**Московский государственный технический университет**

**им. Н.Э. Баумана**

Кафедра ИУ–5

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

Выполнил:

студент гр. ИУ5-32Б

Деркачев А.В.

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

Проверил:

к.т.н., доцент

Гапанюк Ю.Е.

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

**Москва – 2014 г**

1. **Цель выполнения лабораторной работы**

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

1. **Листинг программы**

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace LR5FT

{ static class Program

{ /// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{ Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());}}}

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;

namespace LR5FT

{ public partial class Form1 : Form

{ string ishod;

string zel;

int rasst;

public Form1()

{InitializeComponent();}

private void Form1\_Load(object sender, EventArgs e)

{}

private void button1\_Click(object sender, EventArgs e)

{ishod = textBox1.Text;

zel = textBox2.Text;

label1.Text = LevenshteinDistance(ishod, zel).ToString();}

public static int LevenshteinDistance(string string1, string string2)

{ if (string1 == null) throw new ArgumentNullException("string1");

if (string2 == null) throw new ArgumentNullException("string2");

int diff;

int[,] m = new int[string1.Length + 1, string2.Length + 1];

for (int i = 0; i <= string1.Length; i++) m[i, 0] = i;

for (int j = 0; j <= string2.Length; j++) m[0, j] = j;

for (int i = 1; i <= string1.Length; i++)

for (int j = 1; j <= string2.Length; j++)

{diff = (string1[i - 1] == string2[j - 1]) ? 0 : 1;

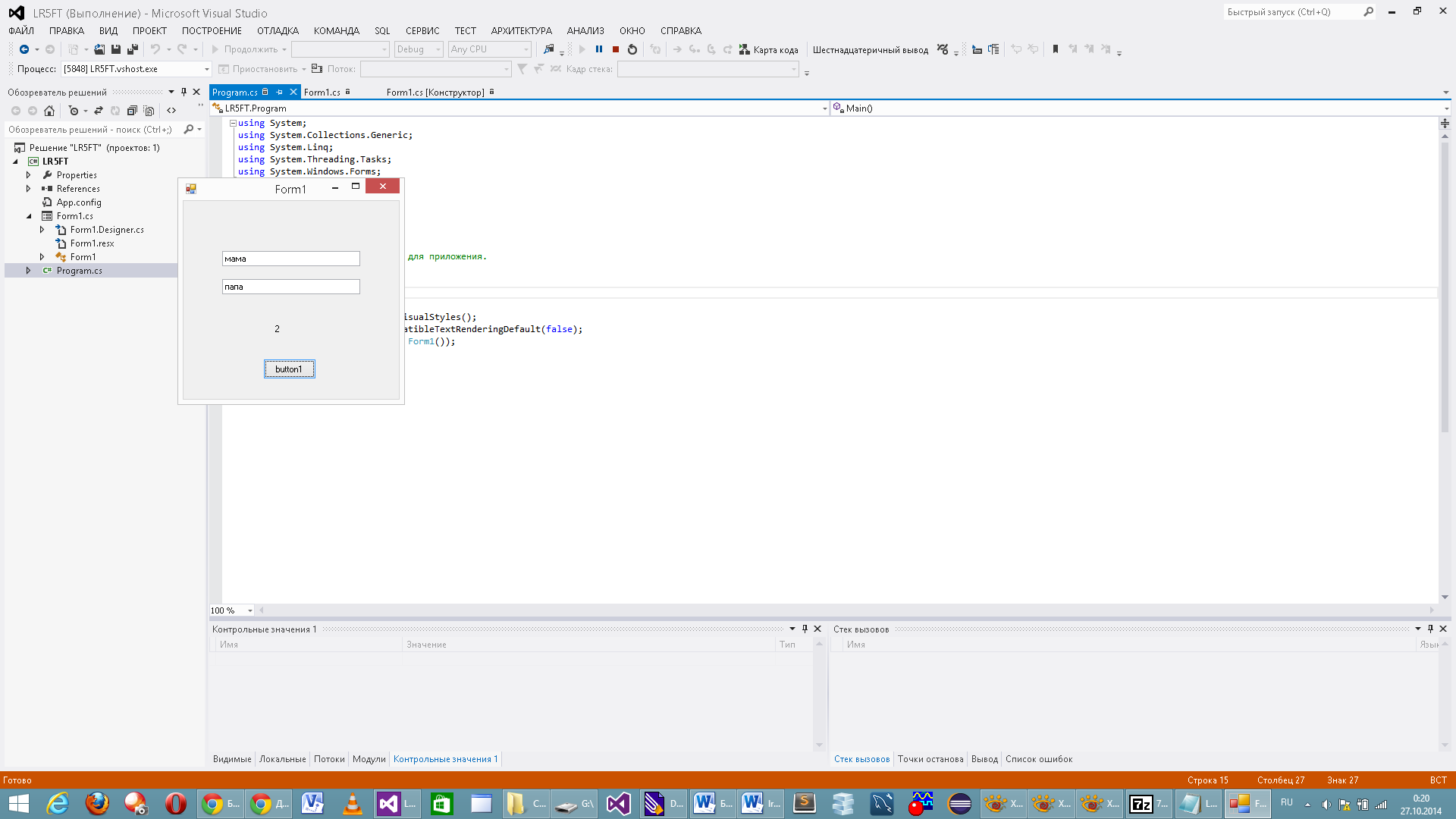
m[i, j] = Math.Min(Math.Min(m[i - 1, j] + 1,

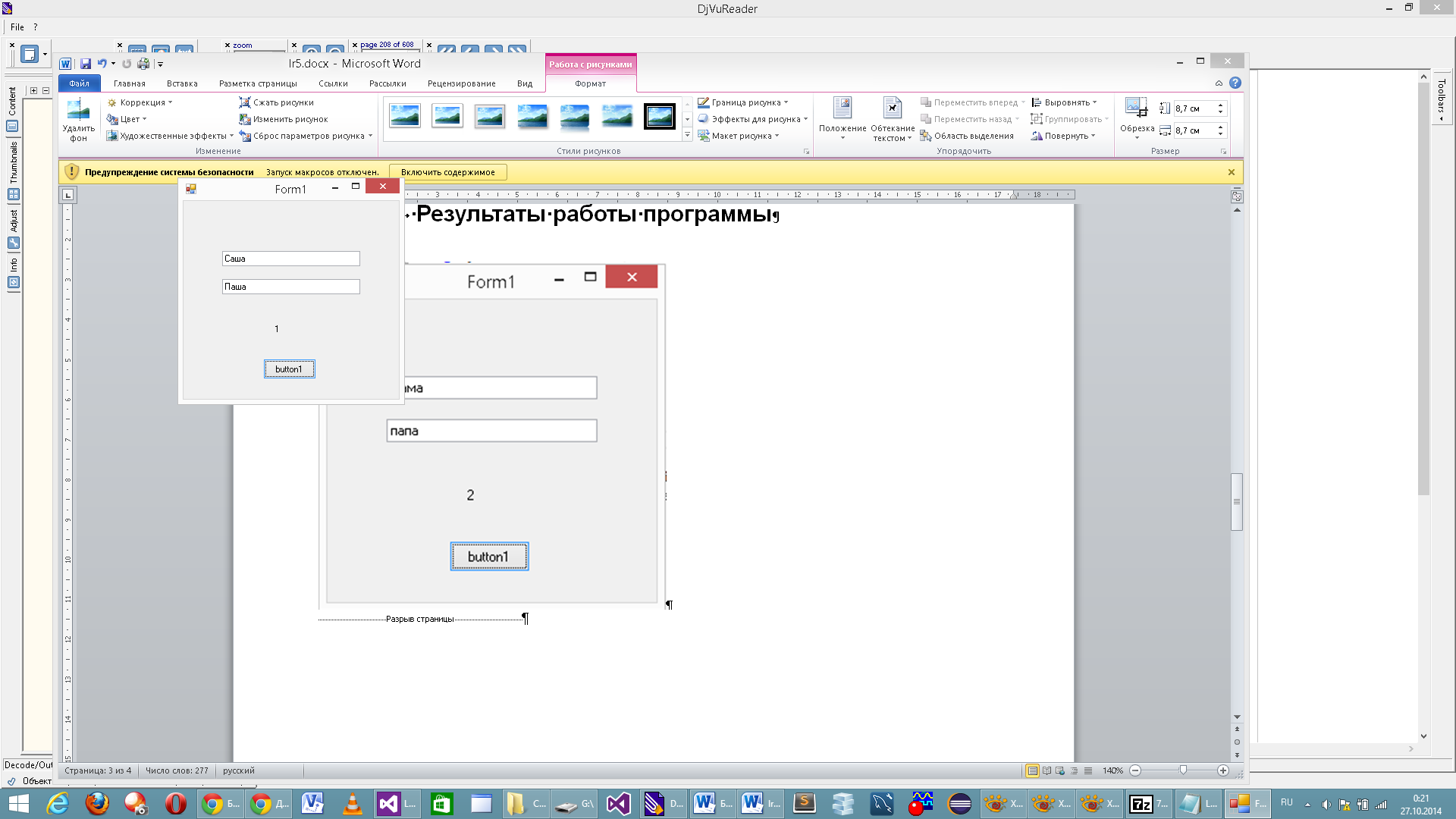
m[i, j - 1] + 1),

m[i - 1, j - 1] + diff);}

return m[string1.Length, string2.Length];}}}

1. **Результаты работы программы**





1. **Диаграмма классов**

