МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

факультет програмної інженерії та бізнесу

кафедра інженерії програмного забезпечення

**Практична робота № 1**

з дисципліни « Архітектура та проектування програмного забезпечення .Net »

на тему: « Розробка компонентів »

Виконав: студент 2 курсу групи № 621п

121 «Інженерія програмного забезпечення»

(код спеціальності)

\_\_\_\_ \_Бахшалієв А. Е.\_\_\_ \_\_\_\_\_\_

(ПІБ студента)

Прийняв: ст. викладач каф. 603

\_\_\_\_\_\_Сьомочкін М. О., Дем’яненко В. А.\_\_\_\_\_\_

(ПІБ викладача)

Національна шкала:

Кількість балів:

Харків – 2024

**ЗМІСТ**

[**ПОСТАНОВКА ЗАДАЧІ** 3](#_Toc160481951)

[**ВИКОНАННЯ РОБОТИ** 3](#_Toc160481952)

[Теоретичний матеріал 3](#_Toc160481953)

[Графічний матеріал 3](#_Toc160481954)

[Лістинг програми 5](#_Toc160481955)

[**ВИСНОВОК** 8](#_Toc160481956)

# **ПОСТАНОВКА ЗАДАЧІ**

**Мета роботи:** Вивчення особливостей створення та використання компонентів та елементів інтерфейсу користувача за допомогою середовища розробки *Visual Studio*, закріплення навичок об'єктно-орієнтованого програмування.

**Завдання**

**Варіант 2:** TextBox для введення цілих чисел із різними кольорами позитивних, негативних і нульового значень.

# **ВИКОНАННЯ РОБОТИ**

## Теоретичний матеріал

Зазвичай, для створення власного компонента на C# необхідно створити свій клас, що є похідним від однієї з базових класів. Базовий клас вибирається на основі вимог до рівня функціональної складності інтерфейсу компонента що розроблюється. У середовищі розробки Visual Studio пропонуються такі стандартні базові класи платформи .Net для створення власних компонентів:

Component – для створення невізуальних (без інтерфейсу користувача) компонентів з мінімальною стандартною функціональністю.

Control – для створення нестандартних візуальних компонентів, у яких зовнішній вигляд (зображення) повністю контролює та реалізує розробник.

UserControl – контейнер для об'єднання та управління існуючих елементів управління як єдиного компонента та по суті є аналогом форми (клас Form). Таким чином використання даного класу як базового дозволяє розширити функціональність компонента, що розробляється як за рахунок успадкування, так і за рахунок агрегації з іншими компонентами.

Крім перерахованих вище як базовий клас може виступати практично будь-який з елементів управління, що знаходяться на панелі інструментів (Toolbox), тому що всі вони спадкоємці як мінімум класу Component.

При створенні компонента слід враховувати, що для повторного використання в інших проектах необхідно розмістити код в окремій бібліотеці (Class Library), яка додається до панелі інструментів.

## Графічний матеріал

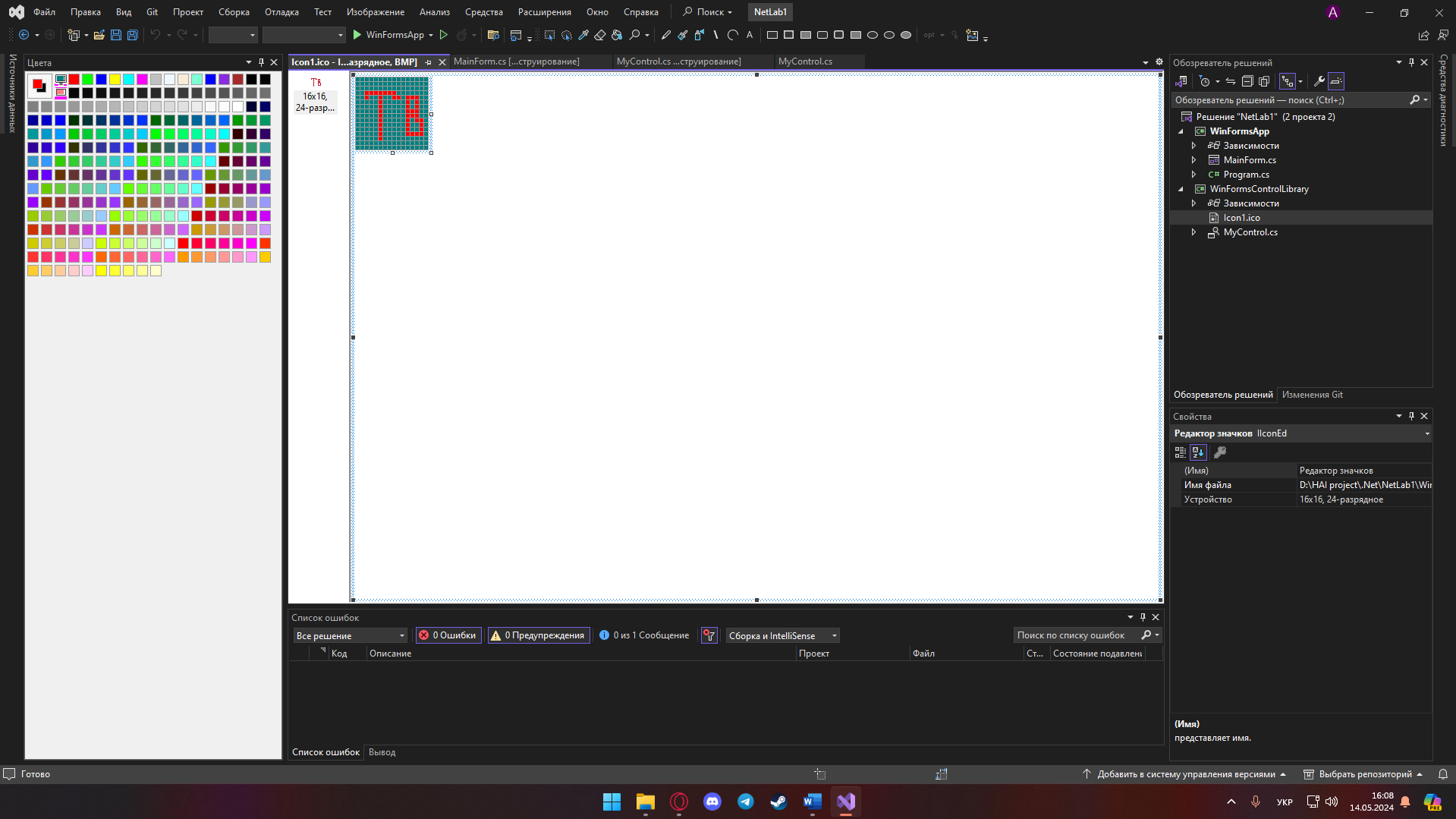


Рисунок 1 – Створена піктограма

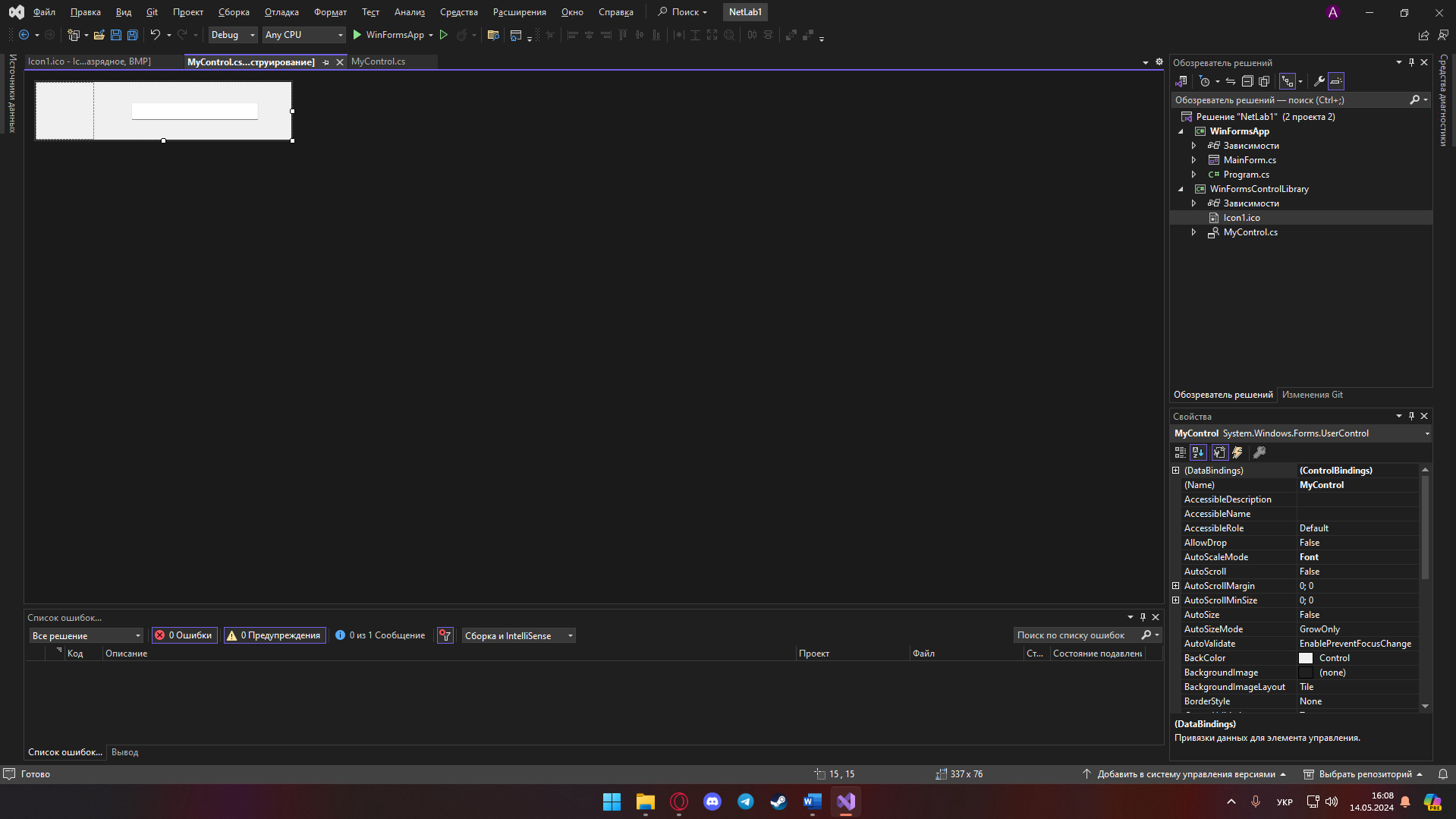


Рисунок 2 – Створення свого компоненту

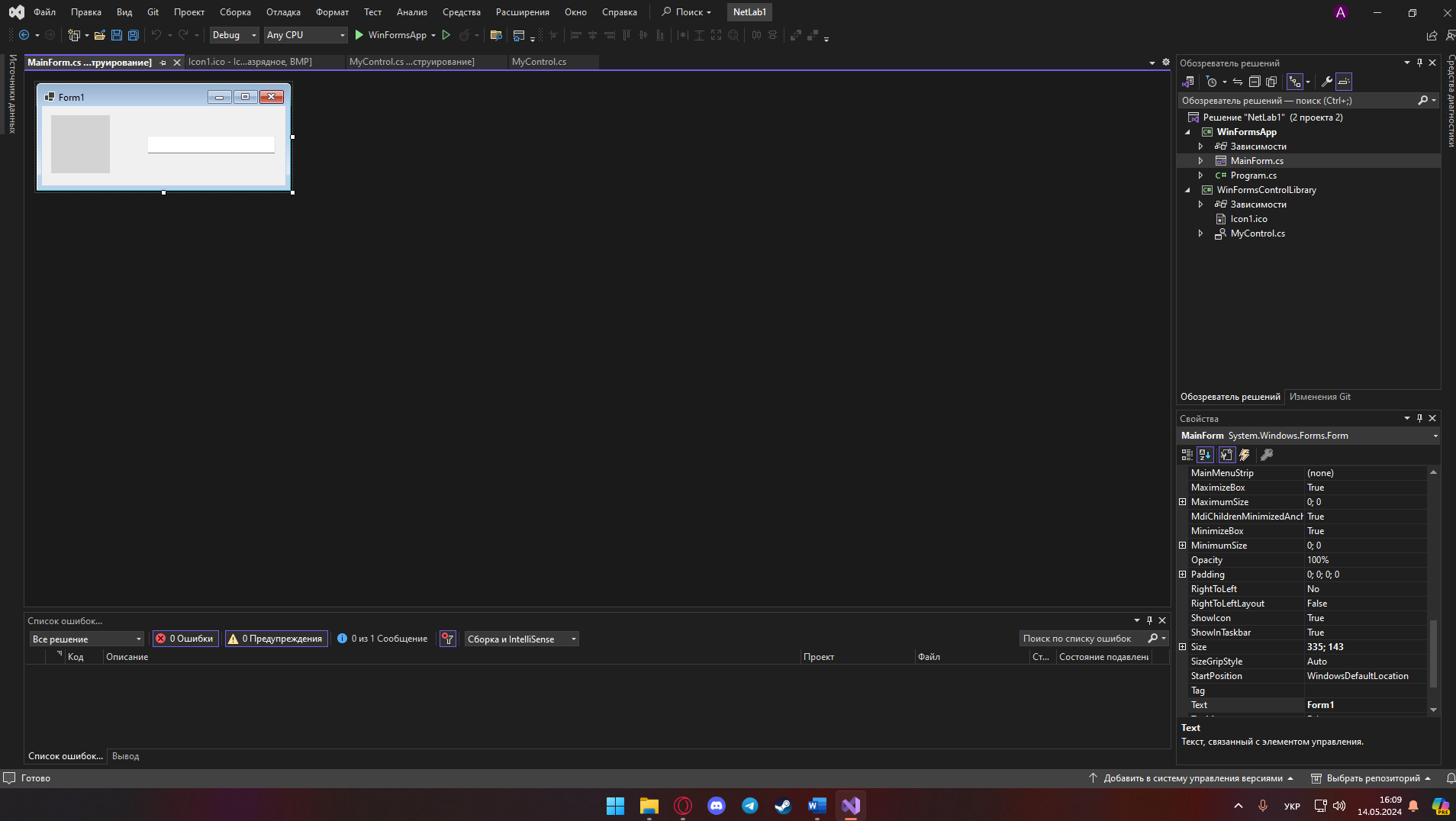


Рисунок 3 – Розміщення свого компоненту на формі

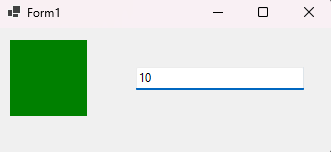


Рисунок 4 – Приклад роботи програми №1

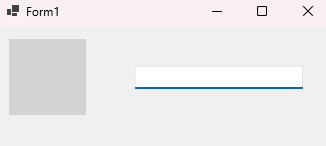


Рисунок 5 – Приклад роботи програми №2

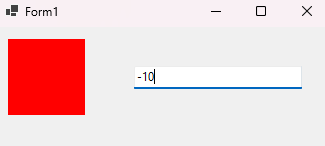


Рисунок 6 – Приклад роботи програми №3

## Лістинг програми

**MyControl.cs**

using System;

using System.Drawing;

using System.Windows.Forms;

namespace WinFormsControlLibrary

{

[ToolboxBitmap(@"D:\HAI project\.Net\NetLab1\WinFormsControlLibrary\Icon1.ico")]

public partial class MyControl : UserControl

{

public MyControl()

{

InitializeComponent();

led.BackColor = Color.LightGray;

}

private void textBox\_TextChanged(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(textBox.Text))

{

led.BackColor = Color.LightGray;

}

else

{

if (int.TryParse(textBox.Text, out int number))

{

if (number > 0)

led.BackColor = Color.Green;

if (number < 0)

led.BackColor = Color.Red;

}

else

{

led.BackColor = Color.LightGray;

}

}

}

}

}

**MainForm.cs**

namespace WinFormsApp

{

public partial class MainForm : System.Windows.Forms.Form

{

public MainForm()

{

InitializeComponent();

}

private void MainForm\_Load(object sender, EventArgs e)

{

}

}

}

# **ВИСНОВОК**

Створено додаток, який складається з двох програмних одиниць – бібліотеки з компонентом та файлу, що запускається та використовує компонент із бібліотеки. Розроблено компонент шляхом об'єднання декількох стандартних компонентів в один власний. Оформлено звіт.