

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

**Звіт**

**Лабораторна робота 10**

з дисципліни « C# .Net »

Виконав: студент групи КН-921Б

Бірюков Д .Є.

Перевірив:

Івашко А.В.

Харків 2023

## Зміст

Завдання 1(слайд шоу).....	1
Завдання 2(слайд шоу автоматично).....	3
Завдання 4(Вибір фото).....	7
Завдання 5(Фінансовий калькулятор).....	12
Завдання 6(Timer).....	13
Завдання 7(NumericUpDown).....	16
Завдання 8(StatusStrip).....	19
Завдання 9(NotifyIcon).....	20
Висновок.....	21

## Завдання 1

Створити застосунок, який реалізує слайд-шоу зображень з галузі згідно до власного варіанту

№ варіанту	Галузь для фото до слайд-шоу
2.	Комп'ютери

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.IO;
```

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

```
namespace PhotoViewerApp
```

```
{
```

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

```
    {
```

```
        private List<string> photoPaths;
```

```
        private int currentPhotoIndex = 0;
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

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

```
        {
```

```
            // Открываем диалоговое окно выбора папки для фотографий
```

```
            FolderBrowserDialog folderDialog = new FolderBrowserDialog();
```

```
            if (folderDialog.ShowDialog() == DialogResult.OK)
```

```
            {
```

```
                // Получаем список файлов из выбранной папки
```

```
                string selectedFolderPath = folderDialog.SelectedPath;
```

```
                string[] photoFiles = Directory.GetFiles(selectedFolderPath, "*.jpg"); //
```

Можно изменить расширения файлов по вашему усмотрению

```
// Загружаем список фотографий
```

```
photoPaths = new List<string>(photoFiles);
```

```
currentPhotoIndex = 0;
```

```
if (photoPaths.Count > 0)
```

```
{
```

```
    DisplayCurrentPhoto();
```

```
}
```

```
else
```

```
{
```

```
    MessageBox.Show("В выбранной папке нет фотографий.");
```

```
}
```

```
}
```

```
}
```

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

```
{
```

```
    if (photoPaths != null && photoPaths.Count > 0)
```

```
    {
```

```
        currentPhotoIndex = (currentPhotoIndex + 1) % photoPaths.Count;
```

```
        DisplayCurrentPhoto();
```

```
    }
```

```
}
```

```
private void button3_Click(object sender, EventArgs e)
```

```
{
```

```
    if (photoPaths != null && photoPaths.Count > 0)
```

```
    {
```

```
        currentPhotoIndex = (currentPhotoIndex - 1 + photoPaths.Count) %  
photoPaths.Count;
```

```
        DisplayCurrentPhoto();
```

```
    }
```

```
}
```

```
private void DisplayCurrentPhoto()
```

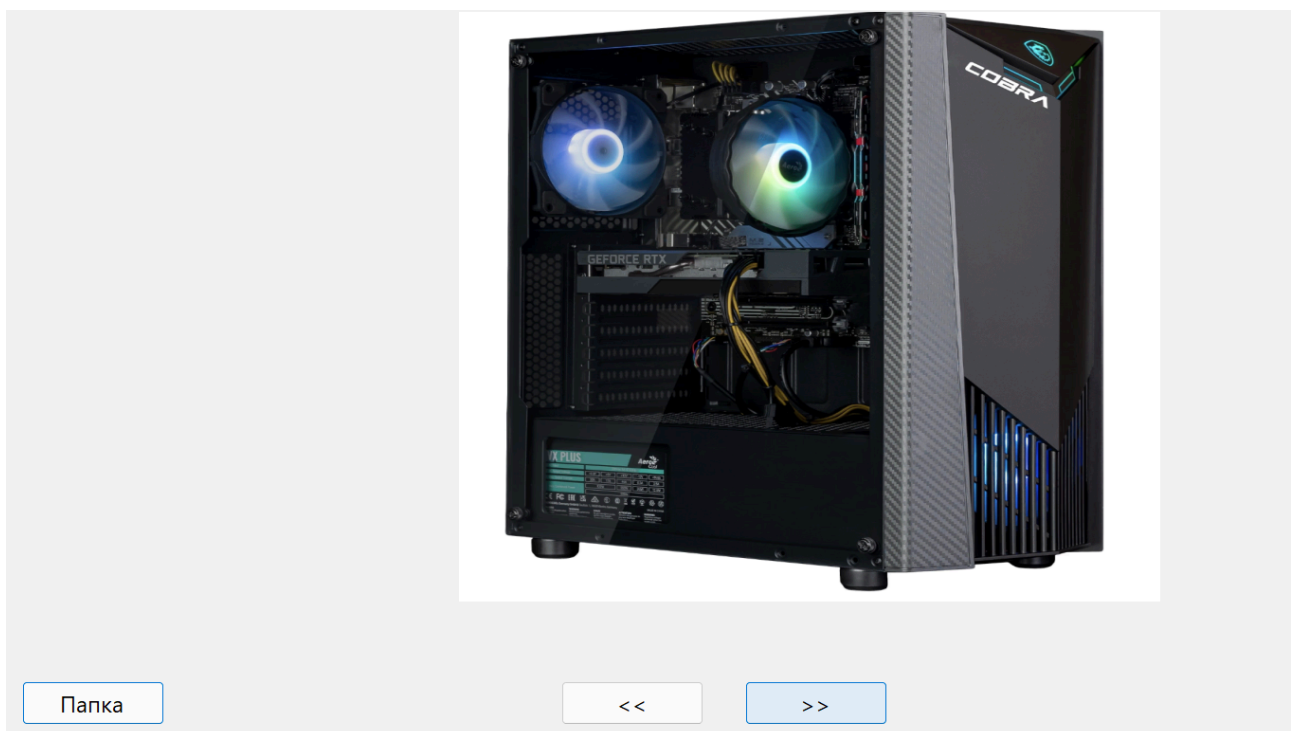
```
{
```

```

if (photoPaths != null && photoPaths.Count > 0)
{
    string currentPhotoPath = photoPaths[currentPhotoIndex];
    pictureBox1.Image = new System.Drawing.Bitmap(currentPhotoPath);
}
}
}
}

```

Результат:



## Завдання 2\*

Виконати минуле завдання, додавши до форми новий(1) компонент(и) та обробники їх подій, що дозволять автоматично демонструвати, зупиняти та налаштовувати швидкість демонстрації фото

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Windows.Forms;

```

```

namespace WinFormsApp1
{
    public partial class Form1 : Form
    {
        private List<string> photoPaths;
        private int currentPhotoIndex = 0;
        private bool isSlideshowRunning = false;
        private int slideshowInterval = 2000; // Интервал слайд-шоу в миллисекундах

        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // Открываем диалоговое окно выбора папки для фотографий
            FolderBrowserDialog folderDialog = new FolderBrowserDialog();

            if (folderDialog.ShowDialog() == DialogResult.OK)
            {
                // Получаем список файлов из выбранной папки
                string selectedFolderPath = folderDialog.SelectedPath;
                string[] photoFiles = Directory.GetFiles(selectedFolderPath, "*.png"); //
                Можно изменить расширения файлов по вашему усмотрению

                // Загружаем список фотографий
                photoPaths = new List<string>(photoFiles);
                currentPhotoIndex = 0;

                if (photoPaths.Count > 0)
                {
                    DisplayCurrentPhoto();
                }
                else
                {

```

```

        MessageBox.Show("В выбранной папке нет фотографий.");
    }
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    if (photoPaths != null && photoPaths.Count > 0)
    {
        currentPhotoIndex = (currentPhotoIndex + 1) % photoPaths.Count;
        DisplayCurrentPhoto();
    }
}

```

```

private void button3_Click(object sender, EventArgs e)
{
    if (photoPaths != null && photoPaths.Count > 0)
    {
        currentPhotoIndex = (currentPhotoIndex - 1 + photoPaths.Count) %
photoPaths.Count;
        DisplayCurrentPhoto();
    }
}

```

```

private void DisplayCurrentPhoto()
{
    if (photoPaths != null && photoPaths.Count > 0)
    {
        string currentPhotoPath = photoPaths[currentPhotoIndex];
        pictureBox1.Image = new System.Drawing.Bitmap(currentPhotoPath);
    }
}

```

```

private void button4_Click(object sender, EventArgs e)
{
    if (photoPaths != null && photoPaths.Count > 0)

```

```

{
    if (isSlideshowRunning)
    {
        // Останавливаем слайд-шоу
        isSlideshowRunning = false;
        button4.Text = "Запустить слайд-шоу";
        timer1.Stop();
    }
    else
    {
        // Запускаем слайд-шоу
        isSlideshowRunning = true;
        button4.Text = "STOP";
        StartSlideshow();
    }
}
}

```

```
private void button5_Click(object sender, EventArgs e)
```

```

{
    if (isSlideshowRunning)
    {
        // Останавливаем слайд-шоу, чтобы изменить скорость
        isSlideshowRunning = false;
        button4.Text = "PLAY";
        timer1.Stop();
    }
}

```

```
// Изменяем скорость слайд-шоу
```

```
if (slideshowInterval == 2000)
```

```

{
    slideshowInterval = 1000; // Увеличьте или уменьшите интервал, чтобы

```

изменить скорость

```
    button5.Text = "x2";
```

```

}

```

```
else
```



```

{
    slideshowInterval = 2000;
    button5.Text = "x1";
}
}

```

```

private void StartSlideshow()
{
    if (photoPaths != null && photoPaths.Count > 0)
    {
        timer1 = new System.Windows.Forms.Timer();
        timer1.Tick += new EventHandler(SlideshowTimer_Tick);
        timer1.Interval = slideshowInterval;
        timer1.Start();
    }
}

```

```

private void SlideshowTimer_Tick(object sender, EventArgs e)
{
    if (photoPaths != null && photoPaths.Count > 0)
    {
        currentPhotoIndex = (currentPhotoIndex + 1) % photoPaths.Count;
        DisplayCurrentPhoto();
    }
}

}
}

```

Результат



## Завдання 4

Створити застосунок, який реалізує перегляд зображень згідно до варіанту з першого завдання.

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

            private void button1_Click(object sender, EventArgs e)
            {
                using (FolderBrowserDialog folderDialog = new FolderBrowserDialog())
                {
                    if (folderDialog.ShowDialog() == DialogResult.OK)
                    {
                        string selectedFolderPath = folderDialog.SelectedPath;

                        // Отображаем путь к выбранной папке в Label
                        label1.Text = "" + selectedFolderPath;

                        // Очищаем ListBox и PictureBox
                        listBox1.Items.Clear();
                        pictureBox1.Image = null;

                        // Загружаем изображения из выбранной папки и добавляем их пути
                        в ListBox
                        string[] imageFiles = Directory.GetFiles(selectedFolderPath, "*.png");
                        foreach (string imagePath in imageFiles)
                        {
                            listBox1.Items.Add(imagePath);
                        }
                    }
                }
            }
        }
    }
}
```

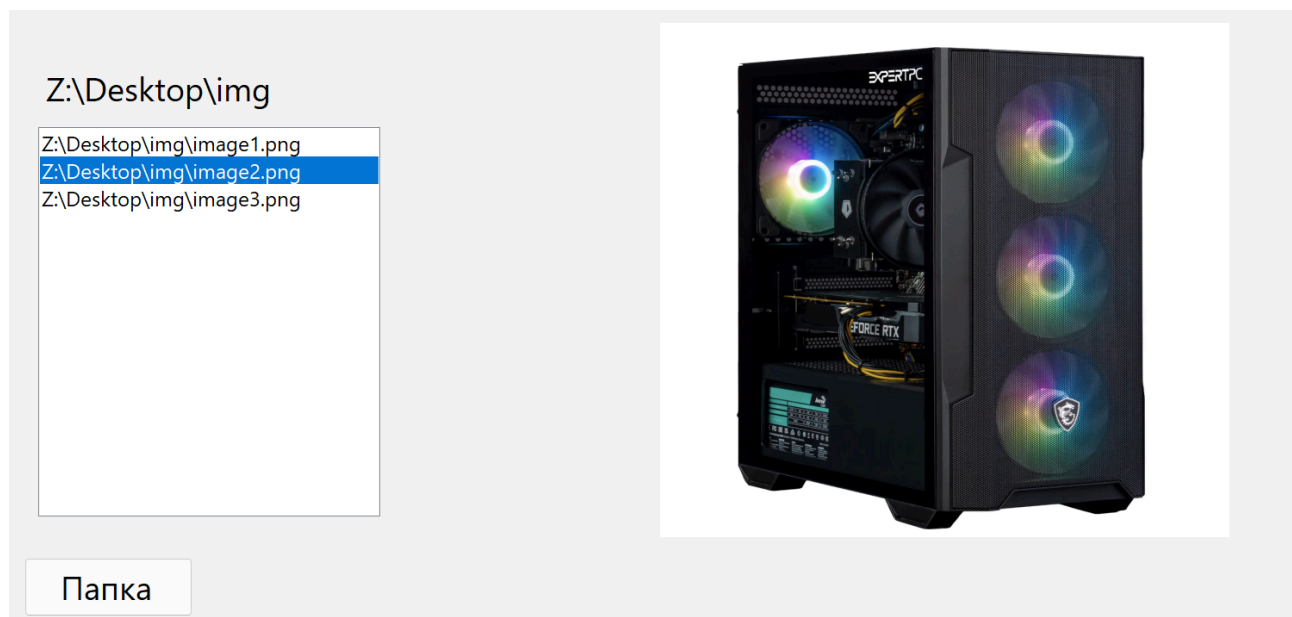
```

    }

    private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (listBox1.SelectedIndex != -1)
        {
            // При выборе элемента из ListBox, отображаем соответствующее
            изображение в PictureBox
            pictureBox1.ImageLocation = listBox1.SelectedItem.ToString();
        }
    }
}

```

Результат:



## Завдання 5

Створити застосунок, який реалізує розрахунок щомісячного графіку фінансових виплат по кредиту зі завданими параметрами.

```
namespace Task_05
```

```
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            // Додайте колонки до ListView
            listView1.View = View.Details;
            listView1.Columns.Add("Місяць", 200);
            listView1.Columns.Add("Борг", 200);
            listView1.Columns.Add("Відсоток", 200);
            listView1.Columns.Add("Платіж", 200);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(textBox1.Text) ||
string.IsNullOrEmpty(textBox2.Text) || string.IsNullOrEmpty(textBox3.Text))
            {
                MessageBox.Show("Будь ласка, заповніть всі поля.");
                return;
            }

            double loanAmount = double.Parse(textBox1.Text);
            int loanTermMonths = int.Parse(textBox2.Text);
            double annualInterestRate = double.Parse(textBox3.Text);

            listView1.Items.Clear();

            double monthlyInterestRate = annualInterestRate / 12 / 100;
            double monthlyPayment = loanAmount * (monthlyInterestRate / (1 -
Math.Pow(1 + monthlyInterestRate, -loanTermMonths)));
```

```

for (int month = 1; month <= loanTermMonths; month++)
{
    double interestPayment = loanAmount * monthlyInterestRate;
    double principalPayment = monthlyPayment - interestPayment;

    ListViewItem item = new ListViewItem(month.ToString());
    item.SubItems.Add(loanAmount.ToString("N2"));
    item.SubItems.Add(interestPayment.ToString("N2"));
    item.SubItems.Add(monthlyPayment.ToString("N2"));

    listView1.Items.Add(item);

    loanAmount -= principalPayment;
}
}
}
}

```

Результат:

Сума:	<input type="text" value="5000"/>
Термін(міс):	<input type="text" value="3"/>
Відсоткова ставка:	<input type="text" value="2"/>
<input type="button" value="OK"/>	

Місяць	Борг	Відсоток	Платіж
1	5 000,00	8,33	1 672,23
2	3 336,11	5,56	1 672,23
3	1 669,44	2,78	1 672,23

## Завдання 6

Створити застосунок, який реалізує роботу класичного секундоміра з відображенням хвилин та секунд. Кожні п'ять секунд змінювати чи відновлювати властивості деяких компонентів на формі згідно до свого варіанту

№ варіанту	Змінювана властивість
2.	Колір фону форми

```
using System.Windows.Forms;
namespace WinFormsApp2
{
    public partial class Form1 : Form
    {
        private int seconds = 0;
        private bool isRunning = false;

        public Form1()
        {
            InitializeComponent();
            // Налаштування таймера
            timer.Interval = 1000; // Оновлення кожну секунду
            timer.Tick += Timer_Tick;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (!isRunning)
            {
                isRunning = true;
                timer.Start();
                button1.Enabled = false;
                button2.Enabled = true;
            }
        }
    }
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    if (isRunning)
    {
        isRunning = false;
        timer.Stop();
        button1.Enabled = true;
        button2.Enabled = false;
    }
}
```

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

```
private void Timer_Tick(object sender, EventArgs e)
{
    seconds++;

    if (seconds % 5 == 0)
    {
        ChangeFormBackgroundColor();
    }

    UpdateTimerDisplay();
}
```

```
private void UpdateTimerDisplay()
{
    TimeSpan time = TimeSpan.FromSeconds(seconds);
    label1.Text = time.ToString(@"hh\:mm\:ss");
}
```

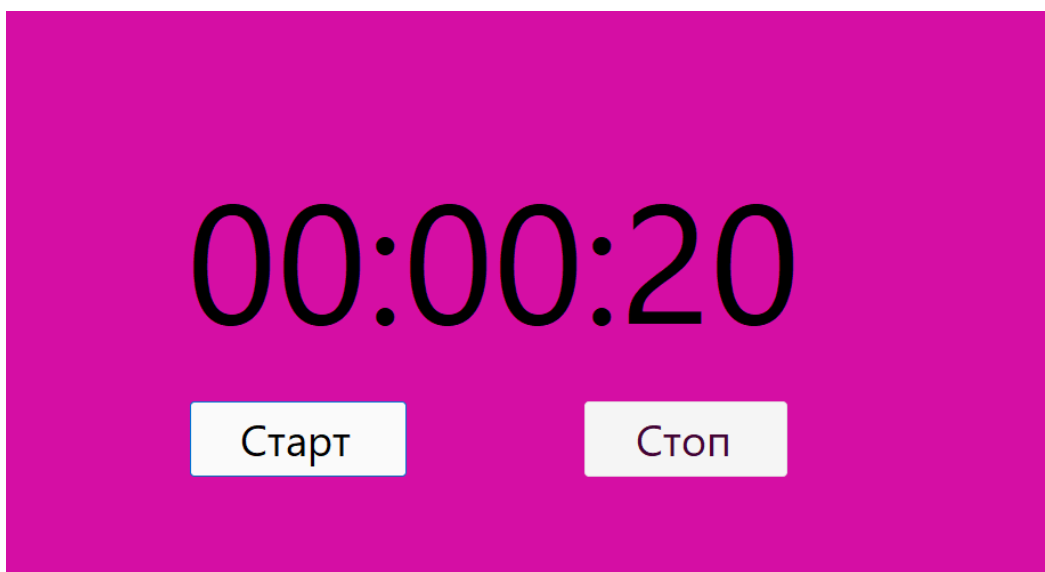
```
private void ChangeFormBackgroundColor()
```

```

{
    Random random = new Random();
    Color randomColor = Color.FromArgb(random.Next(256),
random.Next(256), random.Next(256));
    panel1.BackColor = randomColor;
}
}
}

```

### Результат



### Завдання 7\*

Модифікувати минуле завдання, додавши до застосунку можливість змінювати хвилинний та секундний інтервали роботи секундоміра, використавши наступний зразок форми та компонент `NumericUpDown`

```

using System.Windows.Forms;
namespace WinFormsApp2
{
    public partial class Form1 : Form
    {

        private int seconds = 0;
        private bool isRunning = false;
    }
}

```



```

public Form1()
{
    InitializeComponent();
    // Налаштування таймера
    timer.Interval = 1000; // Оновлення кожну секунду
    timer.Tick += Timer_Tick;
}

private void button1_Click(object sender, EventArgs e)
{
    if (!isRunning)
    {
        isRunning = true;
        button1.Text = "Стоп";
        int minutes = (int)numericUpDown1.Value;
        int seconds = (int)numericUpDown2.Value;

        // Обчислення загального часу в секундах
        this.seconds = minutes * 60 + seconds;
        timer.Start();
    }
    else
    {
        isRunning = false;
        button1.Text = "Старт";
        timer.Stop();
    }
}

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

private void Timer_Tick(object sender, EventArgs e)

```

```

{
    seconds++;

    if (seconds % 5 == 0)
    {
        ChangeFormBackgroundColor();
    }

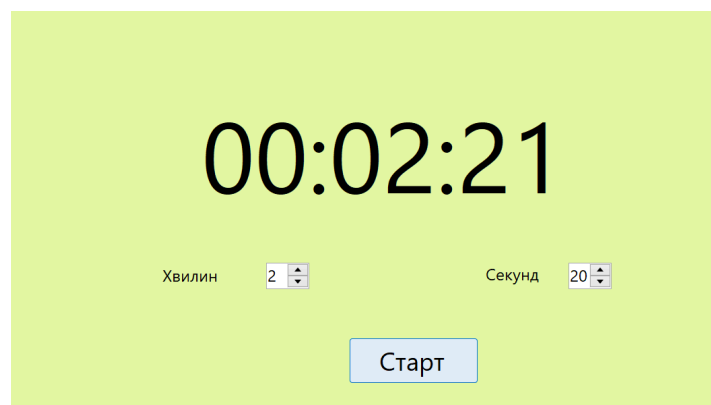
    UpdateTimerDisplay();
}

private void UpdateTimerDisplay()
{
    TimeSpan time = TimeSpan.FromSeconds(seconds);
    label1.Text = time.ToString(@"hh\:mm\:ss");
}

private void ChangeFormBackgroundColor()
{
    Random random = new Random();
    Color randomColor = Color.FromArgb(random.Next(256),
random.Next(256), random.Next(256));
    panel1.BackColor = randomColor;
}
}
}

```

Результат:



## Завдання 8

Створити застосунок, який змінює вміст міток (toolStripStatusLabel1, toolStripStatusLabel2) рядка статусу (StatusStrip1) згідно до свого варіанту під час друку чергового символу у текстовому полі

№ варіанту	Вміст toolStripStatusLabel1	Вміст toolStripStatusLabel2 з текстового поля
2.	Теперішній час (24- годинний формат)	Кількість букв

```
namespace Task_08
```

```
{
```

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

```
    {
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        private void textBox1_TextChanged(object sender, EventArgs e)
```

```
        {
```

```
            // Оновлюємо toolStripStatusLabel2 згідно кількості букв у textBox1
```

```
            toolStripStatusLabel2.Text = $"Кількість букв: {textBox1.Text.Length}";
```

```
            // Оновлюємо toolStripStatusLabel1 і встановлюємо поточний час (24-годинний формат)
```

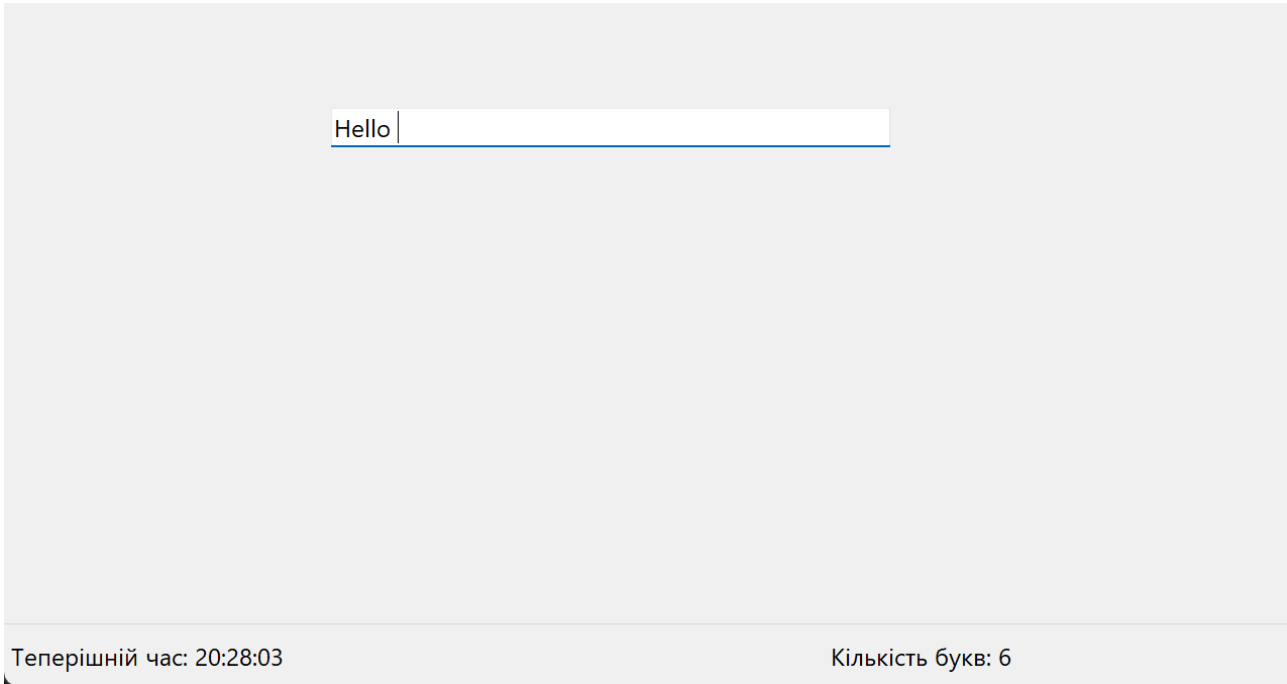
```
            toolStripStatusLabel1.Text = $"Теперішній час: {DateTime.Now.ToString("HH:mm:ss")}";
```

```
        }
```

```
    }
```

```
}
```

Результат :



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

```
namespace Task_08
```

```
{
```

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

```
    {
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
            // Инициализируем NotifyIcon
```

```
            notifyIcon1 = new NotifyIcon();
```

```
            notifyIcon1.Icon = SystemIcons.Application; // Устанавливаем значок
```

```
            notifyIcon1.Text = "Моя програма"; // Устанавливаем текст уведомления
```

```
            // Подключаем контекстное меню к NotifyIcon
```

```
            notifyIcon1.ContextMenuStrip = contextMenuStrip1;
```

// Додаємо обробник події на подвійний клік по значку, щоб показувати або ховати форму

```
notifyIcon1.DoubleClick += ToggleFormVisibility;
}
```

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

```
{
    if (WindowState == FormWindowState.Minimized)
    {
        // Якщо форма згорнута, показуємо її
        Show();
        WindowState = FormWindowState.Normal;
    }
    else
    {
        // Якщо форма вже видима, ховуємо її
        Hide();
        WindowState = FormWindowState.Minimized;
    }
}
```

```
private void textBox1_TextChanged(object sender, EventArgs e)
```

```
{
    // Видаляємо пробіли і оновлюємо toolStripStatusLabel2 згідно кількості символів без пробілів у textBox1
```

```
    string textWithoutSpaces = textBox1.Text.Replace(" ", "");
    toolStripStatusLabel2.Text = $"Кількість символів (без пробілів): {textWithoutSpaces.Length}";
```

```
    // Оновлюємо toolStripStatusLabel1 і встановлюємо поточний час (24-годинний формат)
```

```
    toolStripStatusLabel1.Text = $"Теперішній час: {DateTime.Now.ToString("HH:mm:ss")}";
}
```

```
private void ShowForm()
```

```
{
    Show();
}
```

```

WindowState = FormWindowState.Normal;
}

private void HideForm()
{
    Hide();
    WindowState = FormWindowState.Minimized;
}

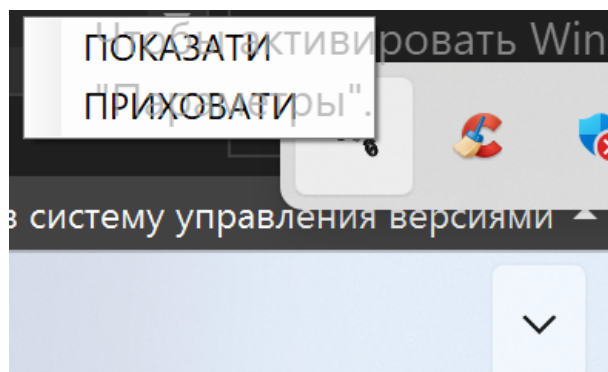
private void ToolStripMenuItem1_Click(object sender, EventArgs e)
{
    ShowForm();
}

private void ToolStripMenuItem2_Click(object sender, EventArgs e)
{
    HideForm();
}

}
}

```

Результат



Висновок: У цій лабораторній роботі було створено Windows Forms додаток, який використовував елемент повідомлень `NotifyIcon` для згортання та розгортання вікна програми у системний трей, а також для відображення контекстного меню. Користувач міг двічі клацнути на значку `NotifyIcon`, щоб показати або приховати вікно програми, і використовувати контекстне меню для виконання додаткових дій.

Основні кроки у створенні цієї програми включають:

1. Ініціалізацію `NotifyIcon`, встановлення його значка та тексту повідомлення.
2. Підключення контекстного меню до `NotifyIcon` для виконання додаткових дій.
3. Додавання обробника подій подвійного клацання по значку `NotifyIcon`, щоб показувати або приховувати вікно програми.
4. Оновлення статусних рядків та відображення поточного часу у форматі "HH:mm:ss" під час введення тексту в текстове поле `textBox1`.
5. Додавання обробника події закриття форми для приховування вікна та відображення `NotifyIcon` у треї, замість завершення роботи програми.

Таким чином, цей додаток демонструє використання `NotifyIcon` для створення зручного інтерфейсу для взаємодії з користувачем та управління додатком у режимі згорнутого вікна.