

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

Звіт

Лабораторна робота 8

з дисципліни « C# .Net »

Виконав: студент групи КН-921Б

Бірюков Д .Є.

Перевірив:

Івашко А.В.

Харків 2023

Зміст

Завдання 1.....	1
Завдання 2.....	3
Завдання 3.....	6
Завдання 4.....	9
Завдання 5.....	13
Завдання 6.....	16
Завдання 7.....	19
Завдання 8.....	22
Завдання 9.....	25
Завдання 10.....	27
Завдання 11.....	29
Завдання 12.....	32
Висновок.....	34

Завдання 1

Для форми зі завдання 3 другої лабораторної роботи свого варіанту встановити обробники подій клацання на кнопки: Скинути, який очищував би текстові усі поля та відмінював би маніпуляції іншої кнопки, Відіслати, який би виконував маніпуляції із вмістом текстових полів згідно до варіанту

№ варіанту	Маніпуляція для обробника кнопки Відіслати
2.	Зміна стандартного кольору фону полів на будь-який інший за бажанням

```
namespace Task__01
```

```
{
```

```
    public partial class Form1 : Form
```

```
    {
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        private void button1_Click(object sender, EventArgs e)
```

```
        {
```

```
            // Проверка, что все четыре TextBox'а заполнены
```

```
            if (string.IsNullOrEmpty(textBox1.Text) ||
```

```
                string.IsNullOrEmpty(textBox2.Text) ||
```

```
                string.IsNullOrEmpty(textBox3.Text) ||
```

```
                string.IsNullOrEmpty(textBox4.Text))
```

```
            {
```

```
                // Один из TextBox'ов пуст, выдаем сообщение
```

```
                MessageBox.Show("Введите все данные!", "Внимание",
```

```
                    MessageBoxButtons.OK, MessageBoxIcon.Warning);
```

```
            }
```

```
        else
```

```
        {
```

```
            // Установка нового цвета фона для текстовых полей
```

```
            textBox1.BackColor = Color.Yellow; // Добавление цвета
```

```
            textBox2.BackColor = Color.Green; // Добавление цвета
```

```
textBox3.BackColor = Color.Blue; //Добавление цвета
textBox4.BackColor = Color.Red; // Добавление вета
```

```
// Опциональное сообщение об успешной отправке
MessageBox.Show("Данные отправлены! Спасибо");
```

```
// Вы можете изменить цвета на любые другие по вашему выбору
}
```

```
}
```

```
private void button2_Click(object sender, EventArgs e)
```

```
{
```

```
    textBox1.Clear();
```

```
    textBox2.Clear();
```

```
    textBox3.Clear();
```

```
    textBox4.Clear();
```

```
    textBox1.BackColor = SystemColors.Window;
```

```
    textBox2.BackColor = SystemColors.Window;
```

```
    textBox3.BackColor = SystemColors.Window;
```

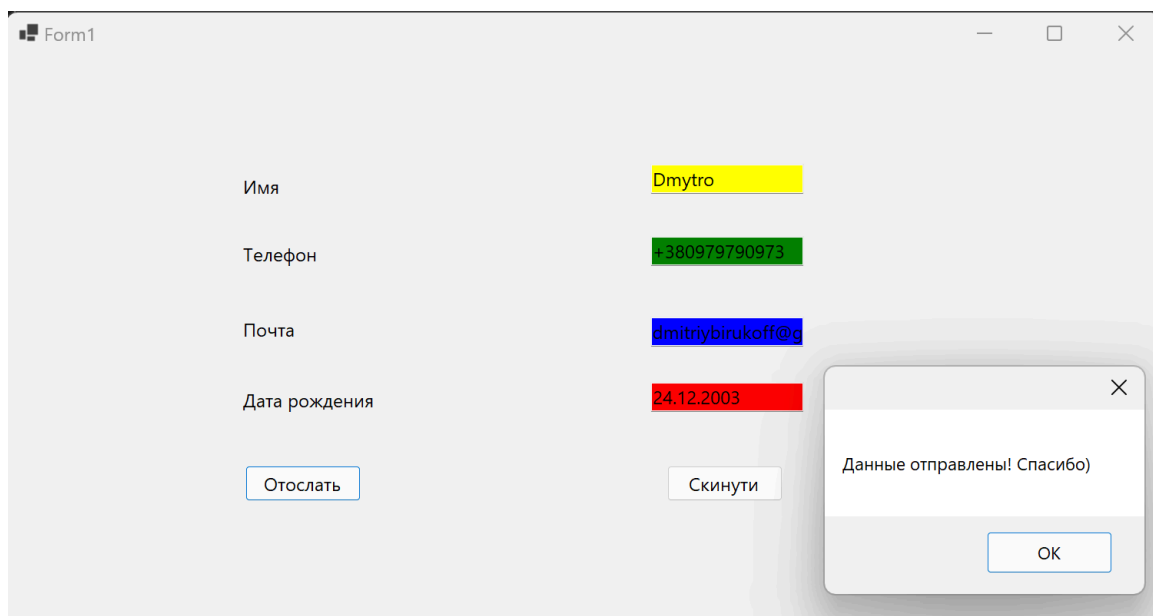
```
    textBox4.BackColor = SystemColors.Window;
```

```
}
```

```
}
```

```
}
```

Результат:



Завдання 2

Для форми зі завдання 4 другої лабораторної роботи свого варіанту встановити обробники подій клацання на кнопки: Скинути, який повертає би комбінацію вибору радіо кнопок у стан за замовчуванням та відмінюванням би маніпуляції іншої кнопки, Відіслати, який би виконував маніпуляції зі вмістом текстових рядків, вибраних відповідними радіо кнопками, згідно до варіанту

№ варіанту	Маніпуляція для обробника кнопки Відіслати
2.	Зміна стандартного кольору фону рядків на будь-який інший за бажанням

```
namespace Task_04
```

```
{
```

```
    public partial class Form1 : Form
```

```
    {
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        private void button2_Click(object sender, EventArgs e)
```

```
        {
```

```
            // Восстанавливаем стандартный цвет фона радиокнопок
```

```
            radioButton1.BackColor = SystemColors.Control;
```

```
            radioButton2.BackColor = SystemColors.Control;
```

```
            radioButton3.BackColor = SystemColors.Control;
```

```
            radioButton4.BackColor = SystemColors.Control;
```

```
            radioButton6.BackColor = SystemColors.Control;
```

```
            radioButton7.BackColor = SystemColors.Control;
```

```
            radioButton8.BackColor = SystemColors.Control;
```

```
        }
```

```
        private void button1_Click(object sender, EventArgs e)
```

```
        {
```

```
            // Проверяем, выбрана ли одна из радиокнопок с номерами от 1 до 4
```

```
bool citySelected = radioButton1.Checked || radioButton2.Checked ||
radioButton3.Checked || radioButton4.Checked;
```

```
// Проверяем, выбрана ли одна из радиокнопок с номерами от 6 до 8
bool transportSelected = radioButton6.Checked || radioButton7.Checked ||
radioButton8.Checked;
```

```
// Если одна из радиокнопок с номерами от 1 до 4 и одна из радиокнопок
с номерами от 6 до 8 выбраны,
```

```
// выводим сообщение "Спасибо за выбор"
```

```
if (citySelected && transportSelected)
```

```
{
```

```
    MessageBox.Show("Спасибо за выбор!");
```

```
}
```

```
else
```

```
{
```

```
    // Иначе выводим сообщение "Выберите город и транспорт"
```

```
    MessageBox.Show("Выберите город и транспорт", "Внимание",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
```

```
}
```

```
// Изменяем цвет фона радиокнопок на основе их состояния
```

```
radioButton1.BackColor = radioButton1.Checked ? Color.Yellow :
SystemColors.Control;
```

```
radioButton2.BackColor = radioButton2.Checked ? Color.Yellow :
SystemColors.Control;
```

```
radioButton3.BackColor = radioButton3.Checked ? Color.Yellow :
SystemColors.Control;
```

```
radioButton4.BackColor = radioButton4.Checked ? Color.Yellow :
SystemColors.Control;
```

```
radioButton6.BackColor = radioButton6.Checked ? Color.Green :
SystemColors.Control;
```

```
radioButton7.BackColor = radioButton7.Checked ? Color.Green :
SystemColors.Control;
```

```
radioButton8.BackColor = radioButton8.Checked ? Color.Green :
SystemColors.Control;
```

```
}
```

```

    }
}

```

Результат:

Завдання 3

Для форми зі завдання 5 другої лабораторної роботи свого варіанту встановити обробники подій клацання на кнопки: Скинути, який повертає комбінацію вибору селективних кнопок (чекбоксів) у стан за замовчуванням, а Відмінював би маніпуляції іншої кнопки, Відіслати, який би виконував маніпуляції зі вмістом текстових рядків, вибраних відповідними чекбоксами, згідно до варіанту

№ варіанту	Маніпуляція для обробника кнопки Відіслати
2.	Зміна стандартного кольору фону рядків на будь-який інший за бажанням

namespace Task_05

```
{
```

```
    public partial class Form1 : Form
```

```
    {
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        private void button2_Click(object sender, EventArgs e)
```

```
        {
```

```
            checkBox1.Checked = false;
```

```
            checkBox2.Checked = true;
```

```
checkBox3.Checked = true;  
checkBox4.Checked = false;
```

```
// Восстанавливаем стандартный цвет фона чекбоксов
```

```
checkBox1.BackColor = SystemColors.Control;  
checkBox2.BackColor = SystemColors.Control;  
checkBox3.BackColor = SystemColors.Control;  
checkBox4.BackColor = SystemColors.Control;
```

```
}
```

```
private void button1_Click(object sender, EventArgs e)
```

```
{
```

```
// Проверяем, выбран ли хотя бы один из чекбоксов
```

```
if (checkBox1.Checked || checkBox2.Checked || checkBox3.Checked ||  
checkBox4.Checked)
```

```
{
```

```
if (checkBox1.Checked)
```

```
    checkBox1.BackColor = Color.Yellow;
```

```
else
```

```
    checkBox1.BackColor = SystemColors.Control;
```

```
if (checkBox2.Checked)
```

```
    checkBox2.BackColor = Color.Purple;
```

```
else
```

```
    checkBox2.BackColor = SystemColors.Control;
```

```
if (checkBox3.Checked)
```

```
    checkBox3.BackColor = Color.Green;
```

```
else
```

```
    checkBox3.BackColor = SystemColors.Control;
```

```
if (checkBox4.Checked)
```

```
    checkBox4.BackColor = Color.Pink;
```

```
else
```

```
    checkBox4.BackColor = SystemColors.Control;
```

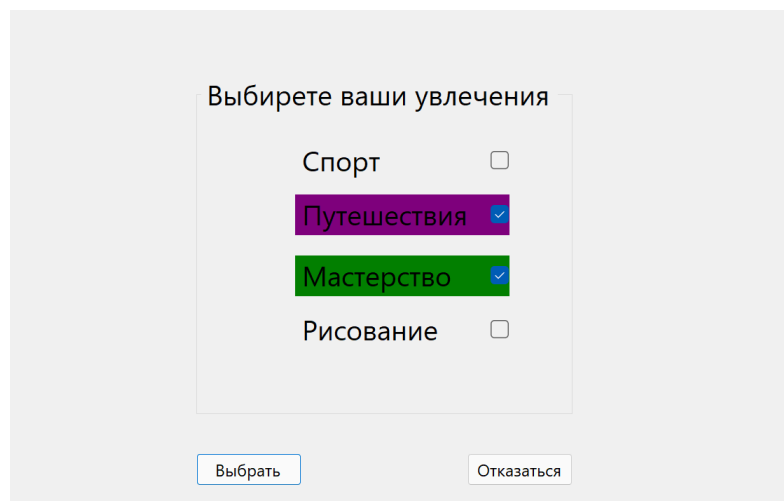


```

        MessageBox.Show("Спасибо за выбор!");
    }
    else
    {
        MessageBox.Show("Выберите хотя бы одно увлечение", "Внимание",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
}
}

```

Результат:



Завдання 4

Для форми зі завдання 6 другої лабораторної роботи свого варіанту встановити обробники подій клацання на кнопки: Очистити, який повертає би стан форми у стан аз замовчуванням, Відправити, який би виводив вибрані параметри форми в окреме діалогове вікно (MsgBox)

```
namespace Task_06
```

```
{
```

```

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

    private void button2_Click(object sender, EventArgs e)
    {
        textBox1.Clear();
        textBox2.Clear();
        textBox3.Clear();
        textBox4.Clear();
        textBox5.Clear();

        comboBox1.Text = "";
        comboBox2.Text = "";

        radioButton1.Checked = false;
        radioButton2.Checked = false;
        radioButton3.Checked = false;
        radioButton4.Checked = false;
        radioButton5.Checked = false;
        radioButton6.Checked = false;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        // Проверяем TextBox'ы с 1 по 5
        if (string.IsNullOrWhiteSpace(textBox1.Text) ||
            string.IsNullOrWhiteSpace(textBox2.Text) ||
            string.IsNullOrWhiteSpace(textBox3.Text) ||
            string.IsNullOrWhiteSpace(textBox4.Text) ||
            string.IsNullOrWhiteSpace(textBox5.Text))
        {

```

```

        MessageBox.Show("Введите все данные", "Внимание",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    // Проверяем ComboBox'ы с 1 по 2
    if (comboBox1.SelectedIndex == -1 || comboBox2.SelectedIndex == -1)
    {
        MessageBox.Show("Введите все данные", "Внимание",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    // Проверяем RadioButton'ы с 1 по 4
    if (!(radioButton1.Checked || radioButton2.Checked || radioButton3.Checked ||
radioButton4.Checked))
    {
        MessageBox.Show("Введите все данные", "Внимание",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    // Проверяем RadioButton'ы с 5 по 6
    if (!(radioButton5.Checked || radioButton6.Checked))
    {
        MessageBox.Show("Введите все данные", "Внимание",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    // Создайте строку, в которой будет храниться сообщение
    string message = $" {groupBox1.Text}: \n {label2.Text}: {textBox1.Text}
\n {label3.Text}: {textBox2.Text} \n {label4.Text}: {textBox3.Text} \n {label5.Text}:
{textBox4.Text} \n {groupBox2.Text}: \n {label6.Text}: {comboBox1.Text} \n
{label7.Text}: {comboBox2.Text} \n {label8.Text}: {textBox5.Text} \n";

    // Создайте переменные для хранения текста выбранных радиокнопок
    string selectedRadioButtonTextPanel1 = "";

```

```
string selectedRadioButtonTextPanel2 = "";
```

```
// Проверьте выбранные RadioButton на panel1 и присвойте их текст
переменной
```

```
if (radioButton1.Checked) selectedRadioButtonTextPanel1 =
radioButton1.Text;
```

```
if (radioButton2.Checked) selectedRadioButtonTextPanel1 =
radioButton2.Text;
```

```
if (radioButton3.Checked) selectedRadioButtonTextPanel1 =
radioButton3.Text;
```

```
if (radioButton4.Checked) selectedRadioButtonTextPanel1 =
radioButton4.Text;
```

```
// Проверьте выбранные RadioButton на panel2 и присвойте их текст
переменной
```

```
if (radioButton5.Checked) selectedRadioButtonTextPanel2 =
radioButton5.Text;
```

```
if (radioButton6.Checked) selectedRadioButtonTextPanel2 =
radioButton6.Text;
```

```
// Добавьте текст выбранных RadioButton к вашей строке message
```

```
if (!string.IsNullOrEmpty(selectedRadioButtonTextPanel1))
```

```
{
    message += $" {groupBox3.Text}: {selectedRadioButtonTextPanel1}\n";
}
```

```
if (!string.IsNullOrEmpty(selectedRadioButtonTextPanel2))
```

```
{
    message += $" {label9.Text}: {selectedRadioButtonTextPanel2}\n";
}
```

```
MessageBox.Show(message, "Информация", MessageBoxButtons.OK,
MessageBoxIcon.Information);
```

```
}
```

```
}
```

```
}
```

Результат:

The screenshot displays a Windows application window titled "Form1" containing a restaurant survey form. The form is divided into three sections: "Контактная информация" (Contact information), "Персональная информация" (Personal information), and "Почему выбрали наше заведение?" (Why did you choose our establishment?).

Контактная информация:

- Имя: Дмитрий
- Телефон: 0979790973
- Почта: dima@gmail.com
- Дата посещения: 12.12.2023

Персональная информация:

- Возраст: 19
- Любимая кухня: Украинская
- Какие блюда вы хотели бы видеть в меню?: Вареники

Почему выбрали наше заведение?:

- ☐ Недалеко от дома/работы
- ☐ Увидел(а) рекламу
- ☒ Посоветовали
- ☐ Оптимальное отношение цена и качество

Порекомендовали бы наше заведение друзьям и знакомым?:

- ☒ Да
- ☐ Нет

At the bottom of the form are two buttons: "Отправить" (Send) and "Отменить" (Cancel).

Overlaid on the form is a small dialog box titled "Информация" (Information) with an information icon. It contains the following text:

Контактная информация:
Имя: Дмитрий
Телефон: 0979790973
Почта: dima@gmail.com
Дата посещения: 12.12.2023
Персональная информация:
Возраст: 19
Любимая кухня: Украинская
Какие блюда вы хотели бы видеть в меню?: Вареники
Почему выбрали наше заведение?: Посоветовали
Порекомендовали бы наше заведение друзьям и знакомым?: Да

The dialog box has an "OK" button at the bottom right.

Завдання 5

Для форми із завдання 7 другої лабораторної роботи свого варіанту встановити обробники подій клацання на кінцеві пункти меню таким чином, щоб вони виводили назви відповідних розділів меню в окреме діалогове вікно (MsgBox)

```
using Microsoft.VisualBasic.ApplicationServices;
```

```
namespace Task_07
```

```
{
```

```
    public partial class Form1 : Form
```

```
    {
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        private void item12ToolStripMenuItem_Click(object sender, EventArgs e)
```

```
        {
```

```
            MessageBox.Show("item11");
```

```
        }
```

```
        private void item12ToolStripMenuItem1_Click(object sender, EventArgs e)
```

```
        {
```

```
            MessageBox.Show("item12");
```

```
        }
```

```
        private void item21ToolStripMenuItem_Click(object sender, EventArgs e)
```

```
        {
```

```
            MessageBox.Show("item21");
```

```
        }
```

```
        private void item22ToolStripMenuItem_Click(object sender, EventArgs e)
```

```
        {
```

```
            MessageBox.Show("item22");
```

```
        }
```

```
        private void item231ToolStripMenuItem_Click(object sender, EventArgs e)
```

```
        {
```

```
            MessageBox.Show("item231");
```

```
        }
```

```
private void item232ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item232");
}
```

```
private void item233ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item233");
}
```

```
private void item234ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item234");
}
```

```
private void item31ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item31");
}
```

```
private void item321ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item321");
}
```

```
private void item322ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item322");
}
```

```
private void item323ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item323");
}
```

```
private void item324ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item324");
}
```

```
private void item33ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item33");
}
```

```
private void item34ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item34");
}
```

```
private void item41ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item41");
}
```

```
private void item42ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item42");
}
```

```
private void item431ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item431");
}
```

```
private void item432ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item432");
}
```

```
private void item433ToolStripMenuItem_Click(object sender, EventArgs e)
```



```

{
    MessageBox.Show("item433");
}

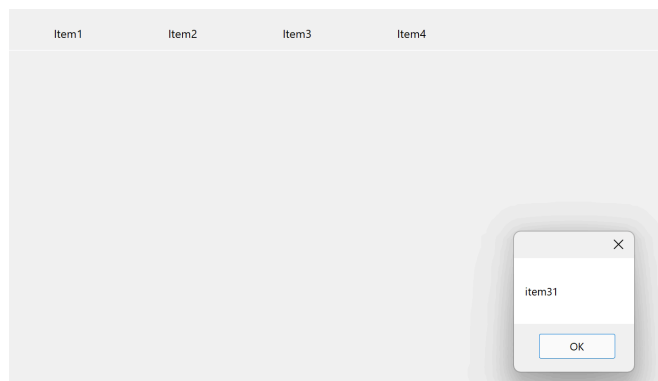
private void item434ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item434");
}

private void item44ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item44");
}

private void item45ToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("item45");
}
}
}

```

Результат:



Завдання 6

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

```
using Microsoft.VisualBasic.ApplicationServices;
```

```

namespace Task_07
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            item12ToolStripMenuItem.Click += Form1_Load;
            item12ToolStripMenuItem1.Click += Form1_Load;
            item21ToolStripMenuItem.Click += Form1_Load;
            item22ToolStripMenuItem.Click += Form1_Load;
            item231ToolStripMenuItem.Click += Form1_Load;
            item232ToolStripMenuItem.Click += Form1_Load;
            item233ToolStripMenuItem.Click += Form1_Load;
            item234ToolStripMenuItem.Click += Form1_Load;
            item31ToolStripMenuItem.Click += Form1_Load;
            item32ToolStripMenuItem.Click += Form1_Load;
            item33ToolStripMenuItem.Click += Form1_Load;
            item34ToolStripMenuItem.Click += Form1_Load;
            item321ToolStripMenuItem.Click += Form1_Load;
            item322ToolStripMenuItem.Click += Form1_Load;
            item323ToolStripMenuItem.Click += Form1_Load;
            item324ToolStripMenuItem.Click += Form1_Load;
            item41ToolStripMenuItem.Click += Form1_Load;
            item42ToolStripMenuItem.Click += Form1_Load;
            item44ToolStripMenuItem.Click += Form1_Load;
            item45ToolStripMenuItem.Click += Form1_Load;
            item431ToolStripMenuItem.Click += Form1_Load;
            item432ToolStripMenuItem.Click += Form1_Load;
            item433ToolStripMenuItem.Click += Form1_Load;
            item434ToolStripMenuItem.Click += Form1_Load;
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // Получаем имя выбранного пункта меню

```

```

var menuItem = (ToolStripMenuItem)sender;
string menuItemName = menuItem.Text;

// Определяем раздел меню, к которому принадлежит выбранный пункт
string sectionName = "";

if (menuItem == item12ToolStripMenuItem1 || menuItem ==
item12ToolStripMenuItem)
{
    sectionName = "Раздел 1";
}
else if (menuItem == item21ToolStripMenuItem || menuItem ==
item22ToolStripMenuItem || menuItem == item231ToolStripMenuItem || menuItem
== item232ToolStripMenuItem || menuItem == item233ToolStripMenuItem ||
menuItem == item234ToolStripMenuItem)
{
    sectionName = "Раздел 2";
}
else if (menuItem == item31ToolStripMenuItem || menuItem ==
item32ToolStripMenuItem || menuItem == item33ToolStripMenuItem || menuItem ==
item34ToolStripMenuItem || menuItem == item321ToolStripMenuItem || menuItem
== item322ToolStripMenuItem || menuItem == item323ToolStripMenuItem ||
menuItem == item324ToolStripMenuItem)
{
    sectionName = "Раздел 3";
}
else if (menuItem == item41ToolStripMenuItem || menuItem ==
item42ToolStripMenuItem || menuItem == item44ToolStripMenuItem || menuItem ==
item45ToolStripMenuItem || menuItem == item431ToolStripMenuItem || menuItem
== item432ToolStripMenuItem || menuItem == item433ToolStripMenuItem ||
menuItem == item434ToolStripMenuItem)
{
    sectionName = "Раздел 4";
}

// Выводим сообщение с названием раздела
MessageBox.Show("Выбран пункт: " + menuItemName + "\nИз раздела: "
+ sectionName);
}

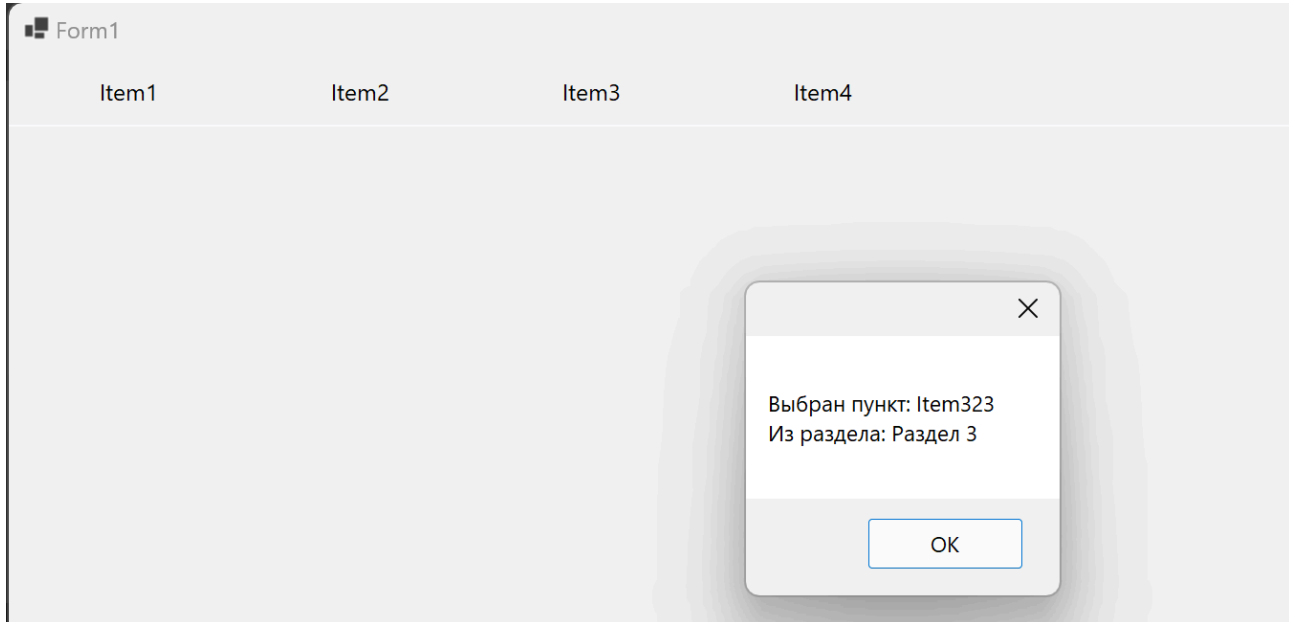
```

```

    }
}

```

Результат:



Завдання 7

Для форми зі завдання 8 другої лабораторної роботи переробити контекстне меню і встановити відповідні обробники подій клацання на пункти оновленого контекстного меню таким чином, щоб вони виконували маніпуляції згідно варіанту

№ варіанту	Маніпуляція для обробників контекстного меню
2.	Зміна кольору фону рядків

```

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

```

```

namespace Task_08
{
    public partial class Form1 : Form

```

```

{
    public Form1()
    {
        InitializeComponent();

        // Привязываем один и тот же ContextMenuStrip ко всем Label.
        label1.ContextMenuStrip = contextMenuStrip1;
        label2.ContextMenuStrip = contextMenuStrip1;
        label3.ContextMenuStrip = contextMenuStrip1;

        // Добавляем обработчики событий для всех пунктов меню.
        foreach (ToolStripMenuItem item in contextMenuStrip1.Items)
        {
            item.Click += ContextMenuItem_Click;
        }
    }

    private void ChangeLabelColor(Label label, Color color)
    {
        label.BackColor = color;
    }

    private void ContextMenuItem_Click(object sender, EventArgs e)
    {
        ToolStripMenuItem menuItem = (ToolStripMenuItem)sender;
        ContextMenuStrip contextMenu = menuItem.GetCurrentParent() as
ContextMenuStrip;
        Label label = contextMenu.SourceControl as Label;

        if (label != null)
        {
            // Определите цвет в зависимости от выбранного пункта меню.
            if (menuItem == item1ToolStripMenuItem)
            {
                ChangeLabelColor(label, Color.Green);
            }
        }
    }
}

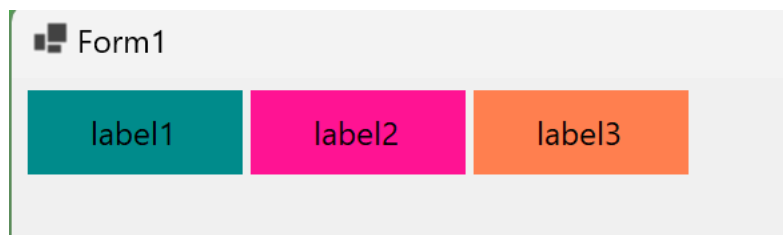
```

```

else if (menuItem == item2ToolStripMenuItem)
{
    ChangeLabelColor(label, Color.Yellow);
}
else if (menuItem == item3ToolStripMenuItem)
{
    ChangeLabelColor(label, Color.Red);
}
}
}
}
}
}

```

Результат:



Завдання 8*

Виконати минуле завдання з одним обробником подій для усіх кінцевих пунктів контекстного меню, використовуючи параметр `object sender`, який приймає обробник.

```

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

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

```

```
// Привязываем один и тот же ContextMenuStrip ко всем Label.
```

```
label1.ContextMenuStrip = contextMenuStrip1;
```

```
label2.ContextMenuStrip = contextMenuStrip1;
```

```
label3.ContextMenuStrip = contextMenuStrip1;
```

```
// Добавляем обработчики событий для всех пунктов меню.
```

```
foreach (ToolStripMenuItem item in contextMenuStrip1.Items)
```

```
{
```

```
    item.Click += ContextMenuItem_Click;
```

```
}
```

```
}
```

```
private void ChangeLabelColor(Label label, Color color)
```

```
{
```

```
    label.BackColor = color;
```

```
}
```

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

```
{
```

```
    ToolStripMenuItem menuItem = (ToolStripMenuItem)sender;
```

```
    ContextMenuStrip contextMenu = menuItem.GetCurrentParent() as  
ContextMenuStrip;
```

```
    Label label = contextMenu.SourceControl as Label;
```

```
    if (label != null)
```

```
{
```

```
    // Определите цвет в зависимости от выбранного пункта меню.
```

```
    if (menuItem == item1ToolStripMenuItem)
```

```
{
```

```
        ChangeLabelColor(label, Color.Green);
```

```
}
```

```
    else if (menuItem == item2ToolStripMenuItem)
```

```
{
```

```
        ChangeLabelColor(label, Color.Yellow);
```

```
}
```

```

else if (menuItem == item3ToolStripMenuItem)
{
    ChangeLabelColor(label, Color.Red);
}
}
}
}
}
}
}

```

Результат:



Завдання 9

Для форми зі завдання 9 другої лабораторної роботи свого варіанту встановити обробники подій клацання на кнопки панелі інструментів таким чином, щоб вони виводили тексти підказок на кнопках в окреме діалогове вікно (MsgBox)

```
using System;
```

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

```
namespace Task_09
```

```
{
```

```
    public partial class Form1 : Form
```

```
    {
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        private void toolStripButton1_Click(object sender, EventArgs e)
```

```
        {
```



```
// Визначаємо, яка саме кнопка була натиснута, використовуючи
параметр sender
```

```
ToolStripButton button = (ToolStripButton)sender;
```

```
// Отримуємо текст підказки для натиснутої кнопки
```

```
string tooltipText = button.ToolTipText;
```

```
// Виводимо текст підказки в діалоговому вікні
```

```
MessageBox.Show(tooltipText, "Підказка");
```

```
}
```

```
private void toolStripButton2_Click(object sender, EventArgs e)
```

```
{
```

```
// Визначаємо, яка саме кнопка була натиснута, використовуючи
параметр sender
```

```
ToolStripButton button = (ToolStripButton)sender;
```

```
// Отримуємо текст підказки для натиснутої кнопки
```

```
string tooltipText = button.ToolTipText;
```

```
// Виводимо текст підказки в діалоговому вікні
```

```
MessageBox.Show(tooltipText, "Підказка");
```

```
}
```

```
private void toolStripButton3_Click(object sender, EventArgs e)
```

```
{
```

```
// Визначаємо, яка саме кнопка була натиснута, використовуючи
параметр sender
```

```
ToolStripButton button = (ToolStripButton)sender;
```

```
// Отримуємо текст підказки для натиснутої кнопки
```

```
string tooltipText = button.ToolTipText;
```

```
// Виводимо текст підказки в діалоговому вікні
```

```
MessageBox.Show(tooltipText, "Підказка");
```

```
}
```

```

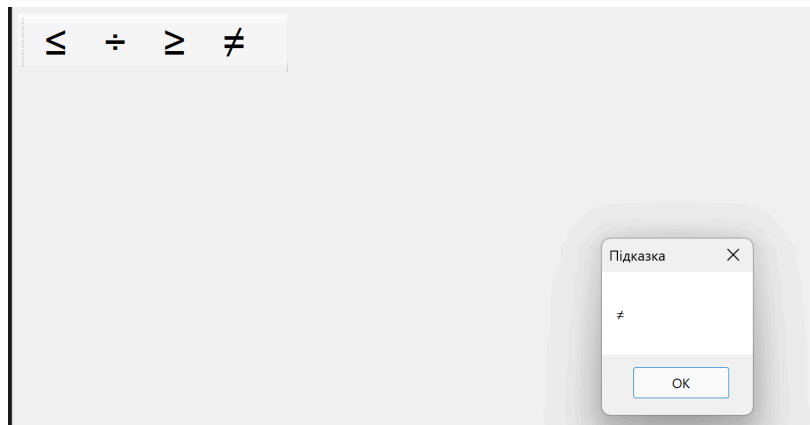
private void toolStripButton4_Click(object sender, EventArgs e)
{
    // Визначаємо, яка саме кнопка була натиснута, використовуючи
    параметр sender
    ToolStripButton button = (ToolStripButton)sender;

    // Отримуємо текст підказки для натиснутої кнопки
    string tooltipText = button.ToolTipText;

    // Виводимо текст підказки в діалоговому вікні
    MessageBox.Show(tooltipText, "Підказка");
}
}
}

```

Результат:



Завдання 10*

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

```

using System;
using System.Windows.Forms;

namespace Task_09
{
    public partial class Form1 : Form
    {

```

```

public Form1()
{
    InitializeComponent();
}

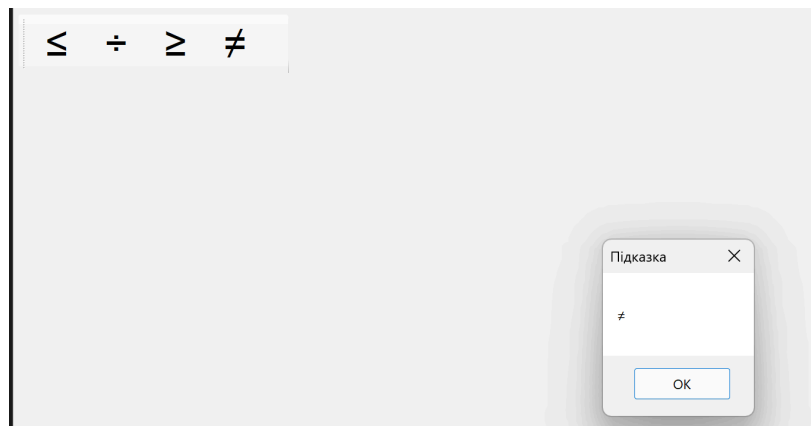
private void toolStripButton_Click(object sender, EventArgs e)
{
    // Визначаємо, яка саме кнопка була натиснута, використовуючи
    параметр sender
    ToolStripButton button = (ToolStripButton)sender;

    // Отримуємо текст підказки для натиснутої кнопки
    string tooltipText = button.ToolTipText;

    // Виводимо текст підказки в діалоговому вікні
    MessageBox.Show(tooltipText, "Підказка");
}
}
}

```

Результат:



Створити в проєкті форму збудь-яким компонентом, наприклад зі кнопкою (Button). Додати до форми не візуальний компонент Timer, для якого створити обробник подій, який перемикає форму застосунку чи її компонент між двома станами через час згідно варіанту

№ варіанту	Інтервал спрацьовування таймеру (с)	Стани для перемикання
1	2	3
2.	0,5	Напис в заголовку форми

```
namespace Task_11
{
    public partial class Form1 : Form
    {
        private bool isStateA = true;
        public Form1()
        {
            InitializeComponent();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            // Зміна стану та заголовок форми
            if (isStateA)
            {
                Text = "Стан А";
            }
            else
            {
                Text = "Стан В";
            }
        }
    }
}
```

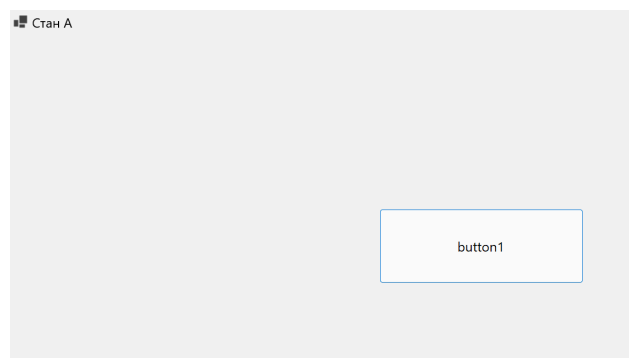
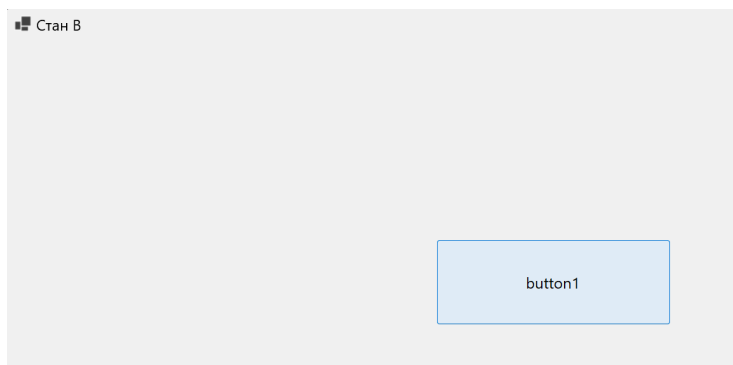
```

// Перемикаємо стан
isStateA = !isStateA;
}

private void button1_Click(object sender, EventArgs e)
{
    // Запускаємо або зупиняємо таймер при кожному натисканні кнопки
    if (timer1.Enabled)
    {
        timer1.Stop();
    }
    else
    {
        timer1.Start();
    }
}
}
}

```

Результат:



Завдання 12*

Виконати минуле завдання додавши компоненти налаштування (запуск, зупинка, зміна інтервалу спрацьовування) таймеру

```
namespace Task_12
{
    public partial class Form1 : Form
    {
        private bool isTimerRunning = false;
        private int timerInterval = 500;
        private bool isStateA = true;
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (isTimerRunning)
            {
                timer1.Stop();
                isTimerRunning = false;
                button1.Text = "Запустити таймер";
            }
            else
            {
                timer1.Start();
                isTimerRunning = true;
                button1.Text = "Зупинити таймер";
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if (int.TryParse(textBox1.Text, out int newInterval) && newInterval > 0)
            {
                timer1.Interval = newInterval;
            }
        }
    }
}
```

```

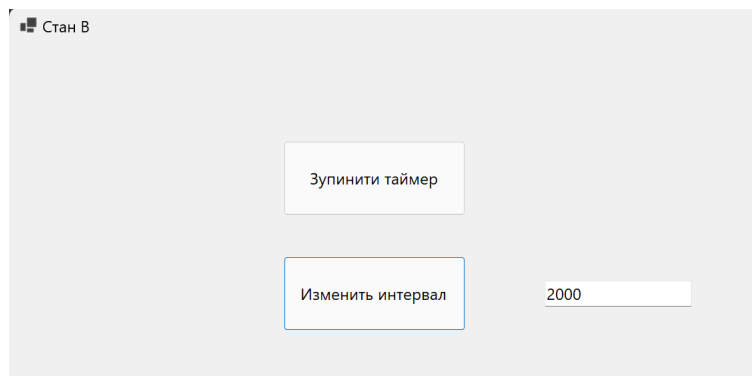
        timer1.Interval = newInterval;
        timerInterval = newInterval;
        textBox1.Text = timerInterval.ToString();
    }
    else
    {
        MessageBox.Show("Введіть коректний інтервал для таймера (ціле
число більше 0).");
    }
}

private void timer1_Tick(object sender, EventArgs e)
{
    // Зміна стану та заголовка форми
    if (isStateA)
    {
        Text = "Стан А";
    }
    else
    {
        Text = "Стан В";
    }

    // Перемикаємо стан
    isStateA = !isStateA;
}
}
}

```

Результат:



Висносок:

У цій лабораторній роботі було продемонстровано створення простої програми Windows Forms в середовищі розробки C#. Програма містить форму з кнопками та таймером, який керує зміною станів форми та її заголовку.

Було також додано компоненти налаштування для таймера, що дозволяють змінювати інтервал та керувати його роботою.

Під час виконання було продемонстровано базові концепції програмування інтерфейсів Windows Forms, такі як додавання компонентів на форму, робота з подіями та управління властивостями елементів у програмі.

Загалом, ця лабораторна робота дозволила поглибити розуміння роботи з формами Windows у C#, а також навичок роботи з елементами керування та обробки подій.

У цій лабораторній роботі ми навчилися встановлювати обробники подій для кнопок панелі інструментів в програмі Windows Forms і виводити тексти підказок на цих кнопках в окреме діалогове вікно (MsgBox). Ми розглянули два способи виконання цього завдання: використання окремого обробника подій для кожної кнопки і використання спільного обробника подій з використанням параметра `'sender'` для визначення, яка саме кнопка була натиснута.

Під час роботи ми навчилися працювати з властивістю `'ToolTipText'` для встановлення тексту підказки на кнопках і використовувати `'MessageBox.Show'` для виводу тексту підказки в діалоговому вікні. Ми також розглянули можливість встановлення спільного тексту для всіх кнопок та додавання номеру кнопки до тексту підказки для ідентифікації конкретної кнопки.

У цій лабораторній роботі ми отримали практичний досвід роботи з обробниками подій і властивостями кнопок у середовищі Windows Forms, що може бути корисним для розробки інтерфейсів користувача в програмах на платформі .NET.