

Міністерство освіти України
Національний технічний університет "ХПІ"
кафедра "Інформатики та інтелектуальної власності"

Звіт

Лабораторна робота 11

з дисципліни « C# .Net »

Виконав: студент групи КН-921Б

Бірюков Д .Є.

Перевірив:

Івашко А.В.

Харків 2023

Зміст

Завдання 1(багатовіковий застосунок).....	1
Завдання 2(Трикутник).....	3
Завдання 4(Вираз).....	7
Завдання 5(Тур містами).....	12
Завдання 6(ColorDialog).....	13
Завдання 7(OpenFileDialog).....	16
Завдання 8(текстовий редактор).....	19
Висновок.....	21

Завдання 1

Використовуючи граф-схему власного варіанту, створити багатовіконий застосунок виклику відповідних форм

Назви форм повинні частково відповідати назвам з вершин графів (наприклад Form1, Form2, . . . чи FormA, FormB, FormC,). Форми мають викликатися в модальному режимі (ShowDialog) та мати різнокольорові характеристики фону та тексту. Виклики форм виконувати за допомогою звичайних кнопок, кнопок з панелей ToolStrip, або за допомогою команд з MenuStrip чи ContextMenuStrip.

№ варіанту	Граф-схема
2.	<pre> graph TD 1((1)) --> 2((2)) 2((2)) --> 3((3)) 2((2)) --> 4((4)) 4((4)) --> 5((5)) 4((4)) --> 6((6)) 6((6)) --> 7((7)) </pre>

```

using System;
using System.Drawing;
using System.Windows.Forms;

```

```

namespace MultiFormApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void form1ToolStripMenuItem_Click(object sender, EventArgs e)
        {
            using (Form2 form2 = new Form2())
            {

```

```
        form2.ShowDialog();
    }
}

private void form2ToolStripMenuItem_Click(object sender, EventArgs e)
{
    using (Form3 form3 = new Form3())
    {
        form3.BackColor = Color.Blue;
        form3.ForeColor = Color.White;
        form3.ShowDialog();
    }
}

private void form3ToolStripMenuItem_Click(object sender, EventArgs e)
{
    using (Form4 form4 = new Form4())
    {
        form4.BackColor = Color.Green;
        form4.ForeColor = Color.Black;
        form4.ShowDialog();
    }
}

private void form4ToolStripMenuItem_Click(object sender, EventArgs e)
{
    using (Form5 form5 = new Form5())
    {
        form5.BackColor = Color.Yellow;
        form5.ForeColor = Color.Black;
        form5.ShowDialog();
    }
}

private void form5ToolStripMenuItem_Click(object sender, EventArgs e)
{

```

```

using (Form6 form6 = new Form6())
{
    form6.BackColor = Color.Orange;
    form6.ForeColor = Color.Black;
    form6.ShowDialog();
}
}

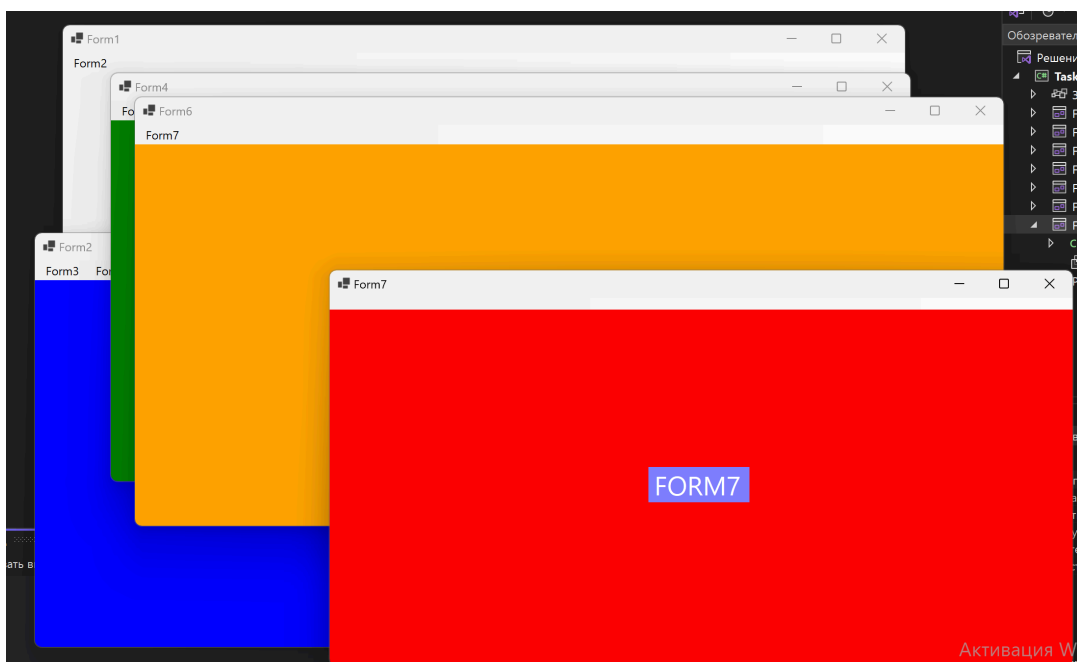
```

```

private void form6ToolStripMenuItem_Click(object sender, EventArgs e)
{
    using (Form7 form7 = new Form7())
    {
        form7.BackColor = Color.Red;
        form7.ForeColor = Color.White;
        form7.ShowDialog();
    }
}
}
}

```

Результат



Завдання 3

За завданням власного варіант

Створити чотири варіанти застосунку (або в окремих проектах, або в одному проєкті на окремих формах, які викликаються з якоїсь головної форми за допомогою кнопок чи команд меню) для вирішення задачі:

- вхідні дані і результати розрахунків відображаються на одній формі;
- вхідні дані отримуються з окремої форми;
- результати розрахунків виводяться в окрему форму;
- вхідні дані та результати розрахунків відображаються на окремих формах.

№ варіанту	Задача
2.	Дани сторони прямокутника, розрахувати його периметр та площу

```
namespace Task_03_1_
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // Проверка наличия введенных данных
            if (string.IsNullOrEmpty(textBox1.Text) ||
                string.IsNullOrEmpty(textBox2.Text) || string.IsNullOrEmpty(textBox3.Text))
            {
                MessageBox.Show("Пожалуйста, заполните все поля.", "Ошибка",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }
        }
    }
}
```

```

    }

    // Попытка парсинга введенных данных
    if (!double.TryParse(textBox1.Text, out double side1) || !
double.TryParse(textBox2.Text, out double side2) || !double.TryParse(textBox3.Text,
out double side3))
    {
        MessageBox.Show("Пожалуйста, введите корректные числовые
значения.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    // Проверка на возможность построения треугольника
    if (!IsValidTriangle(side1, side2, side3))
    {
        MessageBox.Show("Треугольник с такими сторонами не существует.",
"Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    // Расчет периметра
    double perimeter = side1 + side2 + side3;

    // Расчет площади по формуле Герона
    double s = perimeter / 2;
    double area = Math.Sqrt(s * (s - side1) * (s - side2) * (s - side3));

    // Вывод результатов
    label4.Text = $"Периметр: {perimeter}";
    label5.Text = $"Площадь: {area}";
}

// Проверка на возможность построения треугольника
private bool IsValidTriangle(double a, double b, double c)
{
    return a + b > c && a + c > b && b + c > a;
}

```

}

}

Результат:

1 сторона трикутника

2 сторона трикутника

3 сторона трикутника

Периметр: 210

Площадь: 499,3746088859545

Завдання 4

Створити чотири варіанти застосунку (або в окремих проектах, або в одному проекті на окремих формах, які викликаються з якоїсь головної форми за допомогою кнопок чи команд меню) для розрахунку виразу:

- вхідні дані і результати розрахунків виразу відображаються на одній формі;
- вхідні дані отримуються з окремої форми;
- результати розрахунків виводяться в окрему форму;
- вхідні дані та результати розрахунків відображаються на окремих формах.

2.	$\prod_{i=1}^y \sqrt{\frac{i+z}{x-y+i^2}}$
----	--


```

namespace Task_03_1_
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // Проверка наличия введенных данных
            if (string.IsNullOrEmpty(textBox1.Text) ||
string.IsNullOrEmpty(textBox2.Text) || string.IsNullOrEmpty(textBox3.Text))
            {
                MessageBox.Show("Пожалуйста, заполните все поля.", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }

            // Попытка парсинга введенных данных
            if (!double.TryParse(textBox1.Text, out double x) || !
double.TryParse(textBox2.Text, out double y) || !double.TryParse(textBox3.Text, out
double z))
            {
                MessageBox.Show("Пожалуйста, введите корректные числовые
значения.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }

            // Расчет выражения
            double result = 1.0;

            for (int i = 1; i <= y; i++)
            {
                double numerator = i + z;
                double denominator = x - y + Math.Pow(i, 2);

```

```

        if (denominator == 0)
        {
            MessageBox.Show("Ошибка: Знаменатель равен нулю.", "Ошибка",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }

        result *= Math.Sqrt(numerator / denominator);
    }

    // Вывод результата
    label4.Text = $"Результат выражения: {result}";
}
}
}

```

The screenshot shows a web application interface. On the left, there are three input fields labeled 'x', 'y', and 'z'. The 'x' field contains the value '1', the 'y' field contains '1', and the 'z' field contains '21'. To the right of these fields is a large mathematical expression:
$$\prod_{i=1}^y \sqrt{\frac{i+z}{x-y+i^2}}$$
. Below the expression is a button labeled 'Розрахувати'. At the bottom of the interface, the result of the calculation is displayed: 'Результат выражения: 4,69041575982343'.

Завдання 5

Створити застосунок, який на головній формі дозволяв би вибирати місто країни згідно до свого варіанту та відобразити фото та короткий опис на додатковій формі

№ варіанту	Країна	Кількість міст
2.	Ізраїль	7

```

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 Task_05
{
    public partial class Form2 : Form
    {
        public string SelectedItem { get; set; }

        public Form2()
        {
            InitializeComponent();

            private void Form2_Load(object sender, EventArgs e)
            {
                if (SelectedItem == "1. Єрусалим")
                {
                    label1.Text = "Єрусалим";
                    pictureBox1.Image = Properties.Resources.Image1; // Замените "Image1"
на имя вашего изображения в ресурсах
                    label2.Text = "Єрусалим є столицею та найбільш священним місцем
для трьох основних світових релігій - юдаїзму, християнства та ісламу. Місто
багате історією та культурою, його старовинний центр включений до списку
Світової спадщини ЮНЕСКО.";
                }
                else if (SelectedItem == "2. Тель-Авів")
                {
                    label1.Text = "Тель-Авів";
                    pictureBox1.Image = Properties.Resources.Image2; // Замените "Image2"
на имя вашего изображения в ресурсах
                    label2.Text = "Тель-Авів вважається економічним і культурним
центром Ізраїлю, відомим своїми сучасними хмарами і великою кількістю

```

пляжів. Місто славиться своєю динамічною атмосферою, модними кафе і нічним життям.";

```
}
```

```
else if (SelectedItem == "3. Хайфа")
```

```
{
```

```
    label1.Text = "Хайфа";
```

pictureBox1.Image = Properties.Resources.Image3; // Замените "Image2" на имя вашего изображения в ресурсах

label2.Text = "Розташована на узбережжі Середземного моря, Хайфа є важливим портом та індустріальним центром. Місто також славиться Багдадським садом, який є одним із найкрасивіших садів у світі.";

```
}
```

```
else if (SelectedItem == "4. Беер-Шева")
```

```
{
```

```
    label1.Text = "Беер-Шева";
```

pictureBox1.Image = Properties.Resources.Image4; // Замените "Image2" на имя вашего изображения в ресурсах

label2.Text = "Розташована в південній частині країни, Беер-Шева відома своєю історією, що налічує тисячоліття. Місто є важливим центром для інновацій та технологічного розвитку.";

```
}
```

```
else if (SelectedItem == "5. Іерихон")
```

```
{
```

```
    label1.Text = "Іерихон";
```

pictureBox1.Image = Properties.Resources.Image5; // Замените "Image2" на имя вашего изображения в ресурсах

label2.Text = "Як одне з найстаріших міст у світі, Іерихон розташований в долині ріки Йордан. Він відомий своєю археологічною спадщиною і красивими оазисами.";

```
}
```

```
else if (SelectedItem == "6. Нетанья")
```

```
{
```

```
    label1.Text = "Нетанья";
```

pictureBox1.Image = Properties.Resources.Image6; // Замените "Image2" на имя вашего изображения в ресурсах

label2.Text = "Розташована на узбережжі Середземного моря, Нетанья є популярним курортним містом, відомим своїми пляжами, кафе та розвагами.";

```
}
```

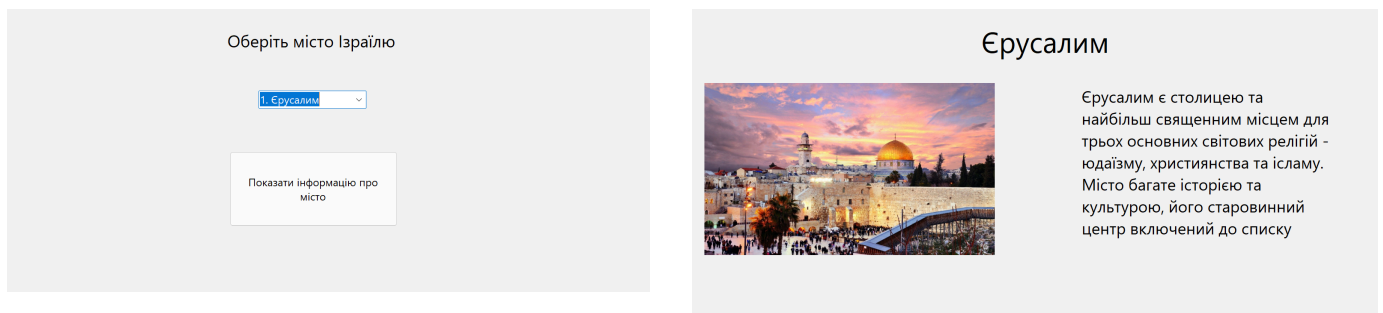
```
else if (SelectedItem == "7. Ашдод")
```

```

{
    label1.Text = "Ашдод";
    pictureBox1.Image = Properties.Resources.Image7; // Замените "Image2"
на имя вашего изображения в ресурсах
    label2.Text = "Ашдод - важливий портовий місто, розташоване на
узбережжі Середземного моря. Місто має історію, яка налічує тисячоліття, і є
важливим торговим та промисловим центром.";
}
}
}
}

```

Результат



Завдання 6

До форми застосунку зі четвертого завдання четвертої роботи додати кнопку із назвою, наприклад, Конфігурація або Параметри, яка б викликала окрему форму для налаштування кольорово-розмірні параметри (не менше п'яти) компонентів основної форми. Для налаштування рекомендується використовувати компоненти діалогів: `ColorDialog` та `FontDialog`

```

using System;
using System.Drawing;
using System.Windows.Forms;

```

```

namespace Task_06
{
    public partial class Form2 : Form
    {
        private Form1 mainForm;
        private FontDialog fontDialog1 = new FontDialog();
    }
}

```

```

public Form2(Form1 form)
{
    InitializeComponent();
    mainForm = form;
}

public Font GetSelectedFont()
{
    return fontDialog1.Font;
}

private void button1_Click(object sender, EventArgs e)
{
    // Діалогове вікно для вибору кольору
    ColorDialog colorDialog1 = new ColorDialog();
    if (colorDialog1.ShowDialog() == DialogResult.OK)
    {
        mainForm.BackColor = colorDialog1.Color;
    }
}

private void button2_Click(object sender, EventArgs e)
{
    // Діалогове вікно для вибору шрифту
    if (fontDialog1.ShowDialog() == DialogResult.OK)
    {
        mainForm.Font = fontDialog1.Font;
    }
}
}

```

Завдання 7*

Модифікувати застосунок минулого завдання додавши до конфігураційної форми можливість зберігати та завантажувати конфігураційні параметри компонентів головної форми у(3) зовнішніх (за бажанням текстових або

двійкових) файлах, застосовуючи для цього компоненти OpenFileDialog та SaveFileDialog. Передбачити можливість автоматичного завантаження параметрів компонентів головної форми під час її запуску.

```
using System;
using System.Windows.Forms;
using System.IO;

namespace Task_7
{
    public partial class Form3 : Form
    {
        private Button buttonClose;

        private void InitializeComponent()
        {
            this.buttonClose = new System.Windows.Forms.Button();
            this.label1 = new System.Windows.Forms.Label();
            this.SuspendLayout();
            //
            // buttonClose
            //
            this.buttonClose.Location = new System.Drawing.Point(76, 140);
            this.buttonClose.Name = "buttonClose";
            this.buttonClose.Size = new System.Drawing.Size(100, 30);
            this.buttonClose.TabIndex = 1;
            this.buttonClose.Text = "Закрити";
            this.buttonClose.Click += new System.EventHandler(this.buttonClose_Click);
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Location = new System.Drawing.Point(15, 15);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(0, 16);
```

```

        this.label1.TabIndex = 2;
        //
        // Form3
        //
        this.ClientSize = new System.Drawing.Size(282, 253);
        this.Controls.Add(this.label1);
        this.Controls.Add(this.buttonClose);
        this.Name = "Form3";
        this.ResumeLayout(false);
        this.PerformLayout();

    }

    public void ShowTextFileContents(string filePath)
    {
        if (File.Exists(filePath))
        {
            string fileContents = File.ReadAllText(filePath);
            label1.Text = fileContents;
        }
    }

    private void buttonClose_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private Label label1;
}
}

```

Результат

The image shows a Windows application window titled "Анкета відвідувача ресторана:". The main window has a red header and a grey body. It contains three sections: "Контактна інформація" with fields for "Ім'я", "Телефон", "Пошта", and "Дата відвідування"; "Персональна інформація" with a "Вік" dropdown, "Улюблена кухня" dropdown, and a text area for "Які страви ви б хотіли бачити в меню?"; and "Оцінка нашої установи" with two groups of radio buttons for feedback. At the bottom are buttons "Відправити", "Очистити", and "Конфігурація". A "Конфігурація" dialog box is open, showing options to "Змінити колір", "Завантажити файл", "Змінити шрифт", and "Зберегти файл".

Завдання 8

Розробити застосунок текстовий редактор та забезпечити йому наступні функції:

- створення нового файлу, збереження та завантаження файлу;
- виведення поточної позиції курсора у документі;
- вікно «Про програму» з ім'ям розробника програми.

№ варіанту	Додаткова функція
2.	Забезпечити можливість вибору кольору шрифту

```
using System;
using System.IO;
using System.Windows.Forms;
using System.Drawing;
```

```
namespace Task_8
{
```

```

public partial class Form1 : Form
{
    private TextBox textBox;
    private string currentFilePath;

    public Form1()
    {
        InitializeComponent();
        InitializeUI();
    }

    private void InitializeUI()
    {
        // Створення основного вікна
        this.Text = "Text Editor";
        this.Size = new Size(800, 600);

        // Створення текстового поля
        textBox = new TextBox();
        textBox.Dock = DockStyle.Fill;
        textBox.Multiline = true;
        textBox.ScrollBars = ScrollBars.Both;
        this.Controls.Add(textBox);

        // Створення меню
        MenuStrip menuStrip = new MenuStrip();
        ToolStripMenuItem fileMenu = new ToolStripMenuItem("File");
        ToolStripMenuItem editMenu = new ToolStripMenuItem("Edit");
        ToolStripMenuItem helpMenu = new ToolStripMenuItem("Help");

        // Додаємо елементи меню "File"
        ToolStripMenuItem newFileMenuItem = new ToolStripMenuItem("New");
        newFileMenuItem.Click += NewFileMenuItem_Click;
        fileMenu.DropDownItems.Add(newFileMenuItem);

        ToolStripMenuItem openFileMenuItem = new ToolStripMenuItem("Open");

```

```
openFileMenuItem.Click += OpenFileMenuItem_Click;
fileMenu.DropDownItems.Add(openFileMenuItem);
```

```
ToolStripMenuItem saveFileMenuItem = new ToolStripMenuItem("Save");
saveFileMenuItem.Click += SaveFileMenuItem_Click;
fileMenu.DropDownItems.Add(saveFileMenuItem);
```

```
// Додаємо елементи меню "Edit"
```

```
ToolStripMenuItem fontColorMenuItem = new ToolStripMenuItem("Font
Color");
fontColorMenuItem.Click += FontColorMenuItem_Click;
editMenu.DropDownItems.Add(fontColorMenuItem);
```

```
ToolStripMenuItem currentPositionMenuItem = new
ToolStripMenuItem("Current Position");
currentPositionMenuItem.Click += CurrentPositionMenuItem_Click;
editMenu.DropDownItems.Add(currentPositionMenuItem);
```

```
// Додаємо елементи меню "Help"
```

```
ToolStripMenuItem aboutMenuItem = new ToolStripMenuItem("About");
aboutMenuItem.Click += AboutMenuItem_Click;
helpMenu.DropDownItems.Add(aboutMenuItem);
```

```
// Додаємо всі меню до основного меню
```

```
menuStrip.Items.Add(fileMenu);
menuStrip.Items.Add(editMenu);
menuStrip.Items.Add(helpMenu);
```

```
// Додаємо меню до форми
```

```
this.Controls.Add(menuStrip);
```

```
}
```

```
private void NewFileMenuItem_Click(object sender, EventArgs e)
```

```
{
```

```
    textBox.Clear();
```

```
    currentFilePath = null;
```

```
}
```

```

private void OpenFileMenuItem_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Filter = "Text Files (*.txt)|*.txt|All Files (*.*)|*.*";
    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        currentFilePath = openFileDialog.FileName;
        textBox.Text = File.ReadAllText(currentFilePath);
    }
}

```

```

private void SaveFileMenuItem_Click(object sender, EventArgs e)
{
    if (currentFilePath == null)
    {
        SaveFileDialog saveFileDialog = new SaveFileDialog();
        saveFileDialog.Filter = "Text Files (*.txt)|*.txt|All Files (*.*)|*.*";
        if (saveFileDialog.ShowDialog() == DialogResult.OK)
        {
            currentFilePath = saveFileDialog.FileName;
        }
        else
        {
            return; // Відмінено користувачем
        }
    }
    File.WriteAllText(currentFilePath, textBox.Text);
}

```

```

private void FontColorMenuItem_Click(object sender, EventArgs e)
{
    ColorDialog colorDialog = new ColorDialog();
    if (colorDialog.ShowDialog() == DialogResult.OK)
    {
        textBox.ForeColor = colorDialog.Color;
    }
}

```

```

    }
}

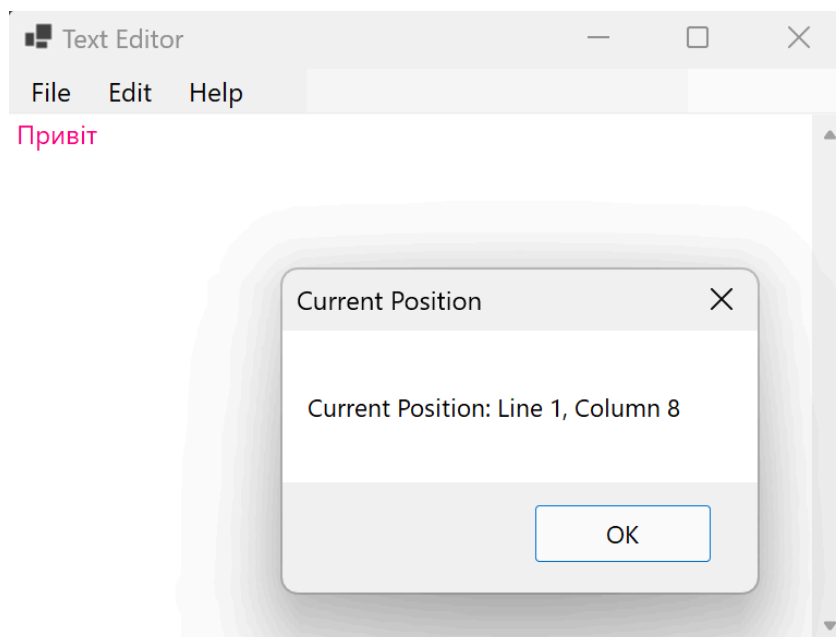
private void CurrentPositionMenuItem_Click(object sender, EventArgs e)
{
    int currentPosition = textBox.SelectionStart;
    int currentLine = textBox.GetLineFromCharIndex(currentPosition);
    int currentColumn = currentPosition -
textBox.GetFirstCharIndexFromLine(currentLine);

    MessageBox.Show($"Current Position: Line {currentLine + 1}, Column
{currentColumn + 1}", "Current Position");
}

private void AboutMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Text Editor\n\nDeveloped by Dmytro Biriukov",
"About");
}
}
}

```

Результат



Висновок:

Лабораторна робота реалізує програму на платформі Windows Forms з використанням С#, що включає в себе дві форми: основну форму для введення та виведення інформації та додаткову форму для налаштування кольорів та шрифтів.

1. Основна форма (Form1):

- Реалізована можливість введення інформації користувачем за допомогою текстових полів, комбо-боксів і радіо-кнопок.
- Здійснена перевірка наявності введених даних та виведення відповідного повідомлення у випадку невалідних даних.
- Використані групи елементів (GroupBox) для організації та групування введених даних.

2. Додаткова форма (Form2):

- Реалізовані можливості вибору кольору та шрифту за допомогою відповідних діалогових вікон.
- Використано властивість `DialogResult``, щоб передавати результат вибору користувача між формами.

3. Обмін даними між формами:

- Використано конструктор додаткової форми для передачі посилання на основну форму.
- Здійснено обмін даними, зокрема шрифтом, між формами для налаштування.

4. Динамічна зміна шрифту:

- Реалізовано функцію для динамічної зміни шрифту для групових елементів та їх вмісту за допомогою рекурсивного перегляду контролів.

Загальною метою лабораторної роботи було розширити функціональність основної програми, дозволяючи користувачам налаштовувати візуальне відображення програми за допомогою додаткової форми. Це включає в себе вибір кольорів і шрифтів, що полегшує персоналізацію та забезпечує зручний інтерфейс для користувачів.