**Министерство науки и высшего образования Российской** **Федерации** **Федеральное государственное бюджетное образовательное учреждение высшего образования** **«Московский государственный технический университет** **имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Радиотехнический»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Лабораторная работа №5

по дисциплине «Базовые компоненты интернет-технологий» на тему:

«Нечеткий поиск. Алгоритм Вагнера-Фишера»

Выполнил:

студент группы РТ5-31Б

В. И. Плешаков

Проверил:

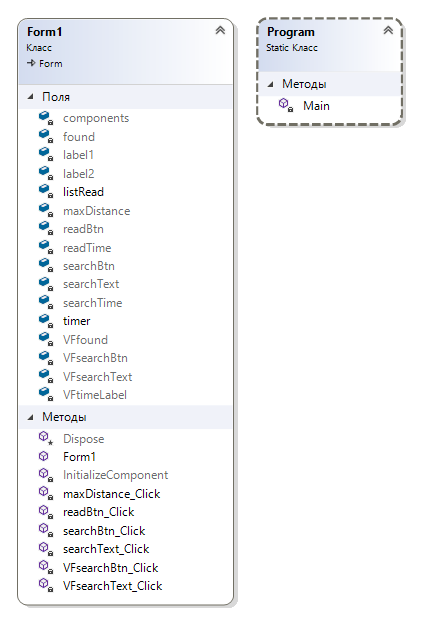
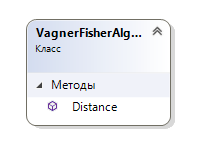
Ю. Е. Гапанюк

2020 г.

Задание:

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

Диаграмма классов:

Текст программы:

Реализация алгоритма Вагнера-Фишера:

using System;

namespace ClassLibrary1

{

public class VagnerFisherAlgorithm

{

public static int Distance(string str1, string str2)

{

if ((str1 == null) || (str2 == null)) return -1;

int str1Len = str1.Length;

int str2Len = str2.Length;

// Если одна из строк пустая, возвращается длина другой

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

if (str1Len == 0) return str2Len;

if (str2Len == 0) return str1Len;

// Приведение к верхнему регистру

string str1Up = str1.ToUpper();

string str2Up = str2.ToUpper();

int[,] matrix = new int[str1Len + 1, str2Len + 2];

// 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

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

{

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

{

// Проверка эквивалентности символов

int symbEqual = ((str1Up.Substring(i - 1, 1) == str2Up.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);// Выбираем минимальный из трех

// Учет перестановок соседних символов

if ((i > 1) && (j > 1) &&

(str1Up.Substring(i - 1, 1) == str2Up.Substring(j - 2, 1)) &&

(str1Up.Substring(i - 2, 1) == str2Up.Substring(j - 1, 1)))

{

matrix[i, j] = Math.Min(matrix[i, j], matrix[i - 2, j - 2] + symbEqual);

}

}

}

//Возвращается правый нижний элемент матрицы

return matrix[str1Len, str2Len];

}

}

}

Пример выполнения программы:

