

**Лабораторная работа №4**  
*студента группы ИТ – 32*  
*Курбатовой Софьи Андреевны*

Выполнение: \_\_\_\_\_ Защита \_\_\_\_\_

**РЕАЛИЗАЦИЯ АЛГОРИТМА ИНДЕКСИРОВАНИЯ ДОКУМЕНТОВ И  
ПОИСКА ПО ИНДЕКСУ.**

**Цель работы:** научиться реализовывать на выбранном языке программирования алгоритмы индексирования документов и осуществлять поиск по индексу.

Содержание работы

Написать программу на выбранном языке программирования, реализующую указанный индексацию документов (не менее 5 документов) указанным выше алгоритмом и осуществить поиск документа, удовлетворяющего заданному запросу. Осуществить ранжирование документов по их релевантности запросу. Программа должна запрашивать имена входных файлов и выводить заголовки и выдержки из найденных по запросу документов.

Ход работы

1.

Листинг 1.1. Программ реализующая алгоритм поиска.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;

namespace invertindexsearch
{
    class Library
    {
        public string name { get; set; }
        public string path { get; set; }
        public double score { get; set; }
        public int flagall { get; set; }
        public int flagprase { get; set; }
        public Library()
        {
            name = "";
            path = "";
            score = 0;
            flagall = 0;
            flagprase = 0;
        }
        public Library(string n, string p)
        {
            name = n;
            path = p;
        }
        public Library(string n, string p, double s, int fa, int fp)
        {
            name = n;
```

```

        path = p;
        score = s;
        flagall = fa;
        flagprase = fp;
    }
}

class Program
{
    static void Find(ref List<Library> libraries, string request)
    {
        string[] requestword = request.Split(' ');
        int[] flag = new int[requestword.Length];
        foreach (Library l in libraries)
        {
            double Score = 0;
            double Wsingle = 0;
            double Wpair = 0;
            double Wall = 0;
            double Wphrase = 0; //встречаемость в запросе целиком
            int i = 0;
            for (i = 0; i < flag.Length; i++)
                flag[i] = 0;
            string text = "";
            string[] textword;
            text = File.ReadAllText(String.Concat(l.path, l.name));
            textword = text.Split(' ');
            for (i = 0; i < textword.Length; i++)
            {
                for (int j = 0; j < requestword.Length; j++)
                {
                    if (String.Compare(textword[i], requestword[j]) == 0)
                    {
                        Wsingle++; //подсчет встреченных слова
                        flag[j] = 1;
                        if (i + 1 < textword.Length && j + 1 < requestword.Length &&
String.Compare(textword[i + 1], requestword[j + 1]) == 0)
                        {
                            Wpair++; //подсчет если пара
                        }
                    }
                }
            }
            i = 0;
            while (i < flag.Length && flag[i] == 1)
                i++;
            if (i == flag.Length)
                Wall = 1;
            if (text.IndexOf(request) != -1)
                Wphrase = 1;
            Score = Wsingle + Wpair + Wall + Wphrase / 350; //расчет ранжирования
            l.score = Score;
            l.flagall = (int)Wall;
            l.flagprase = (int)Wphrase;
        }
    }

    static void Ranging(List<Library> libraries)
    {
        var t = libraries.OrderByDescending(a => a.score);
        bool f1 = false;
        bool f2 = false;
        foreach (var k in t)
        {
            f1 = false;
            f2 = false;
            if (k.flagall == 1)

```

```

        f1 = true;
        if (k.flagprase == 1)
            f2 = true;
        Console.WriteLine(String.Format("Документ {0}:", k.name));
        Console.WriteLine(String.Format("Значение score: {0:f4}", k.score));
        Console.WriteLine(String.Format("Встречаются ли все слова: {0}", f1.ToString()));
        Console.WriteLine(String.Format("Слова в нужном порядке: {0}", f2.ToString()));
    }
}

static void Main(string[] args)
{
    char ch;
    string path = @"C:\test\libr\";
    string name = ".txt";
    List<Library> libraries = new List<Library>();
    for (int i = 0; i < 5; i++)
    {
        libraries.Add(new Library(String.Concat(i + 1, name), path));
    }
    do
    {
        Console.WriteLine("Введите запрос");
        string request = Console.ReadLine();
        Find(ref libraries, request);
        Ranging(libraries);
        ch = Console.ReadKey().KeyChar;
    } while (ch != 8);
}
}

```

## 2. Тестирование:

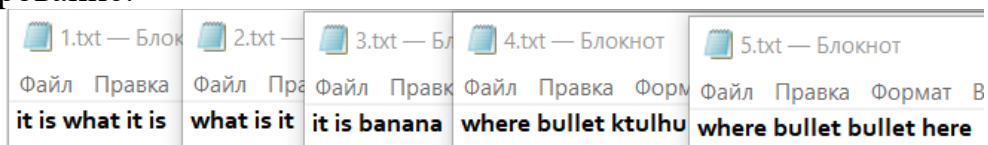


Рис. 4.1. Содержимое корпуса документов

```

C:\Users\Kurbatova\source\LW2020\lw Введите запрос
Введите запрос
bullet
Документ 5.txt:
Значение score: 3,0029
Встречаются ли все слова: True
Слова в нужном порядке: True
Документ 4.txt:
Значение score: 2,0029
Встречаются ли все слова: True
Слова в нужном порядке: True
Документ 1.txt:
Значение score: 0,0000
Встречаются ли все слова: False
Слова в нужном порядке: False
Документ 2.txt:
Значение score: 0,0000
Встречаются ли все слова: False
Слова в нужном порядке: False
Документ 3.txt:
Значение score: 0,0000
Встречаются ли все слова: False
Слова в нужном порядке: False
Документ 4.txt:
Значение score: 0,0000
Встречаются ли все слова: False
Слова в нужном порядке: False
Документ 5.txt:
Значение score: 0,0000
Встречаются ли все слова: False
Слова в нужном порядке: False
what is
Документ 2.txt:
Значение score: 4,0029
Встречаются ли все слова: True
Слова в нужном порядке: True
Документ 1.txt:
Значение score: 4,0000
Встречаются ли все слова: True
Слова в нужном порядке: False
Документ 3.txt:
Значение score: 1,0000
Встречаются ли все слова: False
Слова в нужном порядке: False
Документ 4.txt:
Значение score: 0,0000
Встречаются ли все слова: False
Слова в нужном порядке: False

```

Рис. 4.2. Результат работы

**Вывод:** Таким образом, в ходе выполнения лабораторной работы на языке C# были реализованы алгоритмы