

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Криворізький національний університет
Кафедра моделювання та програмного забезпечення

Звіт
з лабораторної роботи №2
УПРАВЛІННЯ ТЕКСТОМ У ВІЗУАЛЬНИХ ДОДАТКАХ З
ВИКОРИСТАННЯМ СТАНДАРТНИХ ДІАЛОГІВ

Студент групи ПЗ-21-2

Губарєв Р.В.

+380980190289

Викладачі

Козиков А. В.

Гриценко А. М.

Кривий Ріг

2022

1. Короткі теоретичні відомості про візуальні об'єкти (компонентів), що використовуються.

RichTextBox - Дозволяє відображати текст із форматуванням у простому текстовому форматі або форматі RTF.

GroupBox - Групує набір елементів керування (наприклад, перемикачів) на кадрі з позначкою та без підтримки.

ComboBox - Відображає список елементів, що розкривається.

Label - Відображає недоступний для безпосередньої зміни користувачем текст.

MenuStrip - Створює меню, що настраюється.

ListBox - Відображає список текстових та графічних елементів (значків).

2. Короткі теоретичні відомості про управління файлами та багаторядковим текстом у середовищі розробки

Управління файлами:

- Файл є носієм для збереження даних на комп'ютері. В середовищі розробки файли використовуються для збереження вихідного коду програм, ресурсів, конфігураційних файлів та інших даних.
- Для управління файлами в середовищі розробки використовуються операції, такі як створення нового файлу, відкриття існуючого файлу, збереження змін у файлі, переміщення, перейменування та видалення файлів.
- У багатьох середовищах розробки також надаються можливості для автоматичного відстеження змін у файлах (наприклад, системи контролю версій) і спільної роботи над файлами в команді.

Багаторядковий текст:

- Багаторядковий текст (або многостроковий текст) є послідовністю символів, яка охоплює кілька рядків. В середовищі розробки багаторядковий текст використовується для збереження та редагування вихідного коду програм, коментарів, документації та іншої інформації.
- Багаторядковий текст може бути відформатованим з використанням відступів, вирівнювання, кольорів, шрифтів тощо, щоб поліпшити читабельність і структуру коду або тексту.
- В середовищі розробки зазвичай надаються різні функції для роботи з багаторядковим текстом, такі як відступи, перехід на

наступний/попередній рядок, вирівнювання, вибір блоку тексту та інші.

3. Вихідний текст програми

Form1.cs

```
using System.IO;
using System.Windows.Forms;
using static System.Net.Mime.MediaTypeNames;

namespace Lab2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            saveFileDialog1.Filter = "Text File(*.txt)|*.txt";

            string[] fontSizeValues = { "10", "12", "14", "16", "18", "20",
"22", "24", "28" };
            comboBoxFontSize.Items.AddRange(fontSizeValues);

            string[] fontStyles = { "Bold", "Italic", "Underline" };
            comboBoxFontStyle.Items.AddRange(fontStyles);
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void показатиКількістьСлівУТекстіToolStripMenuItem_Click(object sender, EventArgs e)
        {
            // Отримати текст з RichTextBox
            string text = richTextBox1.Text;

            // Підрахувати кількість слів
            int wordCount = GetWordCount(text);

            textBox1.Text = "" + wordCount;
        }

        private void зберегтиФайлToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (saveFileDialog1.ShowDialog() == DialogResult.Cancel)
                return;
            string filename = saveFileDialog1.FileName;
            File.WriteAllText(filename, richTextBox1.Text);
            MessageBox.Show("Файл збережено!");
        }

        private void відкритиФайлToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (openFileDialog1.ShowDialog() == DialogResult.Cancel)
                return;
            string filename = openFileDialog1.FileName;
            string fileText = File.ReadAllText(filename);
            richTextBox1.Text = fileText;
            MessageBox.Show("Файл відкрито!");
        }

        private void завершитиРоботуToolStripMenuItem_Click(object sender, EventArgs e)
        {
        }
    }
}
```

```

    {
        Close();
    }

    private void змінитиКолірТекстуToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        colorDialog1.ShowDialog();
        richTextBox1.SelectionColor = colorDialog1.Color;
    }

    private void richTextBox1_TextChanged(object sender, EventArgs e)
    {
        if (comboBoxFontSize.SelectedItem != null)
        {
            int fontSize =
int.Parse(comboBoxFontSize.SelectedItem.ToString());
            Font currentFont = richTextBox1.SelectionFont;
            FontStyle fontStyle = currentFont.Style;
            richTextBox1.SelectionFont = new Font(currentFont.FontFamily,
fontSize, fontStyle);
        }
    }

    private void comboBoxFontSize_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (comboBoxFontSize.SelectedItem != null)
        {
            int fontSize =
int.Parse(comboBoxFontSize.SelectedItem.ToString());
            richTextBox1.SelectionFont = new
Font(richTextBox1.Font.FontFamily, fontSize);
        }
    }

    private void groupBox1_Enter(object sender, EventArgs e)
    {
    }

    private void comboBoxFontStyle_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (comboBoxFontStyle.SelectedItem != null)
        {
            string selectedStyle =
comboBoxFontStyle.SelectedItem.ToString();

            FontStyle fontStyle = FontStyle.Regular;

            if (selectedStyle == "Bold")
            {
                fontStyle = FontStyle.Bold;
            }
            else if (selectedStyle == "Italic")
            {
                fontStyle = FontStyle.Italic;
            }
            else if (selectedStyle == "Underline")
            {
                fontStyle = FontStyle.Underline;
            }
        }
    }

```

```

        richTextBox1.SelectionFont = new Font(richTextBox1.Font,
fontStyle);
    }
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    // Отримати текст з RichTextBox
    string text = richTextBox1.Text;

    // Підрахувати кількість слів
    int wordCount = GetWordCount(text);

    textBox1.Text = "" + wordCount;
}
private int GetWordCount(string text)
{
    // Розділити текст на окремі слова за допомогою пробілів
    string[] words = text.Split(new char[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);

    // Підрахувати кількість слів
    return words.Length;
}
private int GetLetterCount(string text)
{
    // Видалити пробіли з тексту
    string textWithoutSpaces = text.Replace(" ", string.Empty);

    // Підрахувати кількість літер
    return textWithoutSpaces.Length;
}

private void кількістьЛітерУТекстіToolStripMenuItem_Click(object
sender, EventArgs e)
{
    // Отримати текст з RichTextBox
    string text = richTextBox1.Text;

    // Підрахувати кількість літер
    int letterCount = GetLetterCount(text);

    textBox2.Text = "" + letterCount;
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
    // Отримати текст з RichTextBox
    string text = richTextBox1.Text;

    // Підрахувати кількість літер
    int letterCount = GetLetterCount(text);

    textBox1.Text = "" + letterCount;
}

private void скасуватиФорматуванняToolStripMenuItem_Click(object
sender, EventArgs e)
{
    if (comboBoxFontSize.SelectedItem != null)
    {
        comboBoxFontSize.Text = "10";
        int fontSize = 10;
        richTextBox1.SelectionFont = new
Font(richTextBox1.Font.FontFamily, fontSize);
    }
}

```

```

        // Встановлюємо колір тексту за замовчуванням
richTextBox1.SelectionColor = SystemColors.ControlText;
    }
}

private void FindWords(string letter)
{
    // Отримати текст з поля richTextBox
string memoText = richTextBox1.Text;

    // Розділити текст на окремі слова
string[] words = memoText.Split(new char[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);

    // Очистити список перед заповненням
listBox1.Items.Clear();

    // Пройти по кожному слову і перевірити, чи містить воно введену
літеру
foreach (string word in words)
{
    if (word.Contains(letter))
    {
        // Додати слово в список
listBox1.Items.Add(word);
    }
}

private void знайтиСловаToolStripMenuItem_Click(object sender,
EventArgs e)
{
    string letter = editTextBox.Text;
    FindWords(letter);
}

private void проПрограмуToolStripMenuItem_Click(object sender,
EventArgs e)
{
    AboutBox1 aboutBox = new AboutBox1();
    aboutBox.Show();
}

private void
перекластиТекстУВерхнійРегістрToolStripMenuItem_Click(object sender,
EventArgs e)
{
    // Отримати текст з поля richTextBox
string memoText = richTextBox1.Text;

    // Перевести текст в верхній регістр
string upperCaseText = memoText.ToUpper();

    // Встановити перетворений текст назад у поле richTextBox
richTextBox1.Text = upperCaseText;
}
}
}

```

AboutBox1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Linq;
using System.Reflection;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Lab2
{
    partial class AboutBox1 : Form
    {
        public AboutBox1()
        {
            InitializeComponent();
            this.Text = "Про програму";
            this.labelProductName.Text = "Текстовый редактор (дуже спрощений)";
            this.labelVersion.Text = String.Format("Version {0}", AssemblyVersion);
            this.labelCopyright.Text = AssemblyCopyright;
            this.labelCompanyName.Text = "Rostik Hubariev";
        }

        #region Assembly Attribute Accessors

        public string AssemblyTitle
        {
            get
            {
                object[] attributes = Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyTitleAttribute), false);
                if (attributes.Length > 0)
                {
                    AssemblyTitleAttribute titleAttribute = (AssemblyTitleAttribute)attributes[0];
                    if (titleAttribute.Title != "")
                    {
                        return titleAttribute.Title;
                    }
                }
                return System.IO.Path.GetFileNameWithoutExtension(Assembly.GetExecutingAssembly().CodeBase);
            }
        }

        public string AssemblyVersion
        {
            get
            {
                return Assembly.GetExecutingAssembly().GetName().Version.ToString();
            }
        }

        public string AssemblyDescription
        {
            get
            {

```



```

        object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyDescription
Attribute), false);
        if (attributes.Length == 0)
        {
            return "";
        }
        return
((AssemblyDescriptionAttribute)attributes[0]).Description;
    }

    public string AssemblyProduct
    {
        get
        {
            object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyProductAttr
ibute), false);
            if (attributes.Length == 0)
            {
                return "";
            }
            return ((AssemblyProductAttribute)attributes[0]).Product;
        }
    }

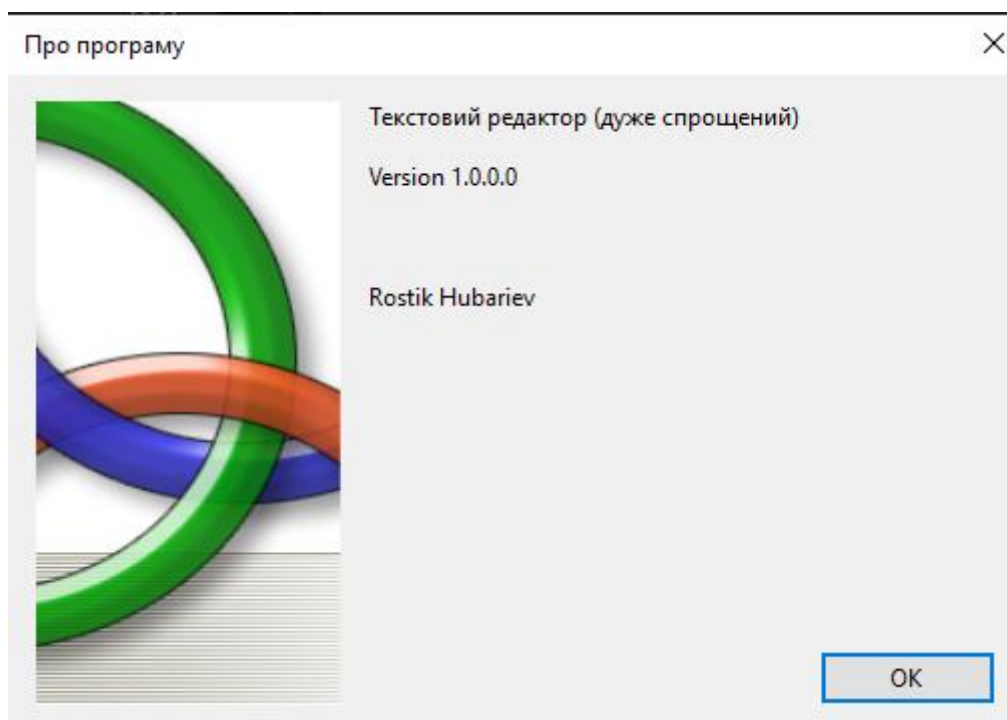
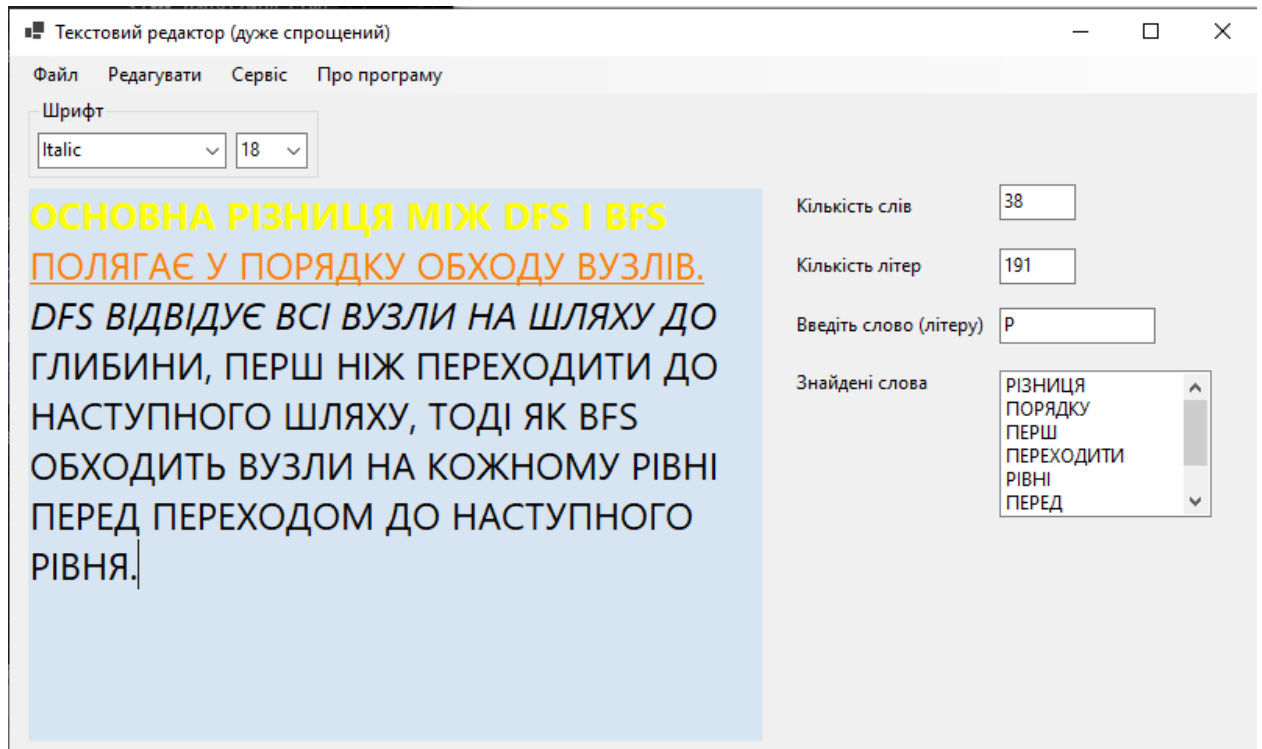
    public string AssemblyCopyright
    {
        get
        {
            object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyCopyrightAt
tribute), false);
            if (attributes.Length == 0)
            {
                return "";
            }
            return ((AssemblyCopyrightAttribute)attributes[0]).Copyright;
        }
    }

    public string AssemblyCompany
    {
        get
        {
            object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyCompanyAttr
ibute), false);
            if (attributes.Length == 0)
            {
                return "";
            }
            return ((AssemblyCompanyAttribute)attributes[0]).Company;
        }
    }
#endregion

    private void okButton_Click(object sender, EventArgs e)
    {
        Close();
    }
}

```

4. Знімок інтерфейсу програми з результатами роботи



5. Короткі висновки

В цій лабораторній роботі я навчився працювати з файлами, зберігати та завантажувати їх, а також редагувати текст в текстовому полі.

6. Список літератури

- <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/controls/windows-forms-controls-by-function?view=netframeworkdesktop-4.8>