

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

Звіт

Лабораторна робота 7

з дисципліни « C# .Net »

Виконав: студент групи КН-921Б

Бірюков Д .Є.

Перевірив:

Івашко А.В.

Харків 2023

Зміст

Завдання 1 (подія Click).....	1
Завдання 2 (подія Click) параметр object sender	3
Завдання 3(подія DbClick)	6
Завдання 4(подія DbClick) параметр object sender.....	8
Завдання 5(події MouseEnter та MouseLeave).....	10
Завдання 6(події MouseEnter та MouseLeave)параметр object sender.....	12
Завдання 7(події MouseDown та MouseUp).....	14
Завдання 8(події MouseDown та MouseUp) параметр object sender.....	15
Завдання 9(події Enter та Leave).....	17
Завдання 10(події Enter та Leave)параметр object sender.....	20
Завдання 11 (події KeyDown, KeyUp та KeyPress).....	22
Завдання 12 (подія SelectedIndexChanged).....	23
Завдання 13 (HotKeys)	24
Завдання 14 (подія Form.Load)	25
Висновок	28

Завдання 1 (подія Click)

Створити проект із формою, в центрі якої мають знаходитися три кнопки (компонент Button) з довільними написами. Для кожної кнопки зробити перехоплення події Click таким чином, щоб відповідна функція- перехоплювач виводила напис, для якої настає подія клацання, в місце форми згідно до варіанту

№ варіанту	Місце виводу повідомлення із написом на кнопці
2.	<div>Напис</div>

```
using System;
using System.Reflection.Emit;
using System.Windows.Forms;
```

```
namespace Task_01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

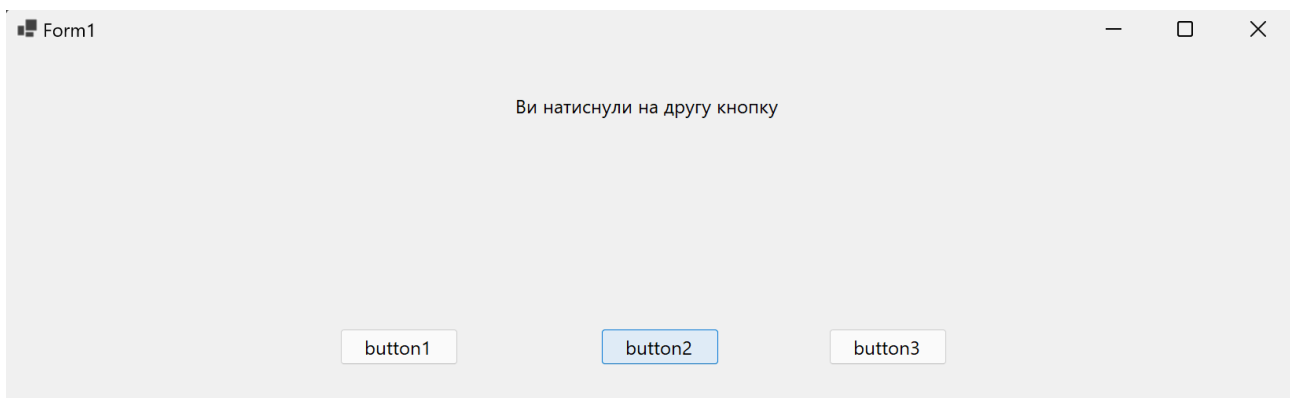
            private void button1_Click(object sender, EventArgs e)
            {
                label1.Text = "Ви натиснули на першу кнопку";
            }

            private void button2_Click(object sender, EventArgs e)
            {
                label1.Text = "Ви натиснули на другу кнопку";
            }
        }
    }
}
```

```
private void button3_Click(object sender, EventArgs e)
{
    label1.Text = "Ви натиснули на третю кнопку";
}

private void MainForm_Load(object sender, EventArgs e)
{
    // Опціонально: налаштування вигляду Label
    label1.TextAlign = ContentAlignment.MiddleCenter;
    label1.Font = new System.Drawing.Font("Arial", 14,
System.Drawing.FontStyle.Bold);
}
}
```

Результат:



Завдання 2*

Виконати минуле завдання з одним обробником подій для усіх кнопок, використовуючи параметр object sender, який приймає обробник

```
using System;
using System.Reflection.Emit;
using System.Windows.Forms;

namespace Task_02
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            InitializeEventHandlers(); // Ініціалізація обробників подій
        }

        private void InitializeEventHandlers()
        {
            // Прив'язка обробників подій до кнопок
            button1.Click += button_Click;
            button2.Click += button_Click;
            button3.Click += button_Click;
        }

        private void button_Click(object sender, EventArgs e)
        {
            Button clickedButton = sender as Button;
            if (clickedButton != null)
            {
                label1.Text = "Ви натиснули на " + clickedButton.Text;
            }
        }

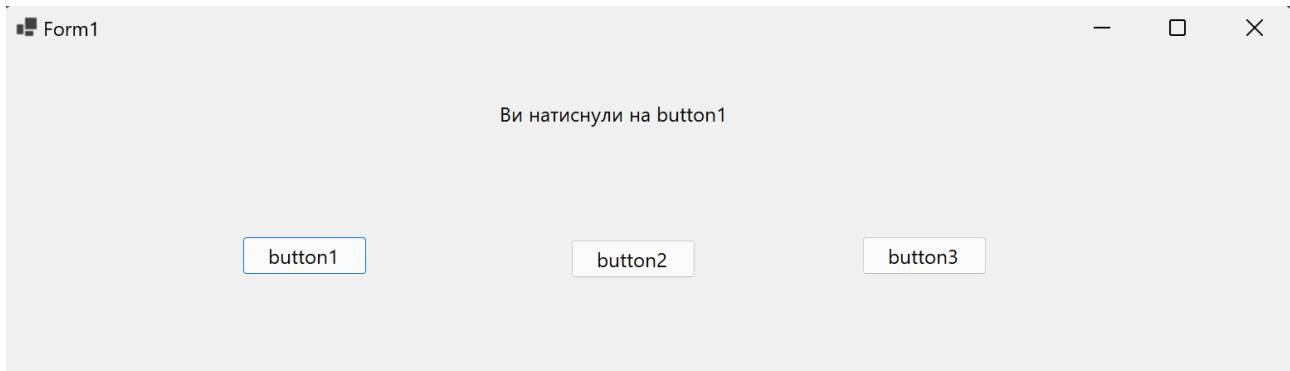
        private void MainForm_Load(object sender, EventArgs e)
        {
            // Опціонально: налаштування вигляду Label
            label1.TextAlign = ContentAlignment.MiddleCenter;
            label1.Font = new System.Drawing.Font("Arial", 14,
System.Drawing.FontStyle.Bold);
        }
    }
}
```

```

    }
}

```

Результат:



Завдання 3 (подія DbClick)

Створити проект зі формою, в центрі якої мають знаходитися три рядки довільного тексту (компонент Label). Для кожного рядка зробити перехоплення події DbClick таким чином, щоб відповідна функція- перехоплювач трансформувала рядок, для якого настає подія подвійного клацання, згідно до варіанту

№ варіанту	Вид трансформації рядка
2.	Розмір шрифту тексту зменшується на 1pt

```

using System;
using System.Windows.Forms;

```

```

namespace Task_03
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void label1_DoubleClick(object sender, EventArgs e)
        {

```

```

Label clickedLabel = sender as Label;
if (clickedLabel != null)
{
    // Отримуємо поточний розмір шрифту
    float currentFontSize = clickedLabel.Font.Size;

    // Зменшуємо розмір шрифту на 1 pt
    float newFontSize = currentFontSize - 1;

    // Мінімальний розмір шрифту, якщо він менший 1 pt
    if (newFontSize < 1)
    {
        newFontSize = 1;
    }

    // Створюємо новий об'єкт шрифту з оновленим розміром
    Font newFont = new Font(clickedLabel.Font.FontFamily, newFontSize);

    // Встановлюємо новий шрифт для Label
    clickedLabel.Font = newFont;
}
}

private void label2_DoubleClick(object sender, EventArgs e)
{
    Label clickedLabel = sender as Label;
    if (clickedLabel != null)
    {
        // Отримуємо поточний розмір шрифту
        float currentFontSize = clickedLabel.Font.Size;

        // Зменшуємо розмір шрифту на 1 pt
        float newFontSize = currentFontSize - 1;

        // Мінімальний розмір шрифту, якщо він менший 1 pt
        if (newFontSize < 1)
        {
            newFontSize = 1;
        }

        // Створюємо новий об'єкт шрифту з оновленим розміром
        Font newFont = new Font(clickedLabel.Font.FontFamily, newFontSize);
    }
}

```

```

        // Встановлюємо новий шрифт для Label
        clickedLabel.Font = newFont;
    }
}

private void label3_DoubleClick(object sender, EventArgs e)
{
    Label clickedLabel = sender as Label;
    if (clickedLabel != null)
    {
        // Отримуємо поточний розмір шрифта
        float currentFontSize = clickedLabel.Font.Size;

        // Зменшуємо розмір шрифта на 1 pt
        float newFontSize = currentFontSize - 1;

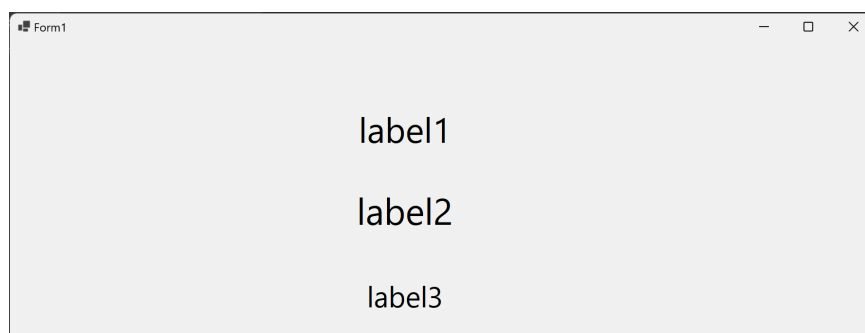
        // Мінімальний розмір шрифта, якщо він менший 1 pt
        if (newFontSize < 1)
        {
            newFontSize = 1;
        }

        // Створюємо новий об'єкт шрифта з оновленим розміром
        Font newFont = new Font(clickedLabel.Font.FontFamily, newFontSize);

        // Встановлюємо новий шрифт для Label
        clickedLabel.Font = newFont;
    }
}
}
}

```

Результат:



Завдання 4*

Виконати минуле завдання з одним обробником подій для усіх рядків, використовуючи параметр `object sender`, який приймає обробник.

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

namespace Task_04
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            InitializeEventHandlers();
        }

        private void InitializeEventHandlers()
        {
            label1.DoubleClick += Label_DoubleClick;
            label2.DoubleClick += Label_DoubleClick;
            label3.DoubleClick += Label_DoubleClick;
        }

        private void Label_DoubleClick(object sender, EventArgs e)
        {
            Label clickedLabel = sender as Label;
            if (clickedLabel != null)
            {
                // Отримуємо поточний розмір шрифту
                float currentFontSize = clickedLabel.Font.Size;

                // Зменшуємо розмір шрифту на 1 pt
                float newFontSize = currentFontSize - 1;

                // Мінімальний розмір шрифту, якщо він менший 1 pt
                if (newFontSize < 1)
                {
                    newFontSize = 1;
                }
            }
        }
    }
}
```

```

// Створюємо новий об'єкт шрифту з оновленим розміром
Font newFont = new Font(clickedLabel.Font.FontFamily, newFontSize);

// Встановлюємо новий шрифт для Label
clickedLabel.Font = newFont;
    }
}
}
}

```

Результат:



Завдання 5 (події MouseEnter та MouseLeave)

Створити проект із формою, в центрі якої мають знаходитися три кнопки (компонент Button) з довільними написами. Для кожної кнопки зробити перехоплення події MouseEnter та MouseLeave таким чином, щоб відповідні функції-перехоплювачі трансформують відповідну кнопку згідно од варіанту, а потім повертають її до початкового стану

№ варіанту	Вид трансформації кнопки
2.	Габаритні розміри зменшуються на 1px

```

namespace Task_05
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}

```

```

private void button1_MouseEnter(object sender, EventArgs e)
{
    Button enteredButton = sender as Button;
    if (enteredButton != null)
    {
        // Збережемо початковий розмір кнопки
        enteredButton.Tag = enteredButton.Size;

        // Змінюємо розмір кнопки
        enteredButton.Width -= 1;
        enteredButton.Height -= 1;
    }
}

```

```

private void button1_MouseLeave(object sender, EventArgs e)
{
    Button leftButton = sender as Button;
    if (leftButton != null && leftButton.Tag is Size)
    {
        // Повертаємо розмір кнопки до початкового стану
        leftButton.Size = (Size)leftButton.Tag;
    }
}

```

```

private void button2_MouseEnter(object sender, EventArgs e)
{
    Button enteredButton = sender as Button;
    if (enteredButton != null)
    {
        // Збережемо початковий розмір кнопки
        enteredButton.Tag = enteredButton.Size;

        // Змінюємо розмір кнопки
        enteredButton.Width -= 1;
        enteredButton.Height -= 1;
    }
}

```

```

private void button2_MouseLeave(object sender, EventArgs e)
{
    Button leftButton = sender as Button;
    if (leftButton != null && leftButton.Tag is Size)

```

```

    {
        // Повертаємо розмір кнопки до початкового стану
        leftButton.Size = (Size)leftButton.Tag;
    }
}

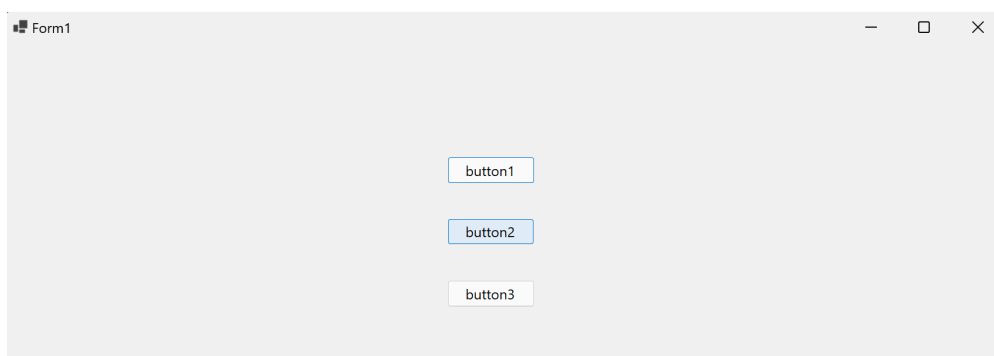
private void button3_MouseEnter(object sender, EventArgs e)
{
    Button enteredButton = sender as Button;
    if (enteredButton != null)
    {
        // Збережемо початковий розмір кнопки
        enteredButton.Tag = enteredButton.Size;

        // Змінюємо розмір кнопки
        enteredButton.Width -= 1;
        enteredButton.Height -= 1;
    }
}

private void button3_MouseLeave(object sender, EventArgs e)
{
    Button leftButton = sender as Button;
    if (leftButton != null && leftButton.Tag is Size)
    {
        // Повертаємо розмір кнопки до початкового стану
        leftButton.Size = (Size)leftButton.Tag;
    }
}
}
}

```

Результат:



Завдання 6*

Виконати минуле завдання з одним обробником подій для усіх кнопок, використовуючи параметр object sender, який приймає обробник.

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

namespace Task_06
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            InitializeEventHandlers();
        }

        private void InitializeEventHandlers()
        {
            // Прив'язка одного обробника подій для всіх кнопок
            button1.MouseEnter += Button_MouseEnter;
            button1.MouseLeave += Button_MouseLeave;
            button2.MouseEnter += Button_MouseEnter;
            button2.MouseLeave += Button_MouseLeave;
            button3.MouseEnter += Button_MouseEnter;
            button3.MouseLeave += Button_MouseLeave;
        }

        private void Button_MouseEnter(object sender, EventArgs e)
        {
            Button enteredButton = sender as Button;
            if (enteredButton != null)
            {
                // Збережемо початковий розмір кнопки
                enteredButton.Tag = enteredButton.Size;

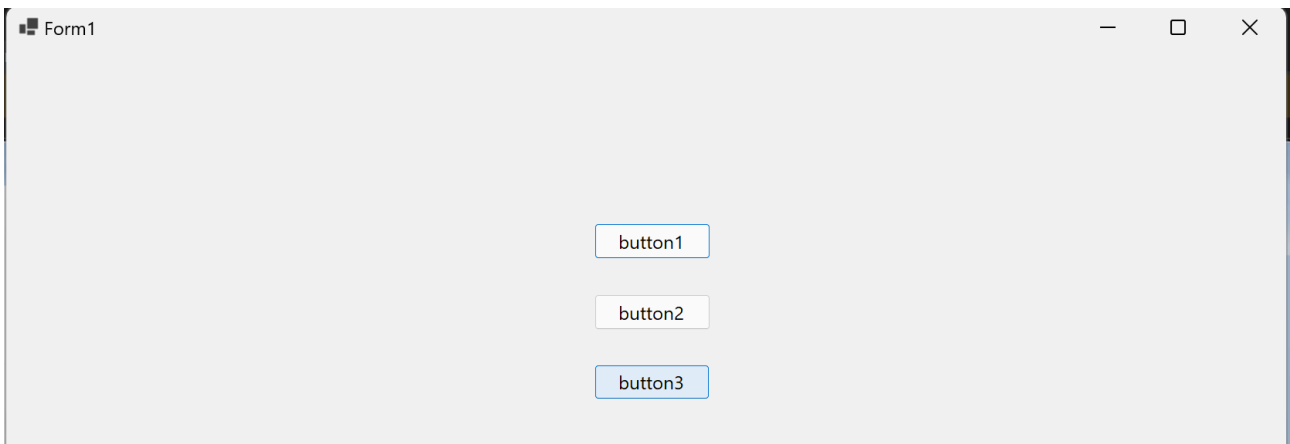
                // Змінюємо розмір кнопки
                enteredButton.Width -= 1;
                enteredButton.Height -= 1;
            }
        }
    }
}
```

```

private void Button_MouseLeave(object sender, EventArgs e)
{
    Button leftButton = sender as Button;
    if (leftButton != null && leftButton.Tag is Size)
    {
        // Повертаємо розмір кнопки до початкового стану
        leftButton.Size = (Size)leftButton.Tag;
    }
}
}
}

```

Результат:



Завдання 7 (події MouseDown та MouseUp)

Створити проект із формою, в центрі якої мають знаходитися три рядки довільного тексту (компонент Label). Для кожного рядка зробити перехоплення подій MouseDown та MouseUp таким чином, щоб відповідна функції-перехоплювачі трансформували б рядок, для якого настає подія натискання кнопки миші, згідно до варіанту, а потім при відпусканні кнопки повертали б його до початкового стану

№ варіанту	Вид трансформації рядка
2.	Розмір шрифту тексту зменшується на 1pt

```

namespace Task_07
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void label1_MouseDown(object sender, MouseEventArgs e)
        {
            // Перевірка, чи це натискання лівої кнопки миші
            if (e.Button == MouseButton.Left)
            {
                Label clickedLabel = sender as Label;
                if (clickedLabel != null)
                {
                    // Змінюємо вигляд рядка тексту при натисканні
                    clickedLabel.Font = new Font(clickedLabel.Font, FontStyle.Bold);
                    clickedLabel.ForeColor = Color.Red;
                }
            }
        }

        private void label1_MouseUp(object sender, MouseEventArgs e)
        {
            // Перевірка, чи це відпускання лівої кнопки миші
            if (e.Button == MouseButton.Left)
            {
                Label clickedLabel = sender as Label;
                if (clickedLabel != null)
                {
                    // Повертаємо вигляд рядка тексту до початкового стану при
                    відпусканні
                    clickedLabel.Font = new Font(clickedLabel.Font, FontStyle.Regular);
                    clickedLabel.ForeColor = SystemColors.ControlText;
                }
            }
        }

        private void label2_MouseDown(object sender, MouseEventArgs e)
        {

```

```
// Перевірка, чи це натискання лівої кнопки миші
if (e.Button == MouseButton.Left)
{
    Label clickedLabel = sender as Label;
    if (clickedLabel != null)
    {
        // Змінюємо вигляд рядка тексту при натисканні
        clickedLabel.Font = new Font(clickedLabel.Font, FontStyle.Bold);
        clickedLabel.ForeColor = Color.Red;
    }
}
}
```

```
private void label2_MouseUp(object sender, MouseEventArgs e)
{
    // Перевірка, чи це відпускання лівої кнопки миші
    if (e.Button == MouseButton.Left)
    {
        Label clickedLabel = sender as Label;
        if (clickedLabel != null)
        {
            // Повертаємо вигляд рядка тексту до початкового стану при
            відпусканні
            clickedLabel.Font = new Font(clickedLabel.Font, FontStyle.Regular);
            clickedLabel.ForeColor = SystemColors.ControlText;
        }
    }
}
```

```
private void label3_MouseDown(object sender, MouseEventArgs e)
{
    // Перевірка, чи це натискання лівої кнопки миші
    if (e.Button == MouseButton.Left)
    {
        Label clickedLabel = sender as Label;
        if (clickedLabel != null)
        {
            // Змінюємо вигляд рядка тексту при натисканні
            clickedLabel.Font = new Font(clickedLabel.Font, FontStyle.Bold);
            clickedLabel.ForeColor = Color.Red;
        }
    }
}
```



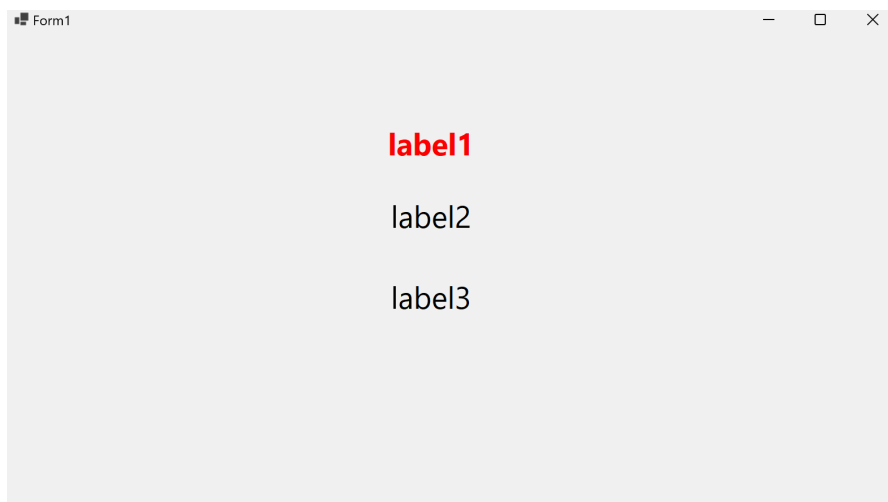
```

    }

private void label3_MouseUp(object sender, MouseEventArgs e)
{
    // Перевірка, чи це відпускання лівої кнопки миші
    if (e.Button == MouseButtons.Left)
    {
        Label clickedLabel = sender as Label;
        if (clickedLabel != null)
        {
            // Повертаємо вигляд рядка тексту до початкового стану при
            відпусканні
            clickedLabel.Font = new Font(clickedLabel.Font, FontStyle.Regular);
            clickedLabel.ForeColor = SystemColors.ControlText;
        }
    }
}
}
}
}

```

Результат:



Завдання 8

Виконати минуле завдання з одним обробником подій для усіх рядків, використовуючи параметр object sender, який приймає обробник.

```
using System;
using System.Windows.Forms; // Убедитесь, что это пространство имен
подключено

namespace Task_08
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            InitializeEventHandlers();
        }

        private void InitializeEventHandlers()
        {
            // Прив'язка одного обробника подій для всіх рядків тексту
            label1.MouseDown += Label_MouseDown;
            label1.MouseUp += Label_MouseUp;
            label2.MouseDown += Label_MouseDown;
            label2.MouseUp += Label_MouseUp;
            label3.MouseDown += Label_MouseDown;
            label3.MouseUp += Label_MouseUp;
        }

        private void Label_MouseDown(object sender, MouseEventArgs e)
        {
            // Перевірка, чи це натискання лівої кнопки миші
            if (e.Button == MouseButtons.Left)
            {
                Label clickedLabel = sender as Label;
                if (clickedLabel != null)
                {
                    // Змінюємо вигляд рядка тексту при натисканні
                    clickedLabel.Font = new Font(clickedLabel.Font, FontStyle.Bold);
                    clickedLabel.ForeColor = Color.Red;
                }
            }
        }
    }
}
```

```

    }

    private void Label_MouseUp(object sender, MouseEventArgs e)
    {
        // Перевірка, чи це відпускання лівої кнопки миші
        if (e.Button == MouseButtons.Left)
        {
            Label clickedLabel = sender as Label;
            if (clickedLabel != null)
            {
                // Повертаємо вигляд рядка тексту до початкового стану при
                відпусканні
                clickedLabel.Font = new Font(clickedLabel.Font, FontStyle.Regular);
                clickedLabel.ForeColor = SystemColors.ControlText;
            }
        }
    }
}

```

Результат:



Завдання 9 (події Enter та Leave)

Створити проект із формою, в центрі якої мають знаходитися три текстові поля (компонент TextBox). Для кожного поля зробити перехоплення події Enter та Leave таким чином, щоб відповідні функції-перехоплювачі трансформують відповідне поле згідно до варіанту, а потім повертають його до початкового стану

№ варіанту	Вид трансформації кнопки
2.	Габаритні розміри зменшуються на 1px

```
namespace Task_09
```

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

        private void textBox1_Enter(object sender, EventArgs e)
        {
            TextBox enteredTextBox = sender as TextBox;
            if (enteredTextBox != null)
            {
                // Збережемо початковий розмір TextBox
                enteredTextBox.Tag = enteredTextBox.Size;

                // Змінюємо розмір TextBox
                enteredTextBox.Width -= 1;
                enteredTextBox.Height -= 1;
            }
        }

        private void textBox1_Leave(object sender, EventArgs e)
        {
            TextBox leftTextBox = sender as TextBox;
            if (leftTextBox != null && leftTextBox.Tag is Size)
            {
                // Повертаємо розмір TextBox до початкового стану
                leftTextBox.Size = (Size)leftTextBox.Tag;
            }
        }
    }
}
```

```

    }
}

```

```

private void textBox2_Enter(object sender, EventArgs e)
{
    TextBox enteredTextBox = sender as TextBox;
    if (enteredTextBox != null)
    {
        // Збережемо початковий розмір TextBox
        enteredTextBox.Tag = enteredTextBox.Size;

        // Змінюємо розмір TextBox
        enteredTextBox.Width -= 1;
        enteredTextBox.Height -= 1;
    }
}

```

```

private void textBox2_Leave(object sender, EventArgs e)
{
    TextBox leftTextBox = sender as TextBox;
    if (leftTextBox != null && leftTextBox.Tag is Size)
    {
        // Повертаємо розмір TextBox до початкового стану
        leftTextBox.Size = (Size)leftTextBox.Tag;
    }
}

```

```

private void textBox3_Enter(object sender, EventArgs e)
{
    TextBox enteredTextBox = sender as TextBox;
    if (enteredTextBox != null)
    {
        // Збережемо початковий розмір TextBox
        enteredTextBox.Tag = enteredTextBox.Size;

        // Змінюємо розмір TextBox
        enteredTextBox.Width -= 1;
        enteredTextBox.Height -= 1;
    }
}

```

```

private void textBox3_Leave(object sender, EventArgs e)

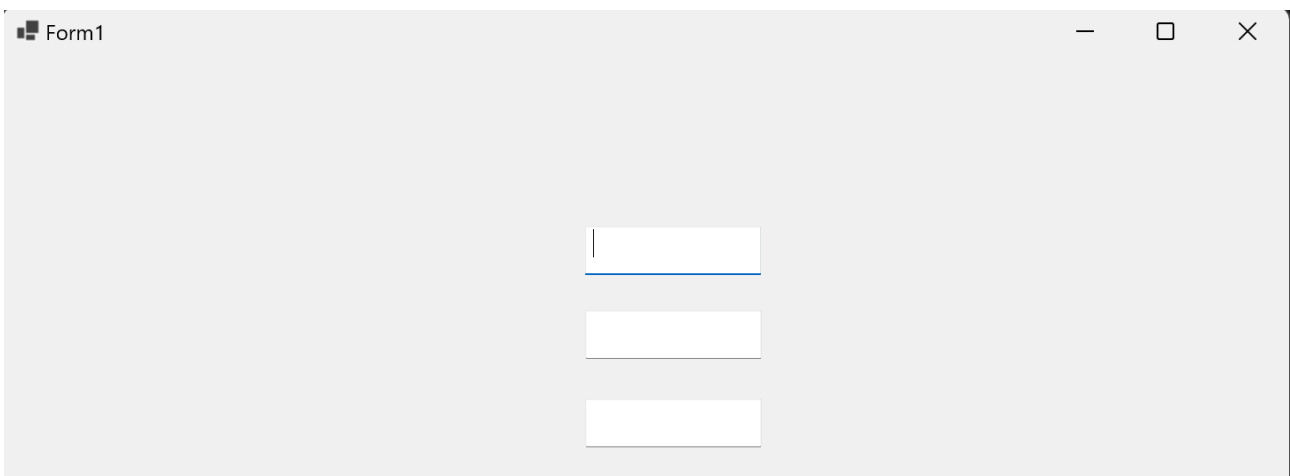
```

```

{
    TextBox leftTextBox = sender as TextBox;
    if (leftTextBox != null && leftTextBox.Tag is Size)
    {
        // Повертаємо розмір TextBox до початкового стану
        leftTextBox.Size = (Size)leftTextBox.Tag;
    }
}
}
}
}

```

Результат:



Завдання 10*

Виконати минуле завдання з одним обробником подій для усіх текстових полів, використовуючи параметр object sender, який приймає обробник.

namespace Task_10

```

{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            // Додаємо обробник подій для всіх текстових полів
            textBox1.Enter += TextBox_Enter;
            textBox1.Leave += TextBox_Leave;
            textBox2.Enter += TextBox_Enter;
            textBox2.Leave += TextBox_Leave;
            textBox3.Enter += TextBox_Enter;
            textBox3.Leave += TextBox_Leave;
        }
    }
}

```

```

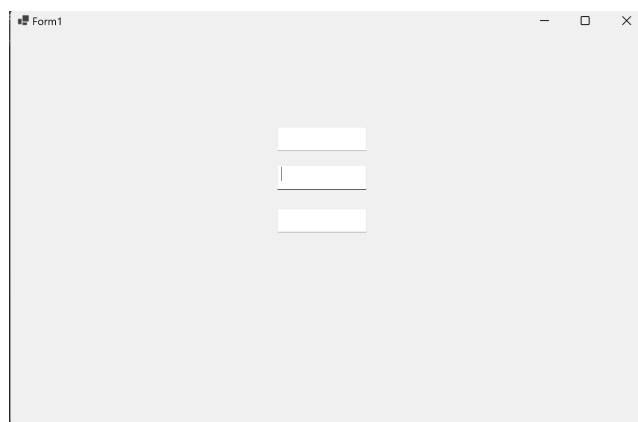
private void TextBox_Enter(object sender, EventArgs e)
{
    TextBox enteredTextBox = sender as TextBox;
    if (enteredTextBox != null)
    {
        // Збережемо початковий розмір TextBox
        enteredTextBox.Tag = enteredTextBox.Size;

        // Змінюємо розмір TextBox
        enteredTextBox.Width -= 1;
        enteredTextBox.Height -= 1;
    }
}

private void TextBox_Leave(object sender, EventArgs e)
{
    TextBox leftTextBox = sender as TextBox;
    if (leftTextBox != null && leftTextBox.Tag is Size)
    {
        // Повертаємо розмір TextBox до початкового стану
        leftTextBox.Size = (Size)leftTextBox.Tag;
    }
}
}
}

```

Результат:



Завдання 11 (події KeyDown, KeyUp та KeyPress)

Створити проект із формою, в центрі якої має знаходитися текстове поле (компонент TextBox), для якого зробити перехоплення однієї з подій KeyDown, KeyUp чи KeyPress таким чином, щоб дозволити друк лише символів згідно до варіанту

№ варіанту	Дозволені символи
2.	Цифри кількістю не більше 10 та лише раз десяткова точка

```

namespace Task_11
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void textBox1_KeyDown(object sender, KeyEventArgs e)
        {
            // Перехоплюємо подію KeyDown
            if (e.KeyCode == Keys.Decimal && (sender as TextBox).Text.Contains("."))
            {
                e.SuppressKeyPress = true; // Забороняємо дублювання десяткової точки
            }
        }

        private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
        {
            // Перехоплюємо подію KeyPress
            if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) && e.KeyChar != '.')
            {
                e.Handled = true; // Забороняємо введення символів, окрім цифр і
                // десяткової точки
            }

            // Перевірка на кількість десяткових точок та кількість цифр

```

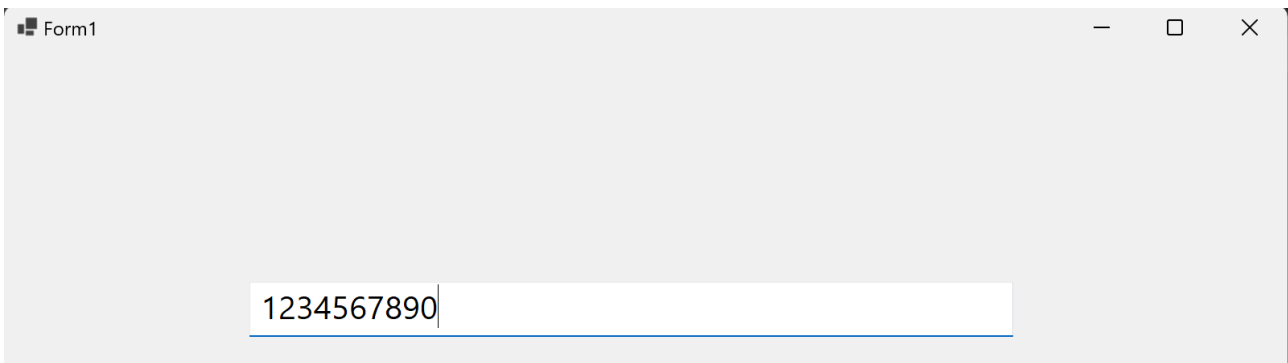


```

if (e.KeyChar == '.')
{
    if ((sender as TextBox).Text.Contains("."))
    {
        e.Handled = true; // Забороняємо дублювання десяткової точки
    }
    else if ((sender as TextBox).Text.Length >= 10)
    {
        e.Handled = true; // Забороняємо введення більше 10 цифр після
десяткової точки
    }
}
else if (char.IsDigit(e.KeyChar))
{
    if (((sender as TextBox).Text.Length >= 10 && !((sender as
TextBox).Text.Contains("."))) ||
        ((sender as TextBox).Text.Length >= 11 && ((sender as
TextBox).Text.Contains("."))))
    {
        e.Handled = true; // Забороняємо введення більше 10 цифр перед
десятковою точкою або більше 11 з нею
    }
}
}
}

```

Результат:



The screenshot shows a standard Windows application window titled "Form1". Inside the window, there is a single text box. The text box contains the numeric string "1234567890". The text box has a white background and a thin blue border. The window's title bar includes the text "Form1" and standard minimize, maximize, and close buttons.

Завдання 12 (подія SelectedIndexChanged)

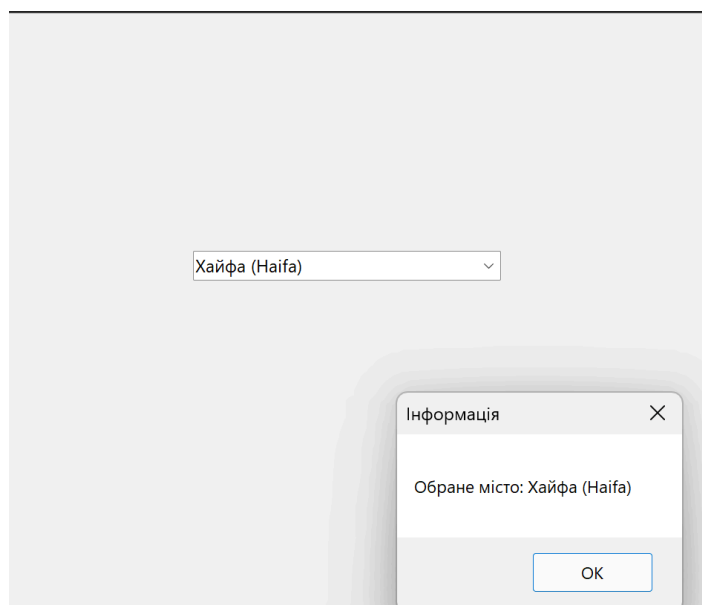
Побудувати на формі проекту ComboBox з першого завдання другої лабораторної роботи. Прив'язати до нього обробник події SelectedIndexChanged таким чином, щоб відповідна функція-обробник виводила б відповідне місто в інформаційне діалогове вікно MsgBox.

```
namespace Task_12
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
            {
                // Отримуємо обране місто з ComboBox
                string selectedCity = comboBox1.SelectedItem.ToString();

                // Виводимо обране місто в MsgBox
                MessageBox.Show("Обране місто: " + selectedCity, "Інформація");
            }
        }
    }
}
```

Результат:



Завдання 13

Створити проект із формою до якої прив'язати клавіатурні комбінації (HotKeys) згідно варіанту, які б додавали а т видаляли компонент управління на форму згідно варіанту

№ варіанту	Клавіатурні комбінації додання/видалення	Доданий/ видалений компонент
2.	Ctrl+A / Ctrl+D	Button

```
namespace Task_13
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.Control && e.KeyCode == Keys.A) // Для Cmd+A
            {
                // Додати нову кнопку на форму
                Button newButton = new Button();
                newButton.Text = "Нова кнопка";
                newButton.Size = new System.Drawing.Size(100, 50);
                newButton.Location = new System.Drawing.Point((this.ClientSize.Width -
newButton.Width) / 2, (this.ClientSize.Height - newButton.Height) / 2);
                this.Controls.Add(newButton);
            }
            else if (e.Control && e.KeyCode == Keys.S) // Змінено з Ctrl+D на Ctrl+S
            {
                // Видалити останню кнопку на формі (якщо є)
                if (this.Controls.Count > 0)
                {
                    Control lastControl = this.Controls[this.Controls.Count - 1];
                    if (lastControl is Button)
                    {
                        this.Controls.Remove(lastControl);
                        lastControl.Dispose();
                    }
                }
            }
        }
    }
}
```

```

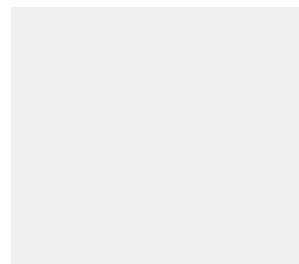
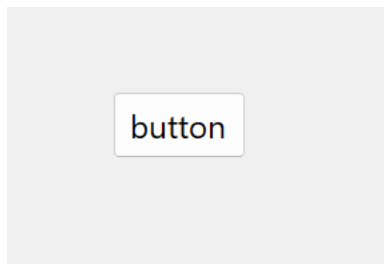
    }
    }
}

private void Form1_Load(object sender, EventArgs e)
{
    // Встановлюємо KeyPreview на true, щоб форма отримувала клавішні
    події
    this.KeyPreview = true;
}

}
}

```

Результат:



Завдання 14 (подія Form.Load)

Для порожньої форми проекту обробити подію Form.Load таким чином, щоб перед завантаження форми з'являлося б діалогове вікно MsgBox зі запитом щодо завантаження форми проекту і варіантами відповідей ТАК та НІ. При виборі варіанта НІ - форма не завантажується, а ТАК - завантажується в режимі згідно до варіанту

№ варіанту	Режим завантаження форми
2.	Форма займає верхню половину екрану

```

using System;
using System.Windows.Forms;

```

```

namespace Task_14
{
    public partial class Form1 : Form
    {

```

```

private bool dialogShown = false;

public Form1()
{
    InitializeComponent();
    // Додаємо обробник події Load до форми
    this.Load += Form1_Load;
}

private void Form1_Load(object sender, EventArgs e)
{
    if (!dialogShown)
    {
        dialogShown = true;

        // Виводимо діалогове вікно з запитом
        DialogResult result = MessageBox.Show("Завантажити форму
проекту?", "Запит", MessageBoxButtons.YesNo);

        // Перевіряємо результат діалогового вікна
        if (result == DialogResult.Yes)
        {
            // Якщо користувач обрав "ТАК", то завантажуюємо форму
            // Задайте розмір і розташування форми (наприклад, верхню
половину екрану)
            this.Size = new
System.Drawing.Size(Screen.PrimaryScreen.Bounds.Width,
Screen.PrimaryScreen.Bounds.Height / 2);
            this.StartPosition = FormStartPosition.Manual;
            this.Location = new System.Drawing.Point(0, 0);

            // Додайте код для завантаження форми проекту тут
        }
        else
        {
            // Якщо користувач обрав "НІ", то не завантажуюємо форму
            this.Close();
        }
    }
}

// Видаляємо обробник події Load
this.Load -= Form1_Load;

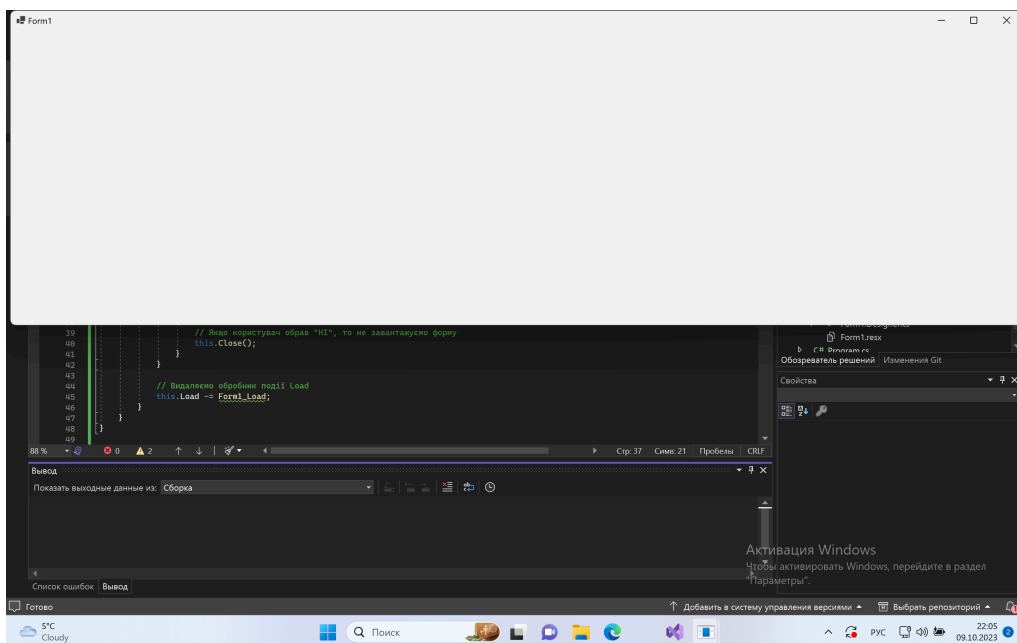
```

Запит

Завантажити форму проекту?

Да

Нет



Висновок :

Лабораторна робота передбачала створення WinForms додатків на C# з використанням обробників подій для елементів інтерфейсу. Ми вивчили, як обробляти різні події, такі як Click, DoubleClick, MouseEnter, MouseLeave, MouseDown, MouseUp, Enter та Leave для різних елементів у Windows Forms додатках.

Процес створення обробників подій включав в себе наступні кроки:

1. Вибір елемента інтерфейсу, для якого ми хочемо створити обробник подій.
2. Додавання обробників подій для обраного елемента, вказавши необхідну подію (наприклад, Click або MouseEnter).
3. Реалізація обробників подій, де виконується код, пов'язаний із відповідною подією (наприклад, зміна розміру елемента при MouseEnter і повернення до початкового розміру при MouseLeave).

Ми також дослідили варіанти, коли один обробник подій використовувався для кількох елементів і визначення об'єкта, який спричинив подію, за допомогою параметра `sender`.

Лабораторна робота дозволила нам отримати практичні навички у створенні і налаштуванні обробників подій для елементів інтерфейсу Windows Forms. Ці навички корисні при розробці програм з інтерактивним інтерфейсом користувача, де важливо реагувати на події, які виникають при взаємодії з користувачем.

У цьому проєкті ми створили Windows Forms додаток з гарячими клавішами (HotKeys) для динамічного додавання та видалення кнопок на формі. Гарячі клавіші Ctrl+A використовуються для додавання кнопки, яка має розмір 100x50 і розташовується по центру форми. Гаряча клавіша Ctrl+S (раніше Ctrl+D) використовується для видалення останньої доданої кнопки.

Цей проєкт показує, як можна динамічно керувати елементами керування на формі за допомогою клавіатурних комбінацій і зручно використовувати гарячі клавіші для додавання та видалення елементів на формі в залежності від ваших потреб.