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;

namespace Slovar

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

openFileDialog1.Filter = "Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*";

}

public Dictionary<string, int> dictionary = new Dictionary<string, int>();

void FormTree(Node root, string word)

{

if (root.word == null)

{

dictionary.Add(word, 1);

root.word = word;

root.freq = 1;

root.left = new Node();

root.right = new Node();

}

else

{

if (word.CompareTo(root.word) < 0)

FormTree(root.left, word);

else if (word.CompareTo(root.word) > 0)

FormTree(root.right, word);

else

{

root.freq++;

dictionary[word]++;

}

}

}

void PrintTree(Node root)

{

if (root != null)

{

if (root.left.word != null)

{

PrintTree(root.left);

}

dataGridView1.Rows.Add(root.word, root.freq);

if (root.right.word != null)

{

PrintTree(root.right);

}

}

}

void Search(Node root, String word)

{

textBox2.Clear();

if (root.left.word != null && word.CompareTo(root.word) < 0)

Search(root.left, word);

else if (root.right.word != null && word.CompareTo(root.word) > 0)

Search(root.right, word);

else if (word.CompareTo(root.word) == 0)

textBox2.Text += root.freq;

else

textBox2.Text = "Слово не найдено";

}

void Filtr(Node node, int x)

{

if (node.left.word != null)

{

Filtr(node.left, x);

}

if (node.word.Length == x)

{

dataGridView1.Rows.Add(node.word, node.freq);

}

if (node.right.word != null)

{

Filtr(node.right, x);

}

}

String[] str;

Node node = new Node();

private void Form1\_Load(object sender, EventArgs e)

{

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

return;

// получаем выбранный файл

string filename = openFileDialog1.FileName;

str = System.IO.File.ReadAllText(filename, Encoding.Default).Split(' ', '\n', ',', '.', '"', '\r', '(', ')', '!', '?', '[', ']', '-', '/', '»', '«', ';', ':', '„', '…').ToArray();

for (int i = 0; i < str.Length; i++)

{

if (str[i].CompareTo(" ") > 0)

FormTree(node, str[i]);

}

}

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

{

dataGridView1.AllowUserToAddRows = true;

dataGridView1.Rows.Clear();

PrintTree(node);

dataGridView1.AllowUserToAddRows = false;

}

private void button2\_Click(object sender, EventArgs e)

{

dataGridView1.AllowUserToAddRows = true;

dataGridView1.Rows.Clear();

foreach (var x in dictionary.OrderBy(key => key.Value))//перебирает элементы массивов, векторов или любых других наборов данных. Он присваивает значение текущего элемента переменной итератора, объявленной внутри цикла.

{

dataGridView1.Rows.Add(x.Key, x.Value);

}

dataGridView1.AllowUserToAddRows = false;

}

private void button3\_Click(object sender, EventArgs e)

{

dataGridView1.AllowUserToAddRows = true;

dataGridView1.Rows.Clear();

foreach (var x in dictionary.OrderByDescending(key => key.Value))

{

dataGridView1.Rows.Add(x.Key, x.Value);

}

dataGridView1.AllowUserToAddRows = false;

}

private void button4\_Click(object sender, EventArgs e)

{

dataGridView1.AllowUserToAddRows = true;

dataGridView1.Rows.Clear();

Filtr(node, Convert.ToInt32(numericUpDown1.Value));

dataGridView1.AllowUserToAddRows = false;

}

private void button5\_Click(object sender, EventArgs e)

{

Search(node, textBox1.Text);

}

}

}