

Казанский (Приволжский) Федеральный университет
Институт Вычислительной математики и информационных
технологий

Отчёт по работе «Частотный анализ текста».

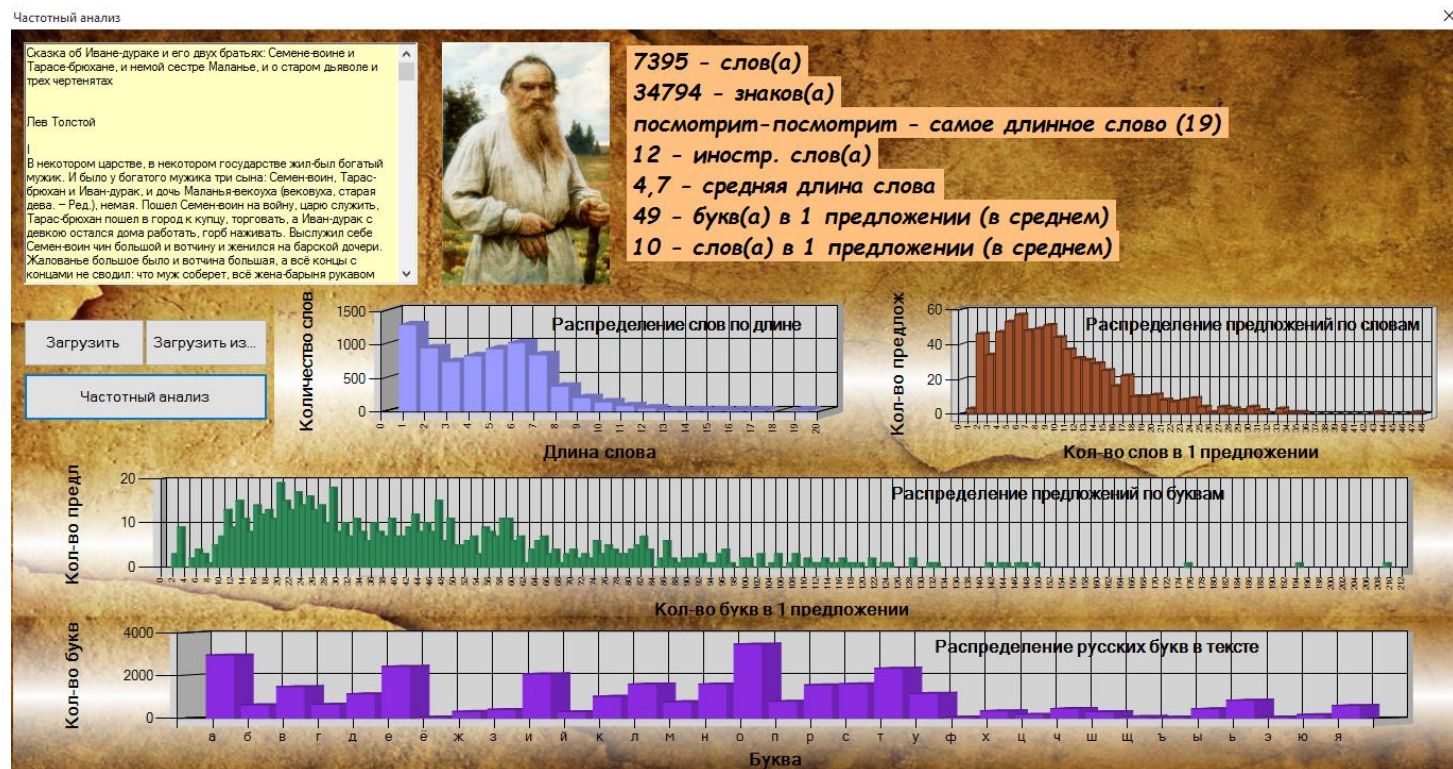
Студент 09-641

Десятов Александр

Казань – 2017

Цель работы: реализовать приложение, которое будет проводить частотные анализы различных текстов.

Скрин:



Код:

```
using System;
using System.Windows.Forms;
using System.IO;
namespace Частотный_анализ
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        //Загрузить
        private void button1_Click(object sender, EventArgs e)
        {
            FileStream file_stream = File.Open("D:\\Users\\Desyatov Alexander\\Документы\\Visual Studio
2012\\Projects\\Частотный анализ\\Рассказы\\Сказка об Иване-дураке и его двух братьях.txt", FileMode.Open,
FileAccess.Read);
            if (file_stream != null)
            {
                StreamReader stream_reader = new StreamReader(file_stream);
                richTextBox1.Text = stream_reader.ReadToEnd();
                file_stream.Close();
            }
        }
    }
}
```

```

    }
    pictureBox1.Visible = true;
}
//Загрузить из...
private void button2_Click(object sender, EventArgs e)
{
    openFileDialog1.Filter = "|*.txt|All files|*.*";
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        string file_name = openFileDialog1.FileName;
        FileStream file_stream = File.Open(file_name, FileMode.Open, FileAccess.Read);
        if (file_stream != null)
        {
            StreamReader stream_reader = new StreamReader(file_stream);
            richTextBox1.Text = stream_reader.ReadToEnd();
            file_stream.Close();
        }
    }
}
//Частотный анализ
private void button3_Click(object sender, EventArgs e)
{
    int length_t = richTextBox1.Text.Length;
    char[] text = richTextBox1.Text.ToCharArray(0, length_t);
    string alfavit_RU = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"; //33
    char[] alf_RU;
    string[] alfavit_RU_str;
    string alfavit_ru = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя";
    char[] alf_ru;
    string[] alfavit_ru_str;
    int al_RU_all = alfavit_RU.Length;
    alf_RU = alfavit_RU.ToCharArray(0, al_RU_all);
    alfavit_RU_str = new string[al_RU_all];
    for (int i = 0; i < al_RU_all; i++) //для график подсчета букв
    {
        alfavit_RU_str[i] = alf_RU[i].ToString();
    }
    int al_ru_all = alfavit_ru.Length;
    alf_ru = alfavit_ru.ToCharArray(0, al_ru_all);
    alfavit_ru_str = new string[al_ru_all];
    for (int i = 0; i < al_ru_all; i++)
    {
        alfavit_ru_str[i] = alf_ru[i].ToString();
    }
    int[] kol_vhozd_RU = new int[al_RU_all];
    int[] kol_vhozd_ru = new int[al_ru_all];
    int kol_slov = 0, max_dlina_slova = 0;
    int kol_in_slov = 0;
    int max_dlina_predl_bukv = 0, dlina_predl_bukv = 0;
    int max_dlina_predl_slov = 0, dlina_predl_slov = 0;
    string samoe_dlinnoe_slovo = "";
    for (int i = 0; i < length_t; i++) //считаем длину слова

```

```

{
    bool in_slovo = false;
    while (i < length_t && (text[i] == ' ' || text[i] == '.' || text[i] == ',' ||
        text[i] == ';' || text[i] == ':' || text[i] == '?' ||
        text[i] == '!' || text[i] == '(' || text[i] == '"' ||
        text[i] == ')) i++;
    int dlina_slova = 0;
    while (i < length_t && text[i] != ' ' && text[i] != '.' &&
        text[i] != ';' && text[i] != ':' && text[i] != '?' &&
        text[i] != '!' && text[i] != '(' &&
        text[i] != '"' && text[i] != ')')
    {
        dlina_slova++;
        if (in_slovo)//проверяем иностранное ли слово
        {
            if (text[i] >= 65 && text[i] <= 90 || text[i] >= 97 && text[i] <= 122)
            {
                in_slovo = true;
            }
        }
        i++;
    }
    dlina_predl_bukv += dlina_slova;
    dlina_predl_slov++;
    if(max_dlina_predl_bukv < dlina_predl_bukv)
        max_dlina_predl_bukv = dlina_predl_bukv;
    if (max_dlina_predl_slov < dlina_predl_slov)
        max_dlina_predl_slov = dlina_predl_slov;
    if (i < length_t && (text[i] == '.' || text[i] == '?' || text[i] == '!' || i==length_t))
    {
        dlina_predl_bukv = 0;//букв в предложении
        dlina_predl_slov = 0;//слов в предложении
    }
    if (dlina_slova > max_dlina_slova)
    {
        max_dlina_slova = dlina_slova;
        самое_dlinnoe_slovo = "";
        i -= max_dlina_slova;
        for (int j = 0; j < max_dlina_slova; j++)
        {
            самое_dlinnoe_slovo += text[i];
            i++;
        }
    }
    if(in_slovo){
        kol_in_slov++;
    }
    if(dlina_slova > 0)
        kol_slov++;
}
label3.Text = самое_dlinnoe_slovo + " - самое длинное слово (" + max_dlina_slova + ")";
label4.Text = kol_in_slov + " - иностр. слов(а)";

```

```

label1.Text = kol_slov.ToString() + " - слов(a)";
int kol_znakov = 0;
for (int i = 0; i < length_t; i++)
{
    if (text[i] != ' ') kol_znakov++;
}
label2.Text = kol_znakov.ToString() + " - знаков(a)";
if (kol_slov != 0)
    label5.Text = (Math.Round( (double) kol_znakov / (double) kol_slov, 1)).ToString() + " - средняя длина
слова";
else
    label5.Text = " - средняя длина слова";
max_dlina_slova++;
int[] kol_slov_s_takoy_dlinoy = new int[max_dlina_slova];
for (int i = 0; i < max_dlina_slova; kol_slov_s_takoy_dlinoy[i] = 0, i++) ;//обнуляем
for (int i = 0; i < length_t; i++)//для графика «кол-во слов с такой длиной»
{
    while (i < length_t && (text[i] == ' ' || text[i] == '.' || text[i] == ',' ||
        text[i] == ';' || text[i] == ':' || text[i] == '?' ||
        text[i] == '!' || text[i] == '(' || text[i] == '"' ||
        text[i] == ')')) i++;
    int dlina_slova = 0;
    while (i < length_t && text[i] != ' ' && text[i] != '.' &&
        text[i] != ';' && text[i] != ':' && text[i] != '?' &&
        text[i] != '!' && text[i] != '(' &&
        text[i] != '"' && text[i] != ')')
    {
        i++;
        dlina_slova++;
    }
    kol_slov_s_takoy_dlinoy[dlina_slova]++;
}
chart1.Series[0].Points.Clear();
for (int i = 1; i < max_dlina_slova; i++)
{
    chart1.Series[0].Points.AddXY(i, kol_slov_s_takoy_dlinoy[i]);
}
max_dlina_predl_bukv++;
int[] kol_predl_s_talim_kol_bukv = new int [max_dlina_predl_bukv];
max_dlina_predl_slov++;
int[] kol_predl_s_talim_kol_slov = new int [max_dlina_predl_slov];
for (int i = 0; i < length_t; i++)
{
    int kol_bukv_v_predl = 0;
    int kol_slov_v_predl = 0;
    while (i < length_t && (text[i] == ' ' || text[i] == '?' || text[i] == '!')) i++;
    while (i < length_t && text[i] != ' ' && text[i] != '?' && text[i] != '!')
    {
        while (i < length_t && text[i] == ' ') i++;
        while (i < length_t && text[i] != ' ' && text[i] != '.' && text[i] != '?' && text[i] != '!')
        {
            if (text[i] >= 'A' && text[i] <= 'Z' || text[i] >= 'a' && text[i] <= 'z' ||

```

```

        text[i] >= 'А' && text[i] <= 'Я' || text[i] >= 'а' && text[i] <= 'я' ||
        text[i] >= '0' && text[i] <= '9')
    {
        kol_bukv_v_predl++;
    }
    i++;
}
kol_slov_v_predl++;
while (i < length_t && text[i] == ' ') i++;
}
kol_predl_s_talim_kol_bukv[kol_bukv_v_predl]++;
kol_predl_s_talim_kol_slov[kol_slov_v_predl]++;
}
chart2.Series[0].Points.Clear();
int kol_predl = 0;
for (int i = 1; i < max_dlina_predl_bukv; i++)
{
    chart2.Series[0].Points.AddXY(i, kol_predl_s_talim_kol_bukv[i]);
    kol_predl += kol_predl_s_talim_kol_bukv[i];
}
chart3.Series[0].Points.Clear();
for (int i = 1; i < max_dlina_predl_slov; i++)
{
    chart3.Series[0].Points.AddXY(i, kol_predl_s_talim_kol_slov[i]);
}
if (kol_znakov > 0)
{
    label6.Text = (Math.Round((double)kol_znakov / (double)kol_predl,0)).ToString() + " - букв(а) в 1
предложении (в среднем)";
    label7.Text = (Math.Round((double)kol_slov / (double)kol_predl,0)).ToString() + " - слов(а) в 1
предложении (в среднем)";
}
else
{
    label6.Text = " - букв(а) в 1 предложении (в среднем)";
    label7.Text = " - слов(а) в 1 предложении (в среднем)";
}
for (int i = 0; i < length_t; i++)//считаем русские буквы
{
    bool nashel = false;
    if (text[i] == ' ' || text[i] == '\n')
    {
        nashel = true;
    }
    for (int j = 0; !nashel && j < al_RU_all; j++)
    {
        if (text[i] == alf_RU[j])
        {
            kol_vhozd_RU[j]++;
            nashel = true;
        }
    }
}

```

```

        for (int j = 0; !nashel && j < al_RU_all; j++)
        {
            if (text[i] == alf_ru[j])
            {
                kol_vhozd_ru[j]++;
                nashel = true;
            }
        }
    }
    chart4.Series[0].Points.Clear();//график длин слов
    for (int i = 0; i < al_ru_all; i++)
    {
        chart4.Series[0].Points.AddXY(alfavit_ru_str[i], kol_vhozd_ru[i] + kol_vhozd_RU[i]);
    }
}
//При загрузке
private void Form1_Load(object sender, EventArgs e)
{
    this.WindowState = System.Windows.Forms.FormWindowState.Maximized;
}
}
}

```