# **Отчёт по лабораторной работе №7**

## **Предмет: Microsoft .NET**

## **Выполнил: ст. гр. О-18-ПРИ-рпс-Б, Лядов. В.С.**

## Постановка задач

1. Разработать Windows Forms приложение для просмотра и сохранения файлов в разных кодировках. Программа должна позволять при отображении содержимого файла выбирать и менять кодировку, а также сохранять файл в выбранной кодировке.
2. Разработать программу, позволяющую выделять из файла информацию с использованием регулярных выражений в соответствии с вариантом (табл. 5).

## Практическая часть

Разработать Windows Forms приложение для просмотра и сохранения файлов в разных кодировках. Программа должна позволять при отображении содержимого файла выбирать и менять кодировку, а также сохранять файл в выбранной кодировке:

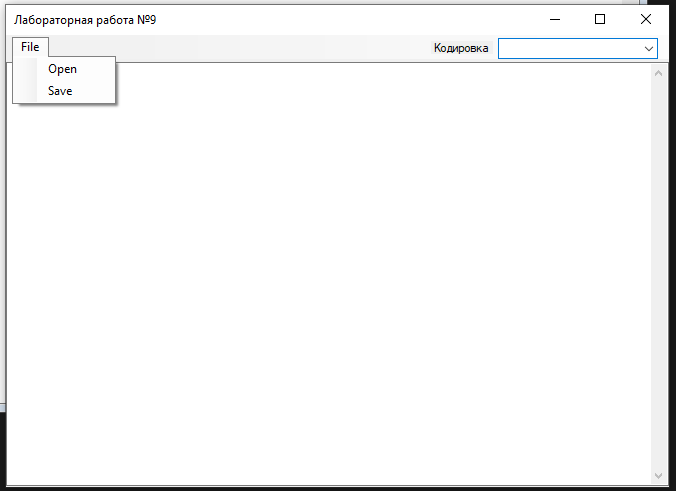
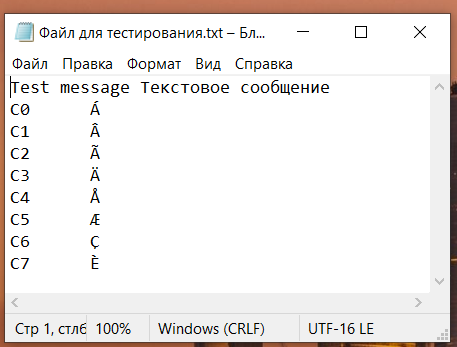


Рис 1. Внешний вид программы



*Рис. 2. Создадим файл через стандартный текстовый редактор и укажем кодировку UTF-16 LE. Добавим латинских символов, кириллицы и символов в кодировке IBM Латиница 1(она же ISO/IEC 8859-1)*

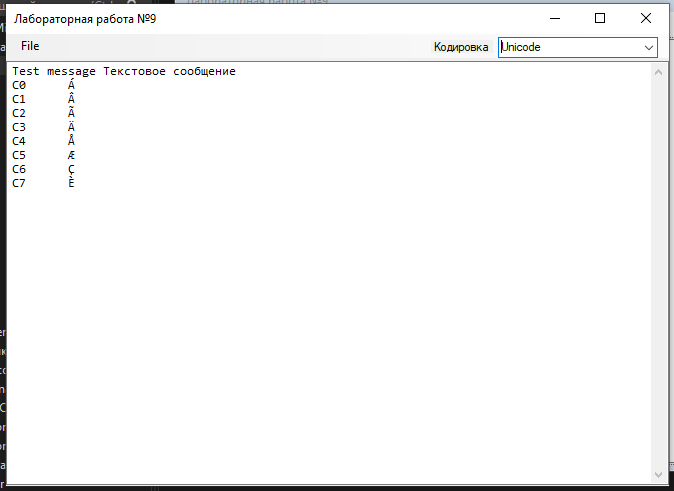


Рис. 3. Откроем файл через приложениe.

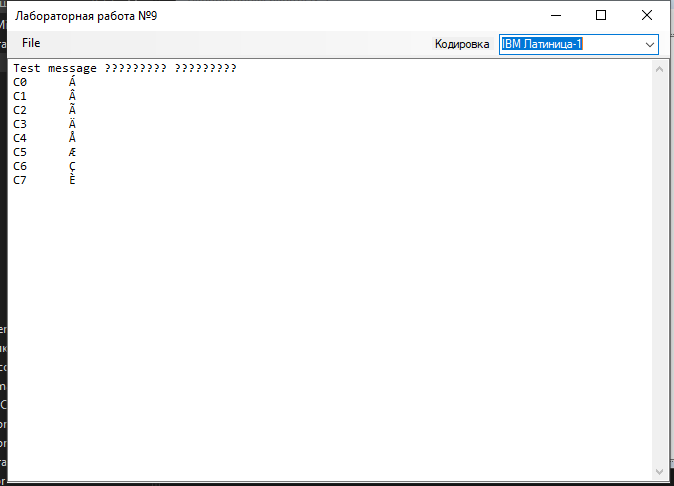


Рис. 4. Поставим кодировку IBM Латиница-1.

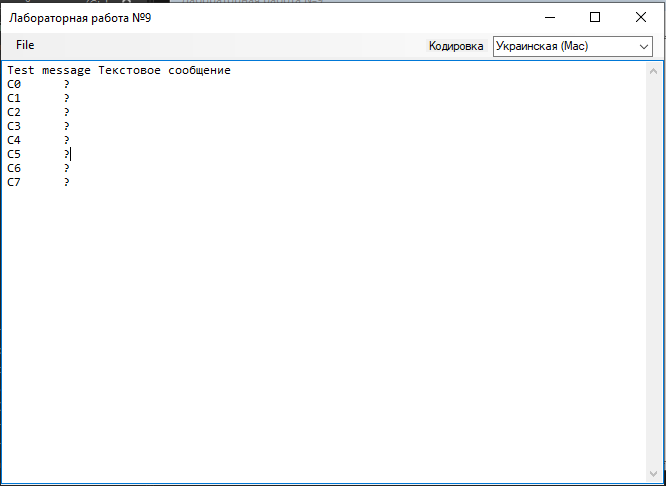


Рис. 5. Поставим кодировку «Украинская»

Разработать программу, позволяющую выделять из файла информацию с использованием регулярных выражений в соответствии с вариантом:

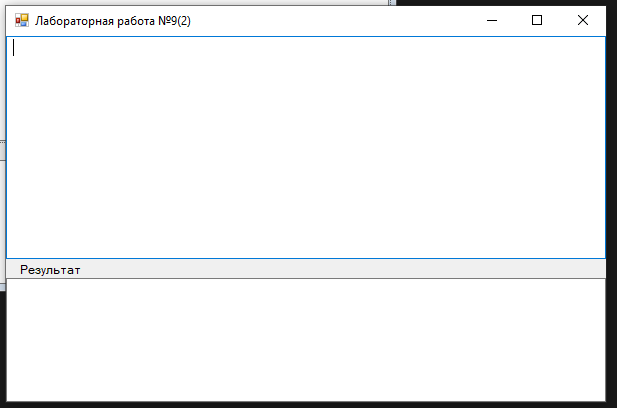
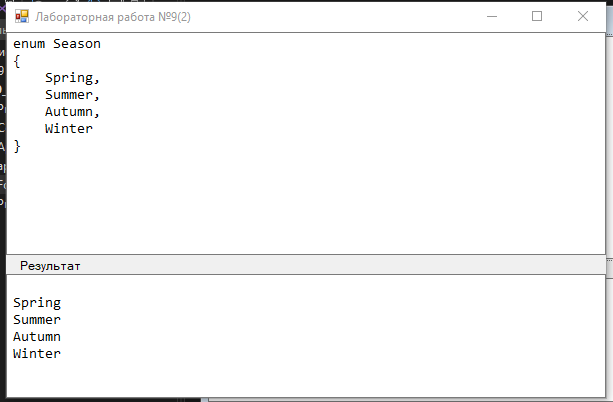


Рис. 6. Внешний вид задания №2.

Задание по варианту:

**Варианты для всех перечислений (enum) на языке C#**



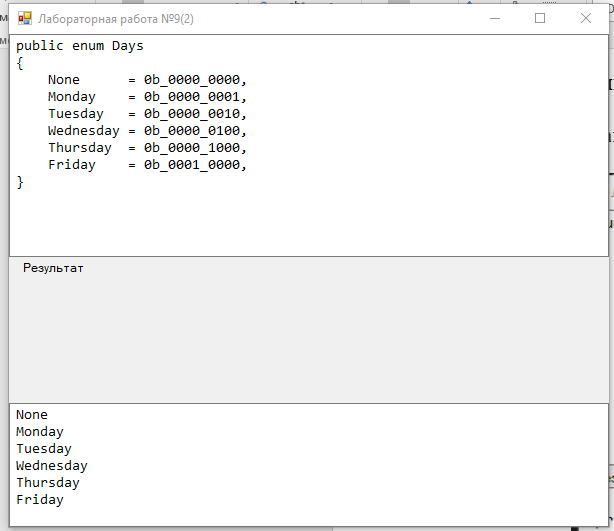


Рис. 7-8. Результат работы программы.

Код

Листинг 1

Исходный код задания №1:

using System;

using System.IO;

using System.Text;

using System.Windows.Forms;

namespace Lab9

{

public partial class FormCodingEditor : Form

{

private string text, original;

private string path;

private Encoding currEncoding, originalEn;

public FormCodingEditor()

{

InitializeComponent();

encodingCombo.Items.Clear();

encodingCombo.DisplayMember = "EncodingName";

foreach (EncodingInfo ei in Encoding.GetEncodings())

{

Encoding e = ei.GetEncoding();

encodingCombo.Items.Add(e);

}

}

private void save()

{

File.WriteAllText(path, text, currEncoding);

}

private void open()

{

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

{

path = openFile.FileName;

text = original = File.ReadAllText(path);

textBox.Text = text;

currEncoding = originalEn = GetEncoding(path);

encodingCombo.SelectedItem = currEncoding;

}

}

private void changeEncoding()

{

var en = (Encoding) encodingCombo.SelectedItem;

byte[] originalBytes = originalEn.GetBytes(original);

byte[] convertedBytes = Encoding.Convert(originalEn, en, originalBytes);

char[] convertedChars = new char[en.GetCharCount(convertedBytes, 0, convertedBytes.Length)];

en.GetChars(convertedBytes, 0, convertedBytes.Length, convertedChars, 0);

text = new string(convertedChars);

textBox.Text = text;

originalEn = en;

}

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

{

open();

}

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

{

save();

}

public static Encoding GetEncoding(string filename)

{

var bom = new byte[4];

using (var file = new FileStream(filename, FileMode.Open, FileAccess.Read))

{

file.Read(bom, 0, 4);

}

if (bom[0] == 0x2b && bom[1] == 0x2f && bom[2] == 0x76) return Encoding.UTF7;

if (bom[0] == 0xef && bom[1] == 0xbb && bom[2] == 0xbf) return Encoding.UTF8;

if (bom[0] == 0xff && bom[1] == 0xfe && bom[2] == 0 && bom[3] == 0) return Encoding.UTF32;

if (bom[0] == 0xff && bom[1] == 0xfe) return Encoding.Unicode;

if (bom[0] == 0xfe && bom[1] == 0xff) return Encoding.BigEndianUnicode;

if (bom[0] == 0 && bom[1] == 0 && bom[2] == 0xfe && bom[3] == 0xff) return new UTF32Encoding(true, true);

return Encoding.ASCII;

}

private void encodingCombo\_SelectedIndexChanged(object sender, EventArgs e)

{

changeEncoding();

}

}

}

Листинг 2

Листинг 3

Ссылка на GitHub на проект: