МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего

образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

**ОТЧЕТ**

**по Лабораторной работе №5**

по дисциплине

Технология программирования

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Багиров М.Б.

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Вихрова С.С

(подпись) (фамилия, и.,о.)

18-ИСТ-4

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород   
2020

Оглавление

[Вариант №4 3](#_Toc37505861)

[Листинг программы 3](#_Toc37505862)

[Результат работы программы 7](#_Toc37505863)

# Вариант №4

Задается словарь. Найти в нем все анаграммы (слова, составленные из одних и тех же букв).

# Листинг программы

--Program.cs--

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Program

{

public List<string> dictionary = new List<string>();//основной словарь

static void Main(string[] args)

{

Program prog = new Program();

prog.ReadFile();//чтение из файла

Console.WriteLine("Количество слов" + prog.dictionary.Count());

for (int i = 0; i < prog.dictionary.Count(); i++) {

//Console.WriteLine("----------------" + i);

//Console.WriteLine(prog.dictionary[i]);

string line = prog.dictionary[i];

char[] lineString = line.ToCharArray();//разбор каждого слова на символы

//пузырькова сортировака для перестаноки всех символов в порядке алфавита

for (int k = 0; k < lineString.Length; k++)

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

if ((int)lineString[j] > (int)lineString[k])

{

var c = lineString[k];

lineString[k] = lineString[j];

lineString[j] = c;

}

//Array.Sort(lineString);

line = "";

//"собирание" слова обратно с уже перестановленными буквами

for (int j = 0; j < lineString.Count(); j++)

line += lineString[j];

//фукнция отыскания всех анаграм для данного слова

//этой функции будет передаваться индекс текущей позиции в словаре, перестановленное слово и то члово, которое было изначально

int ind = prog.Anagram(i, line, prog.dictionary[i]);

i = i + ind;//увеличение индекса, если в словаре переставились слова

}

//вывод всего словаря

for (int i = 0; i < prog.dictionary.Count(); i++)

{

Console.WriteLine(prog.dictionary[i]);

}

Console.ReadKey(true);

}

//функция для отыскания анаграмм

public int Anagram(int k, string line, string enterWord) {

List<String> anagrams = new List<string>();//лист со словами, которые будут является анаграммами для полученного слова

anagrams.Add(enterWord);//добавление изначального слова в словарь

for (int i= k+1; i<dictionary.Count(); i++)

{

char[] lineString = dictionary[i].ToCharArray();

Array.Sort(lineString);//обычный встроенный метод сортировки букв по алфавиту

string word = "";

for (int j = 0; j < lineString.Count(); j++)

word += lineString[j];

//если перестановленные по алфавиту буквы совпадают с изначальным словом

if (line == word)

{

//Console.WriteLine("--" + dictionary[i]+ "--") ;

anagrams.Add(dictionary[i]);

}

}

//если нашлись анаграммы

if (anagrams.Count > 1)

{

Console.WriteLine(anagrams[1]);

for (int i = 1; i < anagrams.Count(); i++)

{

Console.WriteLine(anagrams[i]);

dictionary.Remove(anagrams[i]);//удаляем их с текущего положения, чтобы в дальнейшем случайно не посчитать их ещё раз

}

Console.WriteLine("----------------");

for (int i = 1; i < anagrams.Count(); i++)

dictionary.Insert(k+i, anagrams[i]);//вставка полученных анаграмм после выбранного слова

return anagrams.Count - 1;//возвращение количества анаграмм для данногос слова

}

return 0;

}

//функция чтения из файла и заполнение словаря

public void ReadFile() {

StreamReader read = new StreamReader("Словарь.txt");

string line;

while (!read.EndOfStream) {

line = read.ReadLine();

dictionary.Add(line);

}

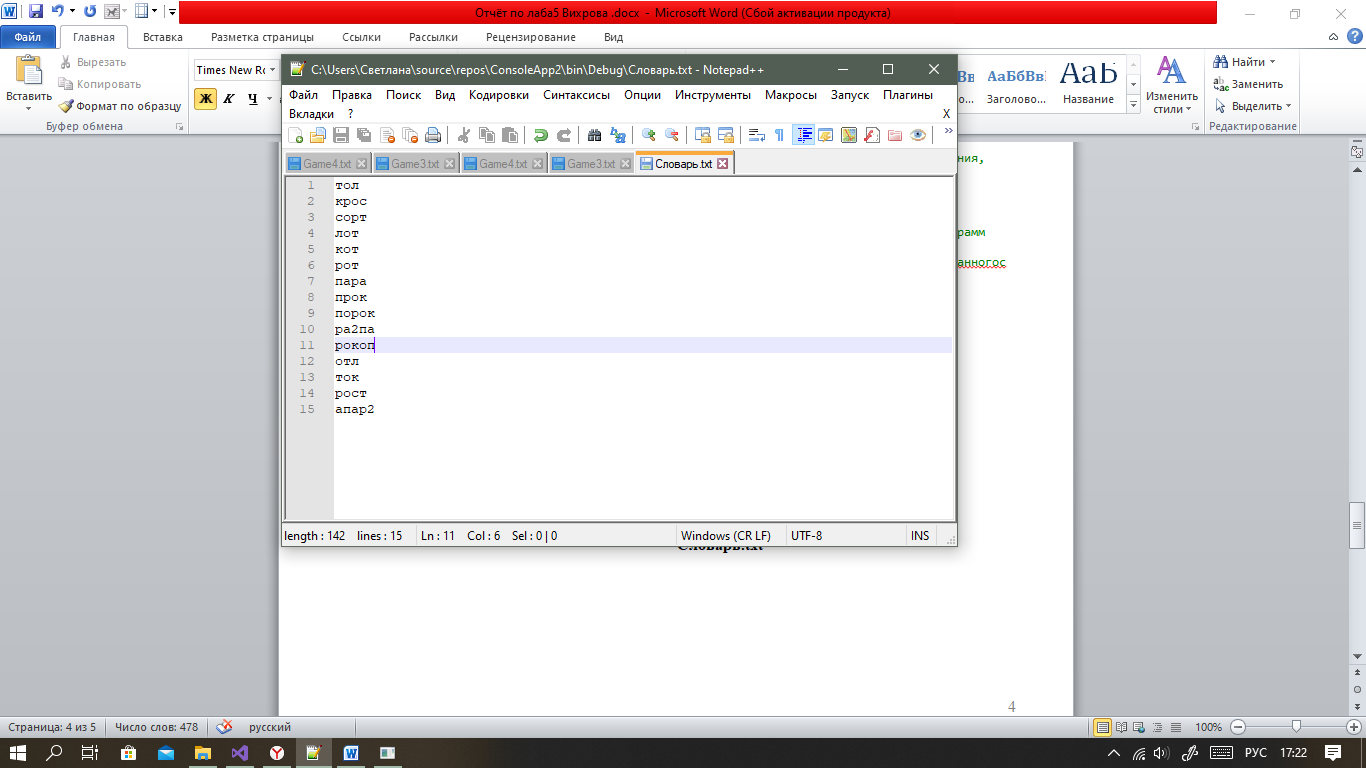
read.Close();

}

}

}

**--Словарь.txt—**



# Результат работы программы

