - 1, j] + 1, matrix[i, j - 1] + 1), matrix[i - 1, j - 1] + d);

}

}

return matrix[str1.Length, str2.Length];

}

//Поиск слов по расстоянию Левенштейна

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

{

string checkword = this.CheckWord.Text.Trim();

int maxdistance = Convert.ToInt32(this.MaxDistance.Text.Trim());

if (!string.IsNullOrWhiteSpace(checkword) && ListWords.Count > 0)

{

string checkwordup = checkword.ToUpper();

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

foreach (string word in ListWords)

{

int distance = LevenshteinDistance(checkwordup, word.ToUpper());

if (distance <= maxdistance) SimilarWords.Add(word);

}

this.ListSimilarWords.BeginUpdate();

this.ListSimilarWords.Items.Clear();

foreach (string word in SimilarWords)

{

this.ListSimilarWords.Items.Add(word);

}

this.ListSimilarWords.EndUpdate();

}

}

}

# Результат работы программы

