**Отчёт по лабораторной работе №5  
 по курсу  
 «Базовые компоненты интернет-технологий»**

Выполнил студент:

Бакулин Глеб Георгиевич группа: РТ5-31

Проверил:

к.т.н., доцент

Гапанюк Юрий Евгеньевич

Москва, 2017

**Описание задания лабораторной работы:**

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.

**Исходный код программы:**

**Код Form1.cs:**

|  |
| --- |
| 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 WindowsFormsApp1; |
|  |  |
|  | namespace WindowsFormsApp1 |
|  | { |
|  | public partial class Form1 : Form |
|  | { |
|  | public Form1() |
|  | { |
|  | InitializeComponent(); |
|  | } |
|  | List<string> list = new List<string>(); |
|  | private void button1\_Click(object sender, EventArgs e) |
|  | { |
|  | OpenFileDialog FileDownload = new OpenFileDialog(); |
|  | FileDownload.Filter = "текстовые файлы |\*.txt"; |
|  | if (FileDownload.ShowDialog() == DialogResult.OK) |
|  | { |
|  | Stopwatch t = new Stopwatch(); |
|  | t.Start(); |
|  | string text = File.ReadAllText(FileDownload.FileName); |
|  | char[] separators = new char[] {' ','.',',','!','?','/','\t','\n'}; |
|  | string[] textArray = text.Split(separators); |
|  | foreach (string strTemp in textArray) |
|  | { |
|  | string str = strTemp.Trim(); |
|  | if (!list.Contains(str)) list.Add(str); |
|  | } |
|  | t.Stop(); |
|  | this.textBoxFileReadTime.Text = t.Elapsed.ToString(); |
|  | this.textBoxFileReadCount.Text = list.Count.ToString(); |
|  | } |
|  | else |
|  | { |
|  | MessageBox.Show("Необходимо выбрать файл"); |
|  | } |
|  | } |
|  |  |
|  | private void button2\_Click(object sender, EventArgs e) |
|  | { |
|  | string word = this.textBoxFind.Text.Trim(); |
|  |  |
|  | //Если слово для поиска не пусто |
|  | if (!string.IsNullOrWhiteSpace(word) && list.Count > 0) |
|  | { |
|  | //Слово для поиска в верхнем регистре |
|  | string wordUpper = word.ToUpper(); |
|  | //Временные результаты поиска |
|  | List<string> tempList = new List<string>(); |
|  | Stopwatch t = new Stopwatch(); |
|  | t.Start(); |
|  | foreach (string str in list) |
|  | { |
|  | if (str.ToUpper().Contains(wordUpper)) |
|  | { |
|  | tempList.Add(str); |
|  | } |
|  | } |
|  | t.Stop(); |
|  | this.textBoxExactTime.Text = t.Elapsed.ToString(); |
|  | this.listBoxResult.BeginUpdate(); |
|  | //Очистка списка |
|  | this.listBoxResult.Items.Clear(); |
|  | //Вывод результатов поиска |
|  | foreach (string str in tempList) |
|  | { |
|  | this.listBoxResult.Items.Add(str); |
|  | } |
|  | this.listBoxResult.EndUpdate(); |
|  | } |
|  | else |
|  | { |
|  | MessageBox.Show("Необходимо выбрать файл и ввести слово для поиска"); |
|  | } |
|  | } |
|  |  |
|  | private void buttonApprox\_Click(object sender, EventArgs e) |
|  | { |
|  | //Слово для поиска |
|  | string word = this.textBoxFind.Text.Trim(); |
|  | //Если слово для поиска не пусто |
|  | if (!string.IsNullOrWhiteSpace(word) && list.Count > 0) |
|  | { |
|  | int maxDist; |
|  | if (!int.TryParse(this.textBoxMaxDist.Text.Trim(), out maxDist)) |
|  | { |
|  | MessageBox.Show("Необходимо указать максимальное расстояние"); |
|  | return; |
|  | } |
|  | if (maxDist < 1 || maxDist > 5) |
|  | { |
|  | MessageBox.Show("Максимальное расстояние должно быть в диапазоне от 1 до 5"); |
|  | return; |
|  | } |
|  | //Слово для поиска в верхнем регистре |
|  | string wordUpper = word.ToUpper(); |
|  | //Временные результаты поиска |
|  | List<Tuple<string, int>> tempList = new List<Tuple<string, |
|  | int>>(); |
|  | Stopwatch t = new Stopwatch(); |
|  | t.Start(); |
|  | foreach (string str in list) |
|  | { |
|  | //Вычисление расстояния Дамерау-Левенштейна |
|  | int dist = EditDistance.Distance(str.ToUpper(), wordUpper); |
|  | //Если расстояние меньше порогового, то слово добавляется в результат |
|  | if (dist <= maxDist) |
|  | { |
|  | tempList.Add(new Tuple<string, int>(str, dist)); |
|  | } |
|  | } |
|  | t.Stop(); |
|  | this.textBoxApproxTime.Text = t.Elapsed.ToString(); |
|  | this.listBoxResult.BeginUpdate(); |
|  | //Очистка списка |
|  | this.listBoxResult.Items.Clear(); |
|  | //Вывод результатов поиска |
|  | foreach (var x in tempList) |
|  | { |
|  | string temp = x.Item1 + "(расстояние=" + x.Item2.ToString() + |
|  | ")"; |
|  | this.listBoxResult.Items.Add(temp); |
|  | } |
|  | this.listBoxResult.EndUpdate(); |
|  | } |
|  | else |
|  | { |
|  | MessageBox.Show("Необходимо выбрать файл и ввести слово для поиска"); |
|  | } |
|  | }  }  } |

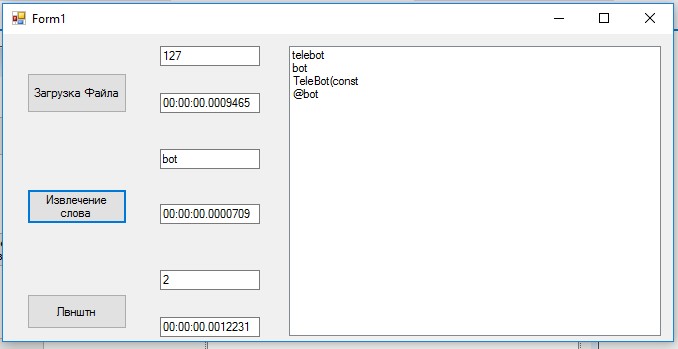
**Код EditDistance.cs:**

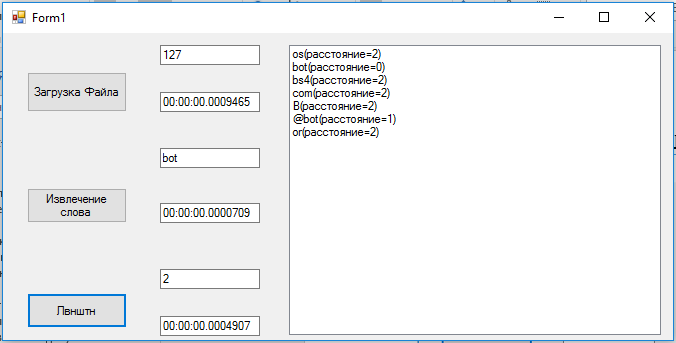
|  |
| --- |
| using System; |
|  | using System.Collections.Generic; |
|  | using System.Linq; |
|  | using System.Text; |
|  | using System.Threading.Tasks; |
|  |  |
|  | namespace WindowsFormsApp1 |
|  | { |
|  | public static class EditDistance |
|  | { |
|  | /// <summary> |
|  | /// Вычисление 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]; |
|  | } |
|  | } |
|  | } |

**Код Form1.Designer.cs:**

|  |
| --- |
| namespace WindowsFormsApp1 |
|  | { |
|  | 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.button1 = new System.Windows.Forms.Button(); |
|  | this.textBoxFileReadTime = new System.Windows.Forms.TextBox(); |
|  | this.textBoxFileReadCount = new System.Windows.Forms.TextBox(); |
|  | this.button2 = new System.Windows.Forms.Button(); |
|  | this.textBoxFind = new System.Windows.Forms.TextBox(); |
|  | this.textBoxExactTime = new System.Windows.Forms.TextBox(); |
|  | this.listBoxResult = new System.Windows.Forms.ListBox(); |
|  | this.buttonApprox = new System.Windows.Forms.Button(); |
|  | this.textBoxMaxDist = new System.Windows.Forms.TextBox(); |
|  | this.textBoxApproxTime = new System.Windows.Forms.TextBox(); |
|  | this.SuspendLayout(); |
|  | // |
|  | // button1 |
|  | // |
|  | this.button1.Location = new System.Drawing.Point(24, 39); |
|  | this.button1.Name = "button1"; |
|  | this.button1.Size = new System.Drawing.Size(100, 40); |
|  | this.button1.TabIndex = 0; |
|  | this.button1.Text = "Загрузка Файла"; |
|  | this.button1.UseVisualStyleBackColor = true; |
|  | this.button1.Click += new System.EventHandler(this.button1\_Click); |
|  | // |
|  | // textBoxFileReadTime |
|  | // |
|  | this.textBoxFileReadTime.Location = new System.Drawing.Point(157, 59); |
|  | this.textBoxFileReadTime.Name = "textBoxFileReadTime"; |
|  | this.textBoxFileReadTime.Size = new System.Drawing.Size(100, 20); |
|  | this.textBoxFileReadTime.TabIndex = 2; |
|  | // |
|  | // textBoxFileReadCount |
|  | // |
|  | this.textBoxFileReadCount.Location = new System.Drawing.Point(157, 12); |
|  | this.textBoxFileReadCount.Name = "textBoxFileReadCount"; |
|  | this.textBoxFileReadCount.Size = new System.Drawing.Size(100, 20); |
|  | this.textBoxFileReadCount.TabIndex = 3; |
|  | // |
|  | // button2 |
|  | // |
|  | this.button2.Location = new System.Drawing.Point(24, 155); |
|  | this.button2.Name = "button2"; |
|  | this.button2.Size = new System.Drawing.Size(100, 35); |
|  | this.button2.TabIndex = 4; |
|  | this.button2.Text = "Извлечение слова"; |
|  | this.button2.UseVisualStyleBackColor = true; |
|  | this.button2.Click += new System.EventHandler(this.button2\_Click); |
|  | // |
|  | // textBoxFind |
|  | // |
|  | this.textBoxFind.Location = new System.Drawing.Point(157, 115); |
|  | this.textBoxFind.Name = "textBoxFind"; |
|  | this.textBoxFind.Size = new System.Drawing.Size(100, 20); |
|  | this.textBoxFind.TabIndex = 5; |
|  | // |
|  | // textBoxExactTime |
|  | // |
|  | this.textBoxExactTime.Location = new System.Drawing.Point(157, 170); |
|  | this.textBoxExactTime.Name = "textBoxExactTime"; |
|  | this.textBoxExactTime.Size = new System.Drawing.Size(100, 20); |
|  | this.textBoxExactTime.TabIndex = 6; |
|  | // |
|  | // listBoxResult |
|  | // |
|  | this.listBoxResult.FormattingEnabled = true; |
|  | this.listBoxResult.Location = new System.Drawing.Point(286, 12); |
|  | this.listBoxResult.Name = "listBoxResult"; |
|  | this.listBoxResult.Size = new System.Drawing.Size(372, 290); |
|  | this.listBoxResult.TabIndex = 7; |
|  | this.listBoxResult.SelectedIndexChanged += new System.EventHandler(this.listBoxResult\_SelectedIndexChanged); |
|  | // |
|  | // buttonApprox |
|  | // |
|  | this.buttonApprox.Location = new System.Drawing.Point(24, 260); |
|  | this.buttonApprox.Name = "buttonApprox"; |
|  | this.buttonApprox.Size = new System.Drawing.Size(100, 35); |
|  | this.buttonApprox.TabIndex = 8; |
|  | this.buttonApprox.Text = "Лвнштн"; |
|  | this.buttonApprox.UseVisualStyleBackColor = true; |
|  | this.buttonApprox.Click += new System.EventHandler(this.buttonApprox\_Click); |
|  | // |
|  | // textBoxMaxDist |
|  | // |
|  | this.textBoxMaxDist.Location = new System.Drawing.Point(157, 236); |
|  | this.textBoxMaxDist.Name = "textBoxMaxDist"; |
|  | this.textBoxMaxDist.Size = new System.Drawing.Size(100, 20); |
|  | this.textBoxMaxDist.TabIndex = 9; |
|  | this.textBoxMaxDist.TextChanged += new System.EventHandler(this.textBoxMaxDist\_TextChanged); |
|  | // |
|  | // textBoxApproxTime |
|  | // |
|  | this.textBoxApproxTime.Location = new System.Drawing.Point(157, 283); |
|  | this.textBoxApproxTime.Name = "textBoxApproxTime"; |
|  | this.textBoxApproxTime.Size = new System.Drawing.Size(100, 20); |
|  | this.textBoxApproxTime.TabIndex = 10; |
|  | this.textBoxApproxTime.TextChanged += new System.EventHandler(this.textBoxApproxTime\_TextChanged); |
|  | // |
|  | // Form1 |
|  | // |
|  | this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F); |
|  | this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font; |
|  | this.ClientSize = new System.Drawing.Size(670, 307); |
|  | this.Controls.Add(this.textBoxApproxTime); |
|  | this.Controls.Add(this.textBoxMaxDist); |
|  | this.Controls.Add(this.buttonApprox); |
|  | this.Controls.Add(this.listBoxResult); |
|  | this.Controls.Add(this.textBoxExactTime); |
|  | this.Controls.Add(this.textBoxFind); |
|  | this.Controls.Add(this.button2); |
|  | this.Controls.Add(this.textBoxFileReadCount); |
|  | this.Controls.Add(this.textBoxFileReadTime); |
|  | this.Controls.Add(this.button1); |
|  | this.Name = "Form1"; |
|  | this.Text = "Form1"; |
|  | this.ResumeLayout(false); |
|  | this.PerformLayout(); |
|  |  |
|  | } |
|  |  |
|  | #endregion |
|  |  |
|  | private System.Windows.Forms.Button button1; |
|  | private System.Windows.Forms.TextBox textBoxFileReadTime; |
|  | private System.Windows.Forms.TextBox textBoxFileReadCount; |
|  | private System.Windows.Forms.Button button2; |
|  | private System.Windows.Forms.TextBox textBoxFind; |
|  | private System.Windows.Forms.TextBox textBoxExactTime; |
|  | private System.Windows.Forms.ListBox listBoxResult; |
|  | private System.Windows.Forms.Button buttonApprox; |
|  | private System.Windows.Forms.TextBox textBoxMaxDist; |
|  | private System.Windows.Forms.TextBox textBoxApproxTime; |
|  | } |
|  | } |

**Результаты выполнения программы:**



****