

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №1
з дисципліни
«Об'єктно-орієнтоване програмування»

Виконав: студент групи ІМ-42
Максим Крамаренко Юрійович
номер варіанту: 17

Перевірив:
Молчанова А. А.

Варіанти завдань та основні вимоги

Для усіх варіантів необхідно створити два пункти меню – "Робота1" та "Робота2".

1. Потрібно, щоб при виборі пункту меню "Робота1" виконувалося

щось згідно **варіанту 1**:

Вікно діалогу з повзуном горизонтального скролінгу

(Horizontal scroll Bar) та дві кнопки: [Так] і [Відміна].

Рухаючи повзунок скролінгу користувач вводить число у діапазоні від 1 до 100. Після натискування кнопки [Так] вибране число буде відображатися у головному вікні.

2. Запрограмувати також, щоб при виборі пункту меню "Робота2" виконувалося щось згідно **варіанту 2**:

Два вікна діалогу. Спочатку з'являється перше, яке має дві кнопки: [Далі >] і [Відміна]. Якщо натиснути кнопку [Далі >],

то воно закриється і з'явиться друге дlg вікно, яке має кнопки: [< Назад], [Так] і [Відміна]. Якщо натиснути кнопку [<Назад], вікно закриється і перехід до першого вікна.

3. Вихідний текст функції 1 запрограмувати в окремому модулі, а вихідний текст для функції 2 – у іншому окремому модулі. Таким чином, проєкт повинен складатися з головного файлу Lab2.cpp та ще, як мінімум, двох файлів модулів з іменами Module1.cs та Module2.cs

Текст програми:

MainWindow.axaml.cs:

```
using Avalonia.Controls;
using Module1;
using Module2;

namespace Lab1;

public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();

        private async void Work1_Click(object? sender,
Avalonia.Interactivity.RoutedEventArgs e)
        {
            int selectedNumber = await Work1.OnClick(this);
            if (selectedNumber == -1) return;
            SelectedNumberTextBlock.Text = $"Вибране число: {selectedNumber}";
        }

        private void Work2_Click(object? sender, Avalonia.Interactivity.RoutedEventArgs
e)
        {
            Work2.OnClick(this);
        }
    }
}
```

Module1:

Work1.cs:

```
using Avalonia.Controls;
using System.Threading.Tasks;
namespace Module1;

public static class Work1
{
    public async static Task<int> OnClick(Window parent)
    {
        var dialog = new Dialog1();
        var result = await dialog.ShowDialog<bool>(parent);
        if (result)
        {
            return dialog.SelectedValue;
        }
    }
}
```

```

    }

    return -1;

}
}

```

Dialog1.cs:

```

using Avalonia.Controls;
using Avalonia.Interactivity;

namespace Module1
{
    public partial class Dialog1 : Window
    {
        public Dialog1()
        {
            InitializeComponent();

            public int SelectedValue =>
(int)this.FindControl<Slider>("NumberSlider")!.Value;

            private void OkButton_Click(object? sender, RoutedEventArgs e)
            {
                Close(true);
            }

            private void CancelButton_Click(object? sender, RoutedEventArgs e)
            {
                Close(false);
            }
        }
    }
}

```

Module2:

Work2.cs:

```

using Avalonia.Controls;

namespace Module2;

public static class Work2
{
    public static int OnClick(Window parent)
    {
        var dialog = new Dialog2();
    }
}

```

```

        dialog.Show();
        return 1;
    }
}

```

Dialog2.cs:

```

using Avalonia.Controls;
namespace Module2;

public partial class Dialog2 : Window
{
    public Dialog2()
    {
        InitializeComponent();

        NextButton.Click += (_, __) =>
        {
            var dialog3 = new Dialog3();
            dialog3.Show();
            Close();
        };

        CancelButton.Click += (_, __) => Close();
    }
}

```

Dialog3.cs:

```

using Avalonia.Controls;
namespace Module2;

public partial class Dialog3 : Window
{
    public Dialog3()
    {
        InitializeComponent();

        OkButton.Click += (_, __) => Close();
        PreviousButton.Click += (_, __) => ShowDialog2();
        CancelButton.Click += (_, __) => ShowDialog2();
    }

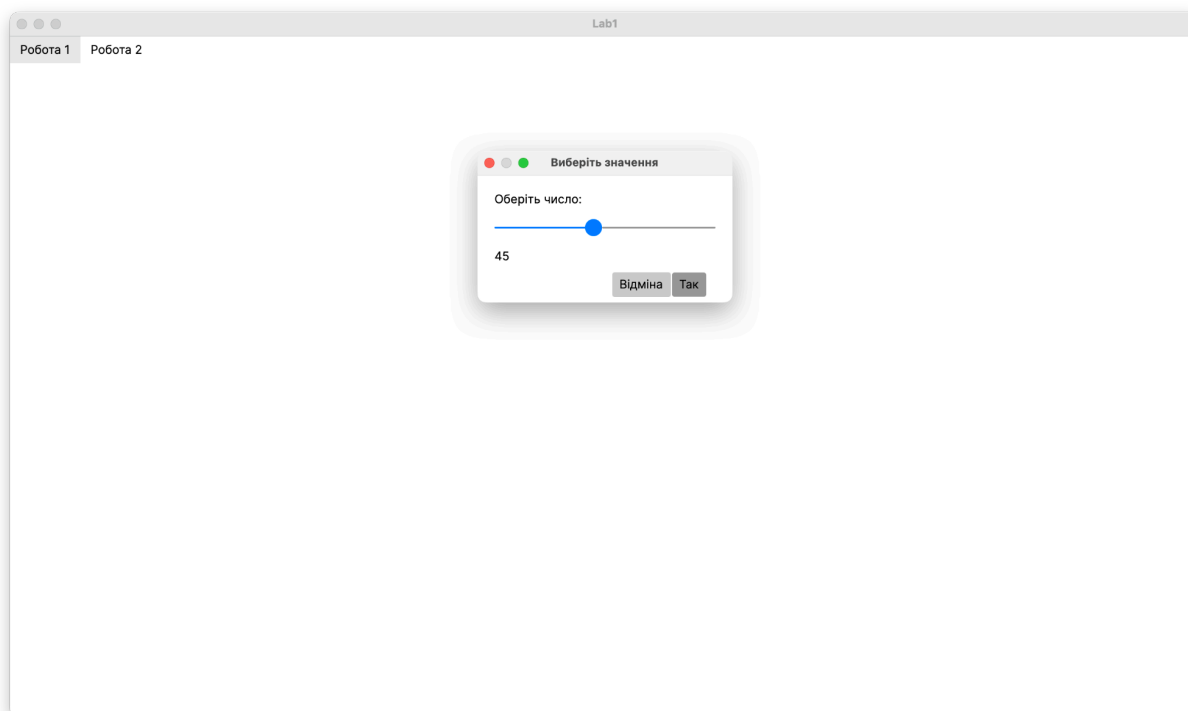
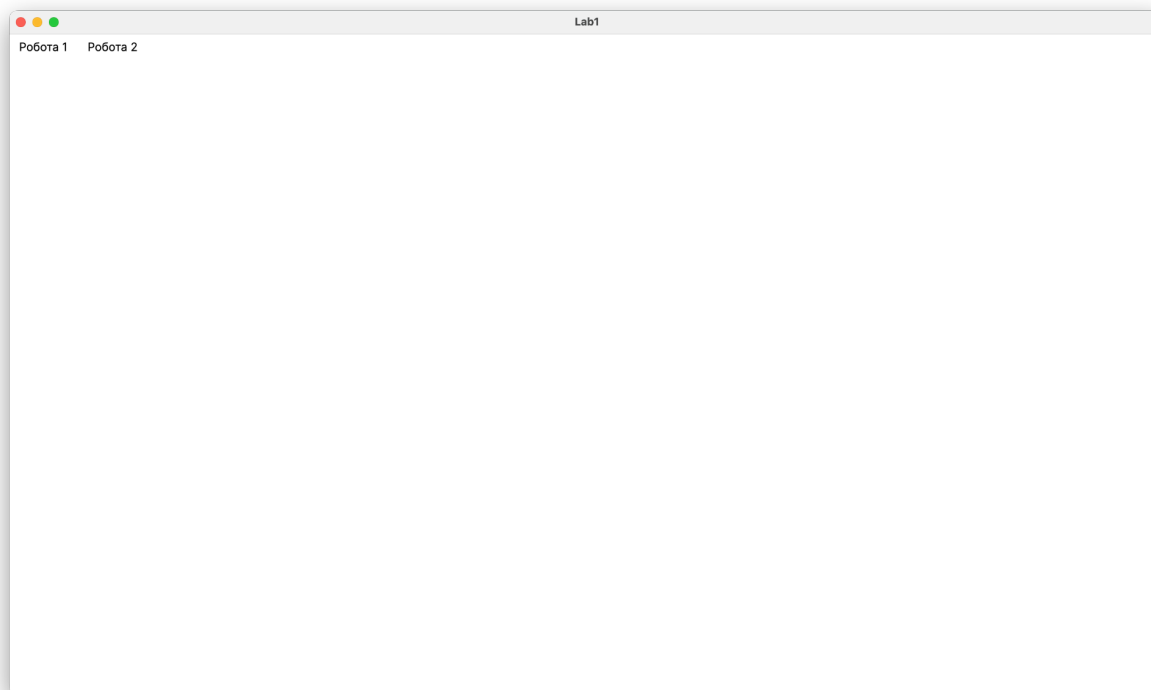
    private void ShowDialog2()
    {

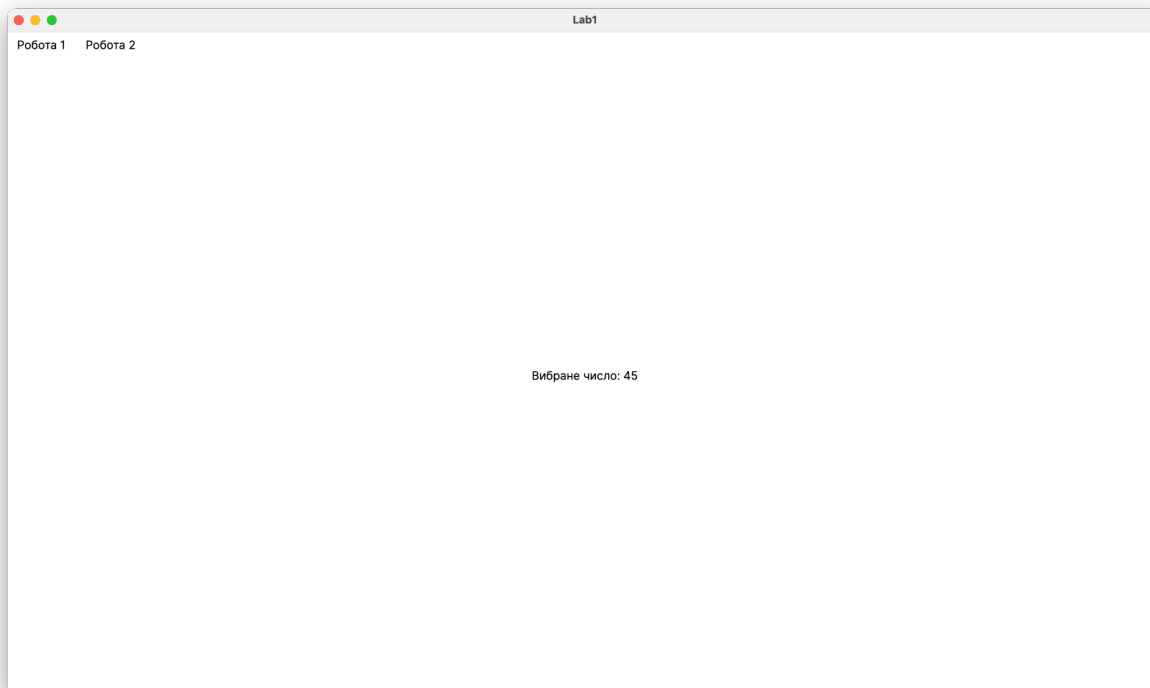
```

```
var dialog2 = new Dialog2();  
  
dialog2.Show();  
  
Close();  
  
}  
}
```

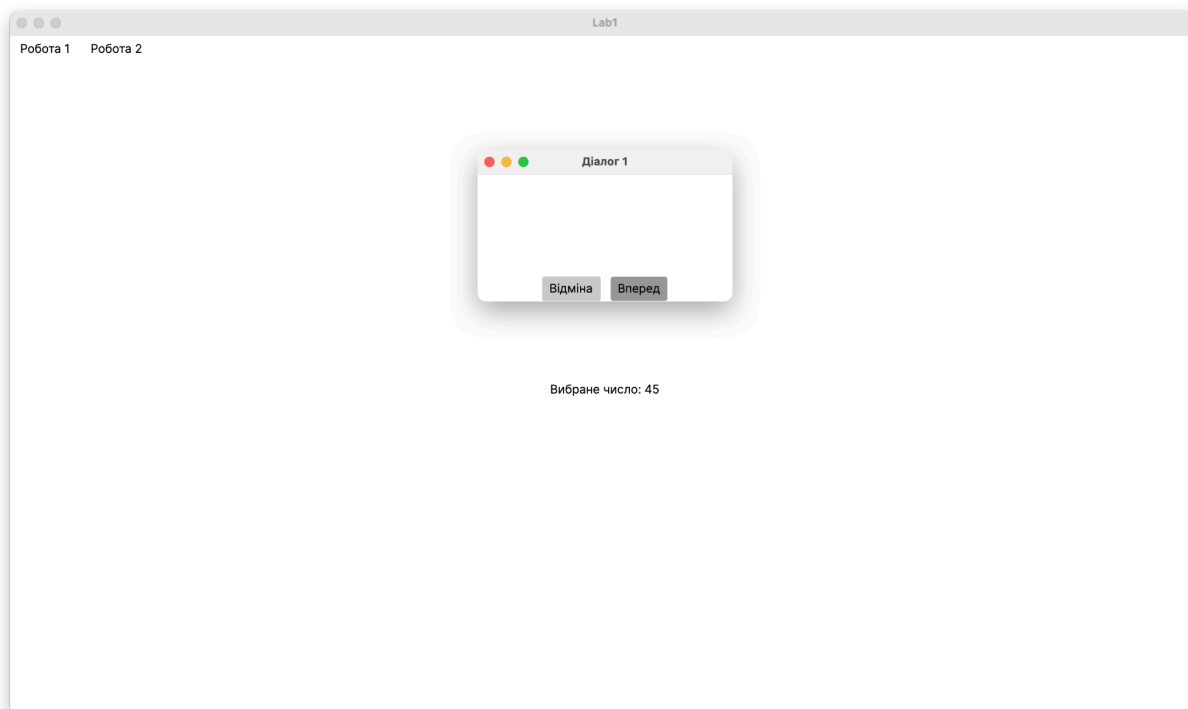
Тестування програми:

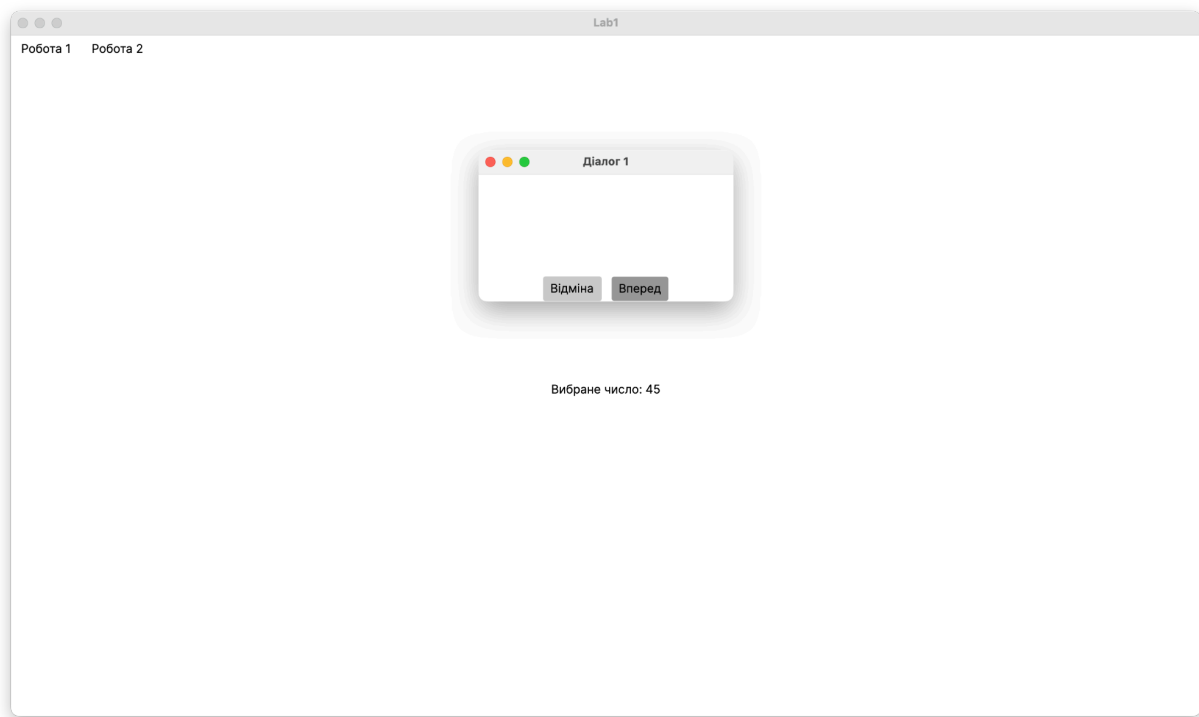
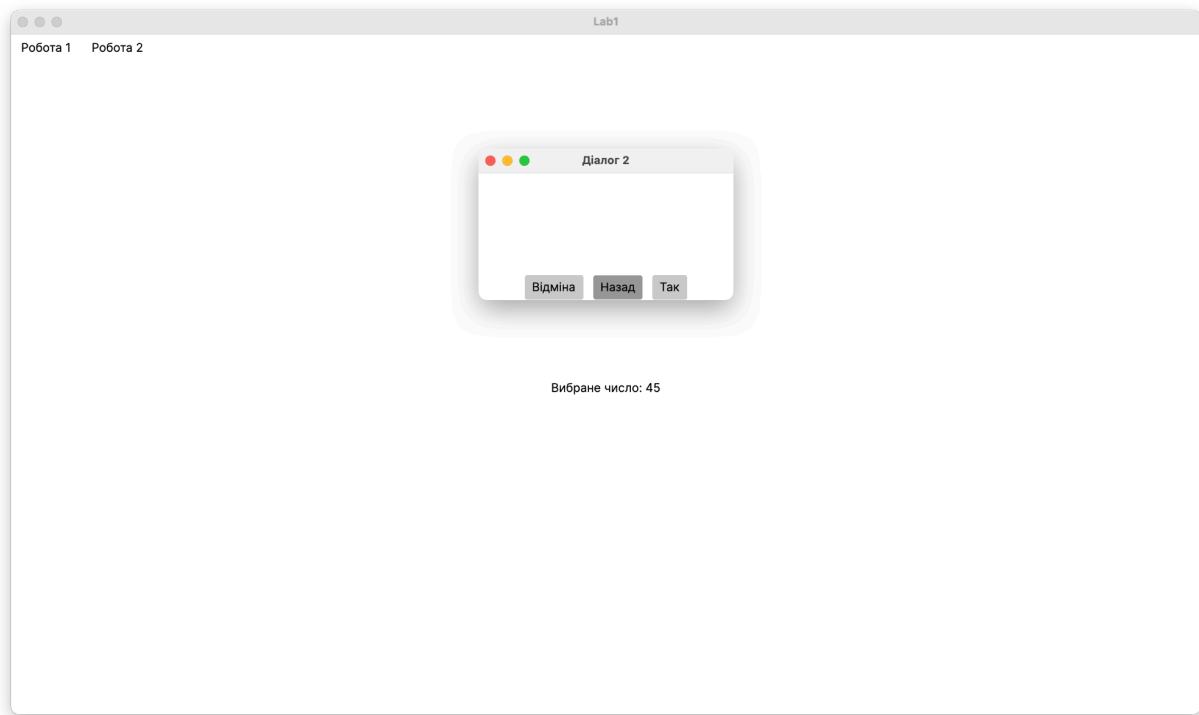
Робота 1:

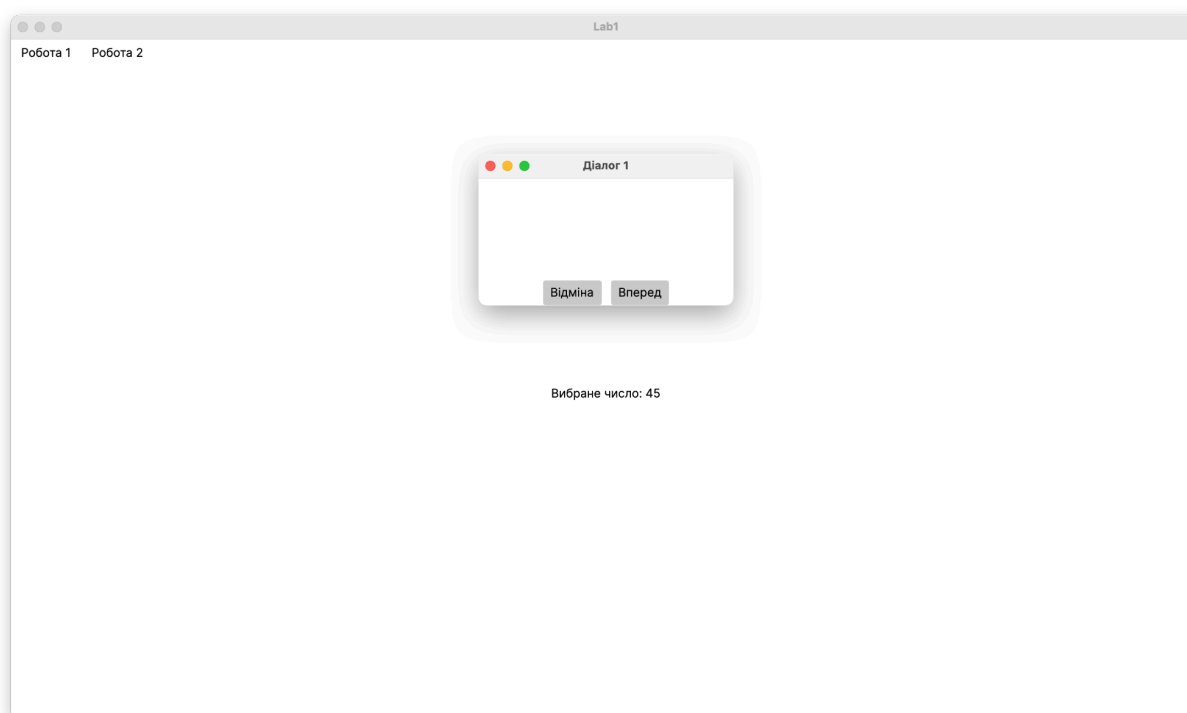
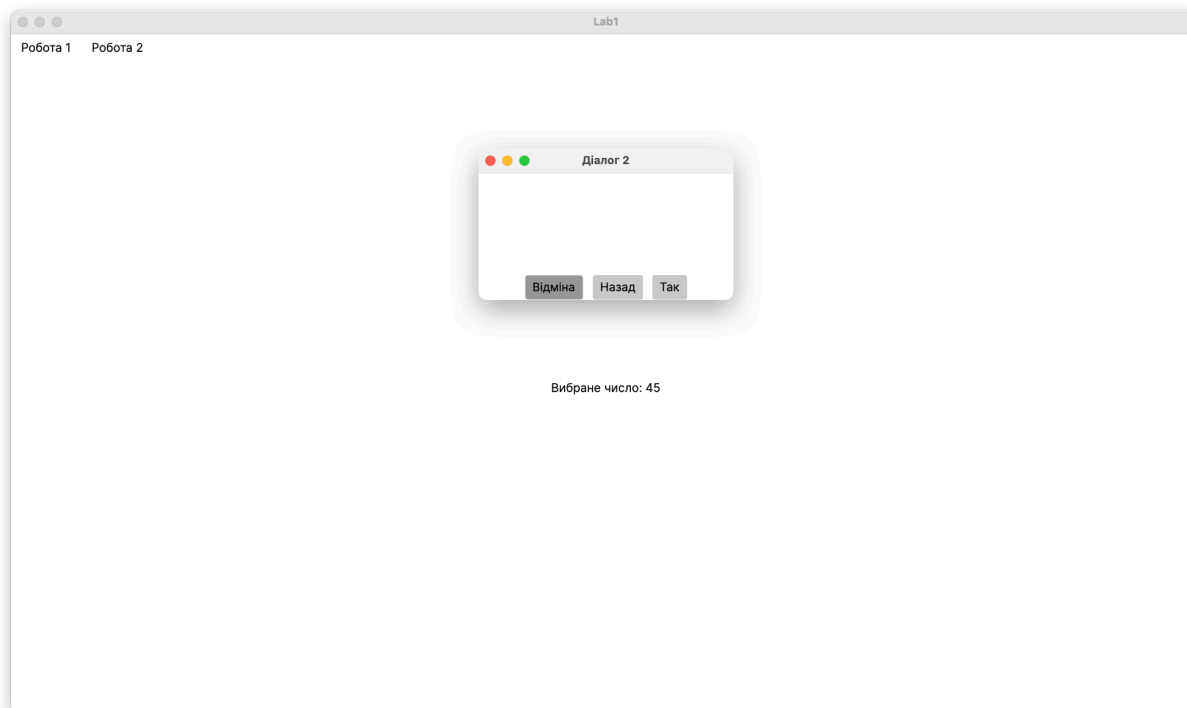




Робота 2:







Висновок:

Засвоїв теоретичний матеріал та набув практичних навичок роботи з віконними застосунками. Навчився розбивати застосунок на незалежні модулі, що є дуже зручним, гнучким і надійним методом будування застосунку в Об'єктно-орієнтованому програмуванні.