

Базовые компоненты интернет технологий.
Отчёт по домашнему заданию.

Белкина Екатерина
Группа ИУ5-31Б

24 декабря 2018

Условие задания:

Пример реализации ДЗ рассмотрен в учебном пособии, глава «Пример многопоточного поиска в текстовом файле с использованием технологии Windows Forms».

Разработать программу, реализующую многопоточный поиск в файле.

1. Программа должна быть разработана в виде приложения Windows Forms на языке C#. По желанию вместо Windows Forms возможно использование WPF.
2. В качестве основы используется макет, разработанный в лабораторных работах №4 и №5.
3. Реализуйте функцию поиска с использованием расстояния Левенштейна в многопоточном варианте. Количество потоков для запуска функции поиска вводится на форме в поле ввода (TextBox).
4. Реализуйте функцию записи результатов поиска в файл отчета. Файл отчета создается в формате .txt или .html.

Текст программы

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using Microsoft.Win32;

using Lab5; //при работе программы используется текст, приведённый в отчёте к 5
            лабораторной работе

namespace Homework
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
```

/// </summary>

```
public partial class MainWindow : Window
```

```
{
    private List<string> _words;
    public MainWindow()
    {
        InitializeComponent();
    }

    private void OpenFile_Click(object sender, RoutedEventArgs e)
    {
        var dialog = new OpenFileDialog
        {
            Filter = "Text Files(*.txt) | *.txt"
        };

        var timer = new Stopwatch();

        if (dialog.ShowDialog() == true)
        {
            timer.Start();
            var file = File.ReadAllText(dialog.FileName);
            _words = file.Trim('.').Trim(',').Split(' ').Distinct().ToList();
            timer.Stop();

            openTimer.Content = "Время чтения: " + timer.ElapsedMilliseconds + " мс";
        }
    }

    private void Search_Click(object sender, RoutedEventArgs e)
    {
        listBox.Items.Clear();

        var expectedWord = findWord.Text.Trim(' ');

        if (expectedWord == "")
        {
            listBox.Items.Add("Введите слово для поиска");
            return;
        }
    }
}
```

```
}
```

```
if (maxDist.Text.Trim(' ') == "")  
{  
    listBox.Items.Add("Введите максимальное расстояние");  
    return;  
}
```

```
if (countOfThreads.Text.Trim(' ') == "")  
{  
    listBox.Items.Add("Введите количество потоков");  
    return;  
}
```

```
var dist = int.Parse(maxDist.Text);  
var tasksCount = int.Parse(countOfThreads.Text);  
var timer = new Stopwatch();  
var tasks = new Task<List<string>>[tasksCount];
```

```
var listOfWords = new List<List<string>>();  
for (var i = 0; i < tasksCount; i++)  
{  
    listOfWords.Add(new List<string>());  
}
```

```
for (var i = 0; i < _words.Count; i += tasksCount)  
{  
    for (var j = 0; j < tasksCount && i + j < _words.Count; j++)  
    {  
        listOfWords[j].Add(_words[i + j]);  
    }  
}
```

```
timer.Start();
```

```
for (var i = 0; i < tasksCount; i++)  
{  
    var list = listOfWords[i];  
    tasks[i] = Task.Run(() =>  
    {
```

```

        var findList = new List<string>();
        foreach (var word in list)
        {
            if (Lab5.Lab5.Dist(word, expectedWord) <= dist)
            {
                findList.Add(word);
            }
        }

        return findList;
    });
}

```

```

Task.WaitAll(tasks);

```

```

foreach (var task in tasks)
{
    foreach (var word in task.Result)
    {
        listBox.Items.Add(word);
    }
}

```

```

timer.Stop();

```

```

        Console.WriteLine("Count of tasks: " + tasksCount + " has found " +
listBox.Items.Count + " words");

```

```

if (listBox.Items.Count == 0)
{
    listBox.Items.Add("Нет совпадений");
}

```

```

searchTimer.Content = "Время поиска: " + timer.ElapsedMilliseconds + " мс";

```

```

    }
}
}

```

Примеры выполнения программы

Открыть файл

Время чтения: 9 мс

Максимальное расстояние:

Слово для поиска:

Количество потоков:

Найти

Время поиска: 38 мс

Lorem
consectetur
mi
Donec
tincidunt,
vel
tortor.
ac
Etiam
convallis
diam
dui

Открыть файл

Время чтения: 9 мс

Максимальное расстояние:

Слово для поиска:

Количество потоков:

Найти

Время поиска: 33 мс

Lorem
elit.
blandit.
sapien
nisi
ac
est
rutrum
aliquet
velit
purus
amet