**Лабораторная работа №5 « СТРОКИ И РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ»**

# Задания для работы со строками

1. Дан текст. Определить количество слов, начинающихся и оканчивающихся одной и той же буквой.

# Задания на регулярные выражения

1. Дан текстовый файл. Составить регулярное выражение, определяющее, является ли данная строка шестнадцатеричным идентификатором цвета в HTML (где #FFFFFF- для белого, #000000 для черного, #FF0000 для красного и.т.д.).

**1.Постановка задачи**

Данный отчет будет сделан для второй части на тему «**Регулярные выражения**».

**Среда разработки**

Необходимо разработать визуальное приложение в среде разработки Visual Studio Community 2019 версии 16.3.0 C++ в соответствии с заданием. Продемонстрировать работу программы и выявить проблемы возникшие при разработки программы.

В качестве входных данных будет выступать текстовый документ формата .txt. *(Нажав на определенную кнопку, пользователь сумеет выбрать нужный ему файл и продолжить работу уже с ним)*

В качестве выходных данных планируется использовать элемент формы, для вывода сообщения о наличии в строке шестнадцатеричного идентификатора цвета в HTML.

**Ограничения и граничные случаи:**

1. Идентификатором цвета в HTML будет считаться последовательность символов: #FFFFFFX, где # - символ «#», F – любое шестнадцатеричная цифра (Их должно быть ровно 6 штук), а Х – любой символ не являющийся шестнадцатеричной цифрой. (Пример: строка «#12F456G» содержит идентификатор цвета, а строка «#12F456F» не содержит).

**2. Разработка программы**

**2.1 Разработка структуры программы**

В соответствии с заданием разработаем программу, я думаю разбить программу на две части: форма для исполнения программы и сама программа, которая будет вызывать эту форму.

**2.2 Разработка алгоритма**

В программе будет использоваться шаблон регулярных выражений, для определения наличия в строке идентификатора цвета.

Он будет иметь следующий вид: "#[\\dA-Fa-f]{6}[^\\dA-Fa-f]".

**2.3 Разработка пользовательского интерфейса**

Пользователем может являться программист, либо дизайнер, которому нужно определить, есть ли в данном файле с HTML кодом идентификатор цвета и в каких строках.

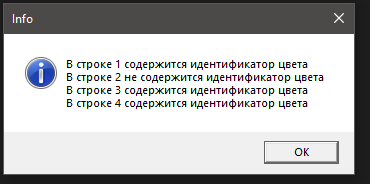
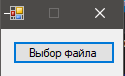
Программа разрабатывается в визуальном виде. В ней будут использоваться элементы, указанные в таблице 1.

Табл.1

*Список используемых элементов управления*

|  |  |  |
| --- | --- | --- |
| Элемент управления | Класс | Описание |
| button1 | Button | Командная кнопка «Выбрать файл». Сразу же после выбора выполняет проверку на наличие в файле идентификатора цвета |
| MessageBox | MessageBox | Окно сообщение, которое показывает, есть в данных строках в файле идентификатор цвета или нет. |
| openFileDialog1 | OpenFileDialog | Позволяет отыскать нужный файл и выбрать его |

Программа, после создания и размещения следующих элементов приняла следующий вид (Рис. 1)



а) б)

Рис. 1 Внешний вид программы, где а) Форма при запуске, б)Окно сообщение

**3. Реализация и тестирование приложения**

***Реализация***

Программа была реализована в Visual Studio Community 2019 версии 16.3.0 в среде разработки C++. На устройстве характеристиками, указанными чуть ниже. Реализация кода программы будет расписана ниже в пункте «**Листинг кода**».

**Описание компьютера:**

Операционная система: Windows 10 (x64)

ОЗУ: 16 ГБ

Процессор: Intel Core i5-2500

Видеокарта: Radeon R7 200 Series

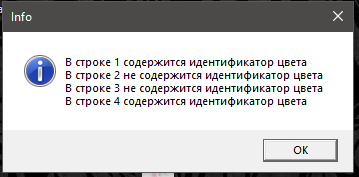
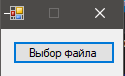
***Тестирование***

**Объект тестирования:** Lab5.exe

**Цель тестирования:** Проверка общей работоспособности программы.

**Метод тестирование:** функциональное тестирование и проход отладчиком для каждой функции по-отдельности и в общем для всей программы.

Тестирование 1 (Рис. 2)

а) б) в)

Рис. 2 Тестирование и проверка ограничения и граничных случаев №1 и нормальных условий, где а) Текст в документе, б) Форма в) Окно сообщение

**Вывод:** Во время выполнения данной работы я научился работать со строками и регулярными выражениями в C#. Данная программа не имеет проверку на то, соответствует выбранный файл нужному формату, так что программу можно усовершенствовать.

**Листинг кода**

**Файл «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;

using System.Text.RegularExpressions;

using System.IO;

namespace WindowsFormsApp2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

string s, buf = "";

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

int flag = 0;

bool fl;

StreamReader f = new StreamReader(openFileDialog1.FileName, Encoding.Default);

Regex pattern = new Regex("#[\\dA-Fa-f]{6}[^\\dA-Fa-f]");

while ((s = f.ReadLine()) != null)

{

fl = false;

s += " ";

++flag;

buf += "В строке " + flag.ToString();

foreach (Match m in pattern.Matches(s))

{

buf += " содержится идентификатор цвета\n";

fl = true;

}

if (!fl) {

buf += " не содержится идентификатор цвета\n";

}

}

f.Close();

}

MessageBox.Show(

buf,

"Info",

MessageBoxButtons.OK,

MessageBoxIcon.Asterisk,

MessageBoxDefaultButton.Button1,

MessageBoxOptions.DefaultDesktopOnly);

}

}

}

**Файл «Program.cs»**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp2

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}