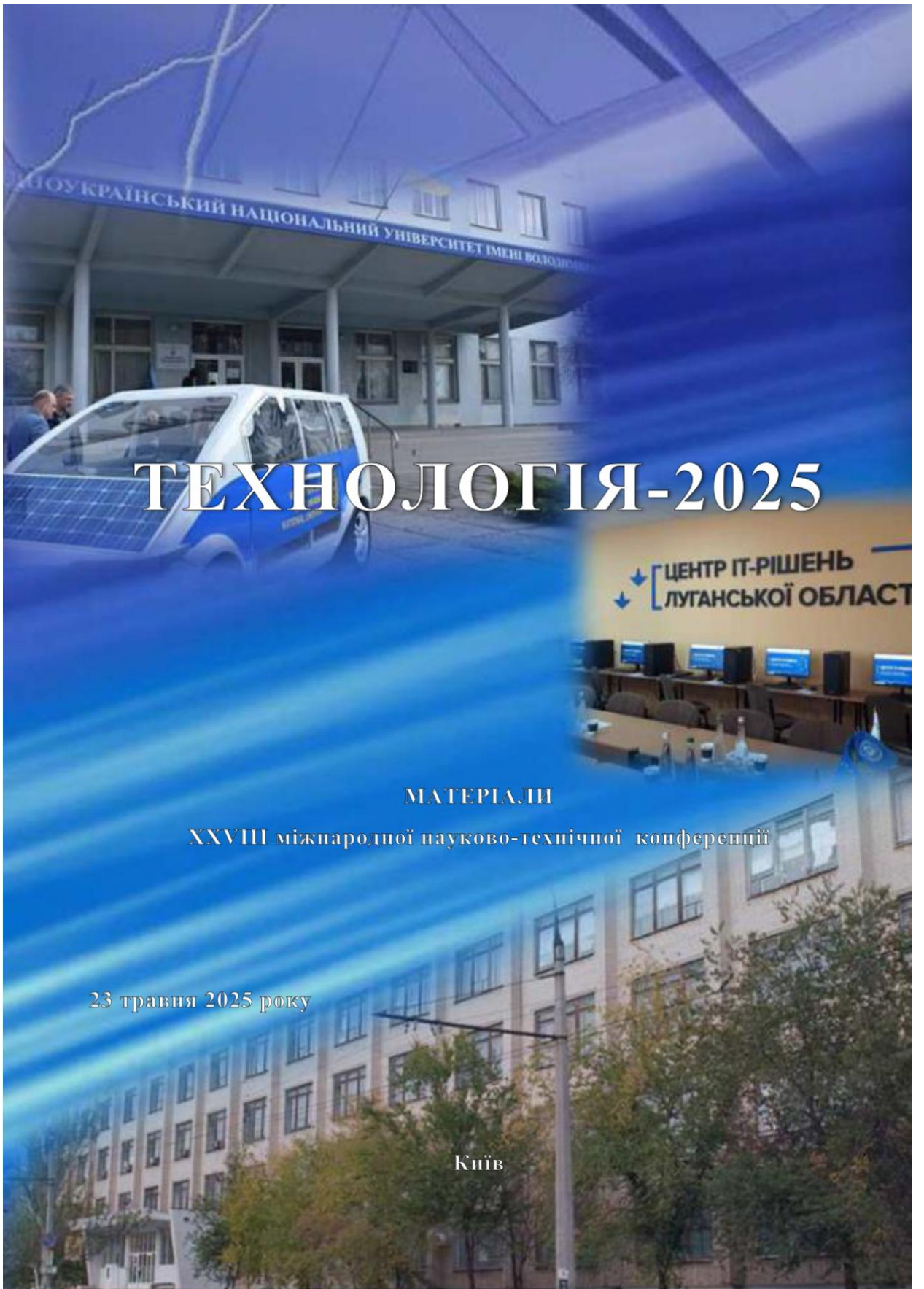


ДОДАТОК А  
Апробація результатів роботи



Рисунок А.1 – Сертифікат за представлення доповіді з теми «Програмний засіб для автоматизації визначення параметрів технологічних процесів у рамках підготовки здобувачів технічної освіти» на міжнародній науковій конференції «Технологія-2025»



# ТЕХНОЛОГІЯ-2025

МАТЕРІАЛИ

XXVIII міжнародної науково-технічної конференції

23 травня 2025 року

Київ

---

**ПРОГРАМНИЙ ЗАСІБ ДЛЯ АВТОМАТИЗАЦІЇ ВИЗНАЧЕННЯ ПАРАМЕТРІВ  
ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ У РАМКАХ ПІДГОТОВКИ ЗДОБУВАЧІВ  
ТЕХНІЧНОЇ ОСВІТИ**

Кислий Б. І., Жарікова І. В., к.т.н., доцент

*Харківський національний університет радіоелектроніки*

На сьогодні велику актуальність мають завдання автоматизації різних етапів моделювання, планування, технічної підготовки та виконання технологічних процесів (ТП) у багатьох галузях промисловості, зокрема й у сфері приладобудування [1-3].

На основі проведеного аналізу предметної області предметом розробки обрано програмний засіб (ПЗ), який реалізує обчислювальні алгоритми для проведення лабораторних занять здобувачів технічних закладів освіти з дисциплін, пов'язаних з дослідженням параметрів автоматизованих ТП.

Сучасна освітня практика у сфері автоматизації ТП та програмування передбачає активне використання інженерних розрахунків під час виконання лабораторних і практичних робіт. Однак, у більшості випадків, ці розрахунки виконуються вручну або з використанням універсальних калькуляторів, що не враховують специфіку методичних вказівок і не дозволяють легко адаптуватися до змін навчального матеріалу.

У той же час, студенти протягом навчання набувають навичок програмування, що відкриває можливість створення власних програмних засобів для спрощення розрахункової частини робіт. Тож набуває особливої актуальності розробка спеціалізованого ПЗ, адаптованого до конкретної спеціальності. Таке ПЗ дасть змогу студентам вводити необхідні параметри, отримувати результати в зручній для аналізу даних формі, а також легко редагувати або розширювати програму відповідно до оновлень методичних вказівок з технічних дисциплін. Відкритий похідний код, розміщений у загальнодоступному репозиторії, забезпечить гнучкість і можливість колективної розробки.

У процесі аналізу існуючих рішень для визначення параметрів автоматизованих ТП приладобудівної галузі було розглянуто три основні підходи: використання калькуляторів, власних програмних засобів і письмових розрахунків. Кожен із них має свої переваги та недоліки, що безпосередньо впливають на зручність, точність та ефективність виконання обчислень у навчальному процесі.

Калькулятори залишаються одним із найбільш популярних і зручних інструментів завдяки своїй стабільності та відносній гнучкості у використанні. Вони дозволяють швидко вводити числові значення й отримувати результат, не потребуючи складних налаштувань або додаткового програмування. Однак за потреби повторення певних розрахунків виникає необхідність кожного разу самостійно переписувати формули вручну, що не лише знижує ефективність, а й підвищує ризик помилок.

Студенти також можуть розробляти власні програмні проекти, які можуть забезпечити максимальну гнучкість і адаптованість до конкретних потреб. Вони дозволяють зберігати структуру формул, автоматизувати процес введення змінних і отримання результатів, а також уникати зайвого переписування одних і тих самих виразів. Такий підхід значно заощадить час і знизить імовірність помилок під час повторних обчислень. Проте основним недоліком є те, що такі інструменти потребують створення студентом, що вимагає певних навичок програмування та витрат часу. Окрім того, на етапі формування функціоналу ПЗ існує ризик допущення додаткових помилок, які залежать від уважності та компетентності розробника.

Письмові розрахунки є найбільш трудомістким процесом, а також суттєвим недоліком є висока ймовірність помилок через людський фактор, особливо в умовах обмеженого часу та/або втоми.

## Секція 2: Технічні науки

Загалом, аналіз показує, що найперспективнішим напрямом є створення власного програмного забезпечення, яке дозволяє об'єднати зручність автоматизації з гнучкістю налаштування під специфічні навчальні потреби. Попри те, що його впровадження потребує початкових зусиль, такі рішення у довгостроковій перспективі є найбільш ефективними та корисними для студентів.

Майбутній інтерфейс розроблюваного ПЗ наведено на рис. 1.

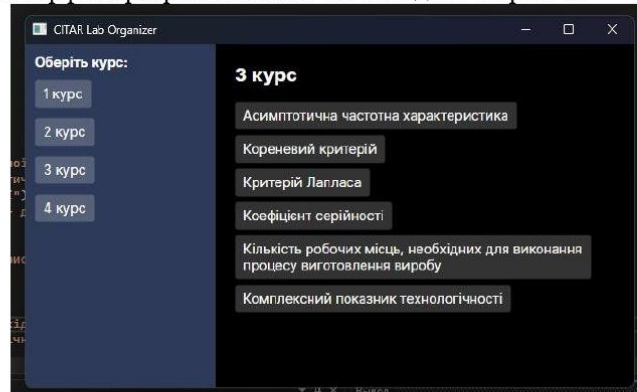


Рис. 1. Інтерфейс розробленого програмного забезпечення

У ПЗ буде передбачено виконання таких функцій:

- розрахунок параметрів автоматизованих ТП згідно зі специфікою методичних вказівок до лабораторного практикуму здобувачів освіти;
- автоматичне створення графіків і діаграм, необхідних для виконання завдань лабораторних робіт;
- покроковий перегляд розрахунків параметрів автоматизованих ТП.

Відкритий характер розробки, передбачений проектом, дозволяє у майбутньому доповнювати функціонал, адаптувати програму до інших дисциплін або специфіки технічного обладнання, що використовується у закладі освіти. Це робить програмний засіб потенційно корисним не лише в межах окремого навчального курсу, а й як частину ширшої цифрової освітньої екосистеми.

Таким чином, створення ПЗ для визначення параметрів автоматизованих ТП не лише відповідає сучасним вимогам до навчального процесу, а й сприяє підвищенню якості інженерної підготовки студентів.

## Література

1. Igor Nevliudov, & et al. (2021). Automation of Mathematical Modeling of Physical and Technological Processes in the Electronic Devices Manufacture. Proceedings of the XII International Scientific Conference «Functional Basis of Nanoelectronics» – Odessa, September 20-24, 74-77.

2. Ботман І.В. Автоматизація процесу реконструювання параметрів технологічного процесу виготовлення гнучких структур // Всеукраїнська науково-практична Інтернет-конференція Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку. – Черкаси, 12-18 березня 2018 року. Черкаси. С. 75-77.

3. Igor Nevliudov, Iryna Botsman, Olena Chala, Kirill Khrustalev. Automated System Development for the Printed Circuit Boards Optical Inspection Using Machine Learning Methods // Proceedings of the 10-th International Scientific and Technical Conference «INFORMATION SYSTEMS AND TECHNOLOGIES (IST-2021)». – Odessa, September 13-19, 2021. – PP. 234-238.

## ДОДАТОК Б

## Програмний код «MainWindow» для ПЗ

```

<Window xmlns="https://github.com/avaloniaui"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:vm="using:CITAR_Lab_Organizer_Beta.ViewModels"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
        mc:Ignorable="d" d:DesignWidth="640" d:DesignHeight="480"
        MinWidth="640" MinHeight="480" MaxWidth="640"
MaxHeight="480"
        x:Class="CITAR_Lab_Organizer_Beta.Views.MainWindow"
        x:DataType="vm:MainWindowViewModel"
        Title="CITAR Lab Organizer"
        Icon="/Assets/logo-final.ico">

    <Design.DataContext>
        <vm:MainWindowViewModel/>
    </Design.DataContext>

    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="200"/>
            <ColumnDefinition Width="*/>
        </Grid.ColumnDefinitions>

        <Border Grid.Column="0" Background="#2E3A59" Padding="10">
            <StackPanel Spacing="10">
                <Image Source="/Assets/test-2.png"></Image>
                <TextBlock Text="Оберіть курс:" FontWeight="Bold"
Foreground="White"/>
                <Button Content="1 курс" Command="{Binding
SelectCourseCommand}" CommandParameter="1"/>

```

```

        <Button Content="2 кypc" Command="{Binding
SelectCourseCommand}" CommandParameter="2"/>
        <Button Content="3 кypc" Command="{Binding
SelectCourseCommand}" CommandParameter="3"/>
        <Button Content="4 кypc" Command="{Binding
SelectCourseCommand}" CommandParameter="4"/>
    </StackPanel>
</Border>

    <ContentControl Grid.Column="1" Content="{Binding
SelectedCourseContent}" />
</Grid>
</Window>

```

```

using Avalonia.Controls;

namespace CITAR_Lab_Organizer_Beta.Views
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }
    }
}

```

```

using Avalonia.Controls;

namespace CITAR_Lab_Organizer_Beta.Views
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }
    }
}

```

```

using ReactiveUI;
using System;
using System.Collections.ObjectModel;
using System.Reactive;
using Avalonia.Controls;
using CITAR_Lab_Organizer_Beta.Views;

namespace CITAR_Lab_Organizer_Beta.ViewModels
{
    public class MainWindowViewModel : ViewModelBase
    {
        private object? _selectedCourseContent;
        public object? SelectedCourseContent
        {
            get => _selectedCourseContent;
            set => this.RaiseAndSetIfChanged(ref
_selectedCourseContent, value);
        }

        public ReactiveCommand<string, Unit> SelectCourseCommand {
get; }

        public MainWindowViewModel()
        {
            SelectCourseCommand =
ReactiveCommand.Create<string>(OnCourseSelected);
        }

        private void OnCourseSelected(string course)
        {
            SelectedCourseContent = new CourseContentView {
DataContext = new CourseContentViewModel(course) };
        }
    }
}

```

## ДОДАТОК В

## Програмний код «CourseContent» для ПЗ

```

<!-- xmlns:views="clr-namespace:CITAR_Lab_Organizer_Beta.Views"-->
<UserControl xmlns="https://github.com/avaloniaui"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
    xmlns:vm="using:CITAR_Lab_Organizer_Beta.ViewModels"
    mc:Ignorable="d"

x:Class="CITAR_Lab_Organizer_Beta.Views.CourseContentView"
    x:DataType="vm:CourseContentViewModel">

    <StackPanel Margin="20" Spacing="10">
        <!--<views:CourseContentView />-->
        <TextBlock Text="{Binding CourseName}" FontSize="20"
FontWeight="Bold"/>
        <ItemsControl ItemsSource="{Binding Buttons}">
            <ItemsControl.ItemTemplate>
                <DataTemplate>
                    <Button Content="{Binding}"
Margin="0,5,0,0"
                                Click="Button_Click"/>
                </DataTemplate>
            </ItemsControl.ItemTemplate>
        </ItemsControl>
    </StackPanel>
</UserControl>

using ReactiveUI;
using System.Collections.ObjectModel;

namespace CITAR_Lab_Organizer_Beta.ViewModels
{
    public class CourseContentViewModel : ViewModelBase
    {
        public string CourseName { get; }
        public ObservableCollection<string> Buttons { get; set; } =
new ObservableCollection<string>();

        public CourseContentViewModel(string course)
        {
            CourseName = $"{course} курс";

            switch (course)
            {

```



```

using System.Linq;
using System;
using System.Text;
using Tmds.DBus.Protocol;
using Avalonia.Input;
using Avalonia.Controls.Primitives;
using Avalonia.Layout;
using Avalonia.Platform;

namespace CITAR_Lab_Organizer_Beta.Views
{
    public partial class CourseContentView : UserControl
    {
        public CourseContentView()
        {
            InitializeComponent();
        }
        void ShowGraph(List<(double t, double R)> values)
        {
            var series = new LineSeries<double>
            {
                Values = values.Select(v => v.R).ToArray(),
                Name = "R(t)"
            };

            var chart = new CartesianChart
            {
                Series = new ISeries[] { series },
                XAxes = new[] { new Axis { Labels = values.Select(v =>
v.t.ToString("0.0")).ToArray(), Name = "t" } },
                YAxes = new[] { new Axis { Name = "R(t)" } },
                Width = 600,
                Height = 300,
                Margin = new Thickness(20)
            };

            var window = new Window
            {
                Title = "Γραφικ R(t)",
                Width = 640,
                Height = 400,
                Content = chart,
                WindowStartupLocation =
WindowStartupLocation.CenterOwner
            };

            window.Show();
        }
        void ShowBarChart(string title, List<(string label, double
value)> values)
        {

```

```

var series = new ColumnSeries<double>
{
    Values = values.Select(v => v.value).ToArray(),
    Name = "W"
};

var chart = new CartesianChart
{
    Series = new ISeries[] { series },
    XAxes = new[] { new Axis { Labels = values.Select(v =>
v.label).ToArray(), Name = "Показники" } },
    YAxes = new[] { new Axis { Name = "Значення W_i" } },
    Width = 600,
    Height = 300,
    Margin = new Thickness(20)
};

var window = new Window
{
    Title = title,
    Width = 640,
    Height = 400,
    Content = chart,
    WindowStartupLocation =
WindowStartupLocation.CenterOwner
};

window.Show();
}

private void CopyButton_OnClick(object? sender)
{
    var clipboard = TopLevel.GetTopLevel(this)?.Clipboard;
    var dataObject = new DataObject();
    dataObject.Set(DataFormats.Text, sender);
    clipboard.SetDataObjectAsync(dataObject);
}

private void ShowFinalInfoMessage(string title)
{
    var children = new List<Control>
    {
        new TextBlock
        {
            Text = "Відповідь:\n" + title,
            Margin = new Thickness(10),
            FontWeight = Avalonia.Media.FontWeight.Bold,
            HorizontalAlignment =
Avalonia.Layout.HorizontalAlignment.Center
        }
    };
};

```

```

var stackPanel = new StackPanel
{
    Margin = new Thickness(10),
    Orientation = Orientation.Vertical,
    HorizontalAlignment =
Avalonia.Layout.HorizontalAlignment.Center
};

foreach (var child in children)
{
    stackPanel.Children.Add(child);
}

// Обгортаємо StackPanel у ScrollViewer
var scrollView = new ScrollViewer
{
    Content = stackPanel,
    VerticalScrollBarVisibility =
ScrollBarVisibility.Auto,
    HorizontalScrollBarVisibility =
ScrollBarVisibility.Disabled
};
var uri = new
Uri("avares://CITAR_Lab_Organizer_Beta/Assets/logo-final.ico");
var icon = new WindowIcon(AssetLoader.Open(uri));
var dialog = new Window
{
    Title = "Інформація",
    Icon = icon,
    Width = 640,
    Height = 360,
    MinWidth = 640,
    MinHeight = 360,
    MaxWidth = 640,
    MaxHeight = 360,
    WindowStartupLocation =
WindowStartupLocation.CenterOwner,
    Content = scrollView
};
dialog.Show();
CopyButton_OnClick("Відповідь:\n" + title);
}
void Main_HollowCylinder(double m, double h, double D, double
d)
{
    double p = ((4 * m) / ((Math.PI * h) * (Math.Pow(D, 2) -
(Math.Pow(d, 2))));
    string final = ("p = " + p + ".");
    ShowFinalInfoMessage(final);
}
void Main_SolidCylinder(double m, double h, double d)

```

```

{
    double p = ((4 * m) / ((Math.PI * h) * (Math.Pow(d, 2))));
    string final = ("p = " + p + ".");
    ShowFinalInfoMessage(final);
}
void Main_PlateDensity(double m, double a, double b, double c)
{
    double p = (m / (a * b * c));
    string final = ("p = " + p + ".");
    ShowFinalInfoMessage(final);
}
void Main_BallDensity(double m, double D)
{
    double p = ((6 * m) / (Math.PI * (Math.Pow(D, 3))));
    string final = ("p = " + p + ".");
    ShowFinalInfoMessage(final);
}
void Main_MaximumGeneralizedUtilityModel(double k_plus_1,
double k_minus_1, double k_plus_2, double k_minus_2, double k_plus_3,
double k_minus_3, double k_1, double k_2, double k_3, double
k_plus_minus_1, double k_plus_minus_2, double k_plus_minus_3, double
k_11, double k_21, double k_31, double k_12, double k_22, double k_32,
double k_13, double k_23, double k_33, double k_14, double k_24,
double k_34, double k_15, double k_25, double k_35, double k_16,
double k_26, double k_36, double k_17, double k_27, double k_37,
double k_18, double k_28, double k_38, double k_19, double k_29,
double k_39, double k_110, double k_210, double k_310)
{
    double p_11_x, p_21_x, p_31_x, p_12_x, p_22_x, p_32_x,
p_13_x, p_23_x,
p_33_x, p_14_x, p_24_x, p_34_x, p_15_x, p_25_x, p_35_x, p_16_x,
p_26_x, p_36_x,
p_17_x, p_27_x, p_37_x, p_18_x, p_28_x, p_38_x, p_19_x, p_29_x,
p_39_x, p_110_x,
p_210_x, p_310_x;

    p_11_x = ((k_11 - k_minus_1) / (k_plus_1 - k_minus_1));
    p_21_x = ((k_21 - k_minus_2) / (k_plus_2 - k_minus_2));
    p_31_x = ((k_31 - k_minus_3) / (k_plus_3 - k_minus_3));
    p_12_x = ((k_12 - k_minus_1) / (k_plus_1 - k_minus_1));
    p_22_x = ((k_22 - k_minus_2) / (k_plus_2 - k_minus_2));
    p_32_x = ((k_32 - k_minus_3) / (k_plus_3 - k_minus_3));
    p_13_x = ((k_13 - k_minus_1) / (k_plus_1 - k_minus_1));
    p_23_x = ((k_23 - k_minus_2) / (k_plus_2 - k_minus_2));
    p_33_x = ((k_33 - k_minus_3) / (k_plus_3 - k_minus_3));
    p_14_x = ((k_14 - k_minus_1) / (k_plus_1 - k_minus_1));
    p_24_x = ((k_24 - k_minus_2) / (k_plus_2 - k_minus_2));
    p_34_x = ((k_34 - k_minus_3) / (k_plus_3 - k_minus_3));
    p_15_x = ((k_15 - k_minus_1) / (k_plus_1 - k_minus_1));
    p_25_x = ((k_25 - k_minus_2) / (k_plus_2 - k_minus_2));
    p_35_x = ((k_35 - k_minus_3) / (k_plus_3 - k_minus_3));

```

```

p_16_x = ((k_16 - k_minus_1) / (k_plus_1 - k_minus_1));
p_26_x = ((k_26 - k_minus_2) / (k_plus_2 - k_minus_2));
p_36_x = ((k_36 - k_minus_3) / (k_plus_3 - k_minus_3));
p_17_x = ((k_17 - k_minus_1) / (k_plus_1 - k_minus_1));
p_27_x = ((k_27 - k_minus_2) / (k_plus_2 - k_minus_2));
p_37_x = ((k_37 - k_minus_3) / (k_plus_3 - k_minus_3));
p_18_x = ((k_18 - k_minus_1) / (k_plus_1 - k_minus_1));
p_28_x = ((k_28 - k_minus_2) / (k_plus_2 - k_minus_2));
p_38_x = ((k_38 - k_minus_3) / (k_plus_3 - k_minus_3));
p_19_x = ((k_19 - k_minus_1) / (k_plus_1 - k_minus_1));
p_29_x = ((k_29 - k_minus_2) / (k_plus_2 - k_minus_2));
p_39_x = ((k_39 - k_minus_3) / (k_plus_3 - k_minus_3));
p_110_x = ((k_110 - k_minus_1) / (k_plus_1 - k_minus_1));
p_210_x = ((k_210 - k_minus_2) / (k_plus_2 - k_minus_2));
p_310_x = ((k_310 - k_minus_3) / (k_plus_3 - k_minus_3));
double a_11, a_12, a_13, a_21, a_22, a_23, a_31, a_32,
a_33, a_41, a_42, a_43;
a_11 = a_12 = a_13 = (1 / 3);
a_21 = 0.3;
a_22 = 0.5;
a_23 = a_32 = a_33 = a_41 = 0.2;
a_31 = 0.6;
a_42 = 0.1;
a_43 = 0.7;
double px_1_1, px_2_1, px_3_1, px_4_1;
double px_1_2, px_2_2, px_3_2, px_4_2;
double px_1_3, px_2_3, px_3_3, px_4_3;
double px_1_4, px_2_4, px_3_4, px_4_4;
double px_1_5, px_2_5, px_3_5, px_4_5;
double px_1_6, px_2_6, px_3_6, px_4_6;
double px_1_7, px_2_7, px_3_7, px_4_7;
double px_1_8, px_2_8, px_3_8, px_4_8;
double px_1_9, px_2_9, px_3_9, px_4_9;
double px_1_10, px_2_10, px_3_10, px_4_10;
px_1_1 = (p_11_x / 3) + (p_21_x / 3) + (p_31_x / 3);
px_2_1 = (p_11_x * a_21) + (p_21_x * a_22) + (p_31_x *
a_23);
px_3_1 = (p_11_x * a_31) + (p_21_x * a_32) + (p_31_x *
a_33);
px_4_1 = (p_11_x * a_41) + (p_21_x * a_42) + (p_31_x *
a_43);
px_1_2 = (p_12_x / 3) + (p_22_x / 3) + (p_32_x / 3);
px_2_2 = (p_12_x * a_21) + (p_22_x * a_22) + (p_32_x *
a_23);
px_3_2 = (p_12_x * a_31) + (p_22_x * a_32) + (p_32_x *
a_33);
px_4_2 = (p_12_x * a_41) + (p_22_x * a_42) + (p_32_x *
a_43);
px_1_3 = (p_13_x / 3) + (p_23_x / 3) + (p_33_x / 3);
px_2_3 = (p_13_x * a_21) + (p_23_x * a_22) + (p_33_x *
a_23);

```

```

px_3_3 = (p_13_x * a_31) + (p_23_x * a_32) + (p_33_x *
a_33);
px_4_3 = (p_13_x * a_41) + (p_23_x * a_42) + (p_33_x *
a_43);
px_1_4 = (p_14_x / 3) + (p_24_x / 3) + (p_34_x / 3);
px_2_4 = (p_14_x * a_21) + (p_24_x * a_22) + (p_34_x *
a_23);
px_3_4 = (p_14_x * a_31) + (p_24_x * a_32) + (p_34_x *
a_33);
px_4_4 = (p_14_x * a_41) + (p_24_x * a_42) + (p_34_x *
a_43);
px_1_5 = (p_15_x / 3) + (p_25_x / 3) + (p_35_x / 3);
px_2_5 = (p_15_x * a_21) + (p_25_x * a_22) + (p_35_x *
a_23);
px_3_5 = (p_15_x * a_31) + (p_25_x * a_32) + (p_35_x *
a_33);
px_4_5 = (p_15_x * a_41) + (p_25_x * a_42) + (p_35_x *
a_43);
px_1_6 = (p_16_x / 3) + (p_26_x / 3) + (p_36_x / 3);
px_2_6 = (p_16_x * a_21) + (p_26_x * a_22) + (p_36_x *
a_23);
px_3_6 = (p_16_x * a_31) + (p_26_x * a_32) + (p_36_x *
a_33);
px_4_6 = (p_16_x * a_41) + (p_26_x * a_42) + (p_36_x *
a_43);
px_1_7 = (p_17_x / 3) + (p_27_x / 3) + (p_37_x / 3);
px_2_7 = (p_17_x * a_21) + (p_27_x * a_22) + (p_37_x *
a_23);
px_3_7 = (p_17_x * a_31) + (p_27_x * a_32) + (p_37_x *
a_33);
px_4_7 = (p_17_x * a_41) + (p_27_x * a_42) + (p_37_x *
a_43);
px_1_8 = (p_18_x / 3) + (p_28_x / 3) + (p_38_x / 3);
px_2_8 = (p_18_x * a_21) + (p_28_x * a_22) + (p_38_x *
a_23);
px_3_8 = (p_18_x * a_31) + (p_28_x * a_32) + (p_38_x *
a_33);
px_4_8 = (p_18_x * a_41) + (p_28_x * a_42) + (p_38_x *
a_43);
px_1_9 = (p_19_x / 3) + (p_29_x / 3) + (p_39_x / 3);
px_2_9 = (p_19_x * a_21) + (p_29_x * a_22) + (p_39_x *
a_23);
px_3_9 = (p_19_x * a_31) + (p_29_x * a_32) + (p_39_x *
a_33);
px_4_9 = (p_19_x * a_41) + (p_29_x * a_42) + (p_39_x *
a_43);
px_1_10 = (p_110_x / 3) + (p_210_x / 3) + (p_310_x / 3);
px_2_10 = (p_110_x * a_21) + (p_210_x * a_22) + (p_310_x *
a_23);
px_3_10 = (p_110_x * a_31) + (p_210_x * a_32) + (p_310_x *
a_33);

```

```

px_4_10 = (p_110_x * a_41) + (p_210_x * a_42) + (p_310_x *
a_43);
    string final = "Функція максимальної узагальненої
корисності\npx_1_1 = " + px_1_1 + "\npx_2_1 = " + px_2_1 + "\npx_3_1 =
" + px_3_1 + "\npx_4_1 = " + px_4_1 + "\npx_1_2 = " + px_1_2 +
"\npx_2_2 = " + px_2_2 + "\npx_3_2 = " + px_3_2 + "\npx_4_2 = " +
px_4_2 + "\npx_1_3 = " + px_1_3 + "\npx_2_3 = " + px_2_3 + "\npx_3_3 =
" + px_3_3 + "\npx_4_3 = " + px_4_3 + "\npx_1_4 = " + px_1_4 +
"\npx_2_4 = " + px_2_4 + "\npx_3_4 = " + px_3_4 + "\npx_4_4 = " +
px_4_4 + "\npx_1_5 = " + px_1_5 + "\npx_2_5 = " + px_2_5 + "\npx_3_5 =
" + px_3_5 + "\npx_4_5 = " + px_4_5 + "\npx_1_6 = " + px_1_6 +
"\npx_2_6 = " + px_2_6 + "\npx_3_6 = " + px_3_6 + "\npx_4_6 = " +
px_4_6 + "\npx_1_7 = " + px_1_7 + "\npx_2_7 = " + px_2_7 + "\npx_3_7 =
" + px_3_7 + "\npx_4_7 = " + px_4_7 + "\npx_1_8 = " + px_1_8 +
"\npx_2_8 = " + px_2_8 + "\npx_3_8 = " + px_3_8 + "\npx_4_* = " +
px_4_8 + "\npx_1_9 = " + px_1_9 + "\npx_2_9 = " + px_2_9 + "\npx_3_9 =
" + px_3_9 + "\npx_4_9 = " + px_4_9 + "\npx_1_10 = " + px_1_10 +
"\npx_2_10 = " + px_2_10 + "\npx_3_10 = " + px_3_10 + "\npx_4_10 = " +
px_4_10;
        ShowFinalInfoMessage(final);
    }
    void Main_CriterionOfMaximumMathematicalExpectation(double
M_x1, double M_x2, double M_x3, double M_x4)
    {
        double E_x1 = Math.Max(M_x1, M_x2);
        double E_x2 = Math.Max(M_x3, M_x4);
        double E_x = Math.Max(E_x1, E_x2);
        string final = ("E_x = " + E_x + ".");
        ShowFinalInfoMessage(final);
    }
    void Main_CriterionOfMinimumDispersion(double D_x1, double
D_x2, double D_x3, double D_x4)
    {
        double E_x1 = Math.Min(D_x1, D_x2);
        double E_x2 = Math.Min(D_x3, D_x4);
        double E_x = Math.Min(E_x1, E_x2);
        string final = ("E_x = " + E_x + ".");
        ShowFinalInfoMessage(final);
    }
    void Main_ExpectationDispersion(double K, double M_x1, double
M_x2, double M_x3, double M_x4, double D_x1, double D_x2, double D_x3,
double D_x4)
    {
        double ED_1, ED_2, ED_3, ED_4;
        ED_1 = (M_x1 - (K * D_x1));
        ED_2 = (M_x2 - (K * D_x2));
        ED_3 = (M_x3 - (K * D_x3));
        ED_4 = (M_x4 - (K * D_x4));
        double ED_x1 = Math.Max(ED_1, ED_2);
        double ED_x2 = Math.Max(ED_3, ED_4);
        double ED = Math.Max(ED_x1, ED_x2);

```

```

        string final = ("ED_1 = " + ED_1 + ".\nED_2 = " + ED_2 +
".\nED_3 = " + ED_3 + ".\nED_4 = " + ED_4 + ".\nED = " + ED + ".");
        ShowFinalInfoMessage(final);
    }
    void Main_AsymptoticFrequencyResponse(double k, double T1,
double T2, double T3, double T4, double s, double v)
    {
        // double k,double T1,double T2,double T3,double T4,double
s,double v
        double W_s = ((k * ((T1 * s) + 1) * (((T2 * Math.Pow(s,
2)) + (T2 * s) + 1))) / (Math.Pow(s, v) * ((T3 * s) + 1) * (((T4 *
Math.Pow(s, 2)) + (T4 * s) + 1))));
        string final = ("W_s = " + W_s + ".");
        ShowFinalInfoMessage(final);
    }
    void Main_RootAndLaplasCriterion(double a, double b, double c,
double s)
    {
        double s1, s2;
        double D = (Math.Pow(b, 2) - (4 * a * c));
        double minusB = (b * (-1));
        double sqrtD = Math.Sqrt(D);
        double doubleA = (2 * a);
        s1 = ((minusB - sqrtD) / doubleA);
        s2 = ((minusB + sqrtD) / doubleA);
        double d = 0;
        double determinant = ((a * b) - (c * d));
        string final = ("Маємо рівняння вигляду:\n" + a + "s^2 + "
+ b + "s + " + c + " = 0.\nРезультат:\ns1 = " + s1 + ";\ns2 = " + s2 +
".\nТепер знайдемо значення детермінанта для критерія Лапласа: " +
determinant + ".");
        ShowFinalInfoMessage(final);
    }
    void Main_SerializationRate(double K_0, double n_p)
    {
        double K_C = K_0 / n_p;
        string final = ("K_C = " + K_C + ".");
        ShowFinalInfoMessage(final);
    }
    void Main_JobsNumber(double N, double K_0, double T_i, double
K, double F_D)
    {
        double n_p = ((N * K_0 * T_i) / (60 * K * F_D));
        string final = ("n_p = " + n_p + ".");
        ShowFinalInfoMessage(final);
    }
    void Main_ComplexManufacturabilityIndicator(double K_i, double
F_i)
    {
        double K = ((7 * K_i * F_i) / (7 * F_i));
        string final = ("K = " + K + ".");
    }

```

```

        ShowFinalInfoMessage(final);
    }
    void Main_NSAV1(double x1, double x2, double x3, double x4,
double x5, double x6, double x7, double x8, double x9, double x10,
double N_A, double T_A, double N_B, double T_B, double I, double
t_max)
    {
        double xx_1 = (x1 + x2 + x3);
        double xx_2 = (x4 + x5 + x6 + x7 + x8 + x9 + x10);
        double E_c_t_a = (xx_1 / I);
        double E_c_t_b = (xx_2 / I);
        double lambda_a = (N_A / T_A);
        double lambda_b = (N_B / T_B);
        double n01 = lambda_b / lambda_a;
        double n02 = ((n01 * E_c_t_a) - E_c_t_b);
        double N_0 = ((I * Math.Abs(n02) / (n01 - 1)));
        double C = (lambda_a / (((N_0) / I) - E_c_t_a));

        var values = new List<(double t, double R)>();
        for (double t = 0; t <= t_max; t += 1.0)
        {
            double R_t_T = Math.Exp(-1 * C * Math.Abs((N_0 / I) -
(E_c_t_a + E_c_t_b)) * t);
            values.Add((t, R_t_T));
        }

        // Графік
        ShowGraph(values);

        // Текстовий результат
        string final = $"E_c_t_a = {E_c_t_a}\nE_c_t_b =
{E_c_t_b}\nlambda_a = {lambda_a}\nlambda_b = {lambda_b}\nN_0 =
{N_0}\nC = {C}";
        ShowFinalInfoMessage(final);
    }

    void Main_NSAV2(
double W_1, double S_1, double V_1,
double W_2, double S_2, double V_2,
double W_3, double S_3, double V_3,
double W_4, double S_4, double V_4,
double W_5, double S_5, double V_5,
double W_6, double S_6, double V_6)
    {
        double N_1 = ((W_1 * S_1) / V_1);
        double N_2 = ((W_2 * S_2) / V_2);
        double N_3 = ((W_3 * S_3) / V_3);
        double N_4 = ((W_4 * S_4) / V_4);
        double N_5 = ((W_5 * S_5) / V_5);
        double N_6 = ((W_6 * S_6) / V_6);
    }

```

```

double r_2 = W_1 - S_1;
W_2 = W_1 - V_1;
N_2 = ((W_2 * S_2) / V_2);
double r_3 = r_2 - S_2;
W_3 = W_2 - V_2;
N_3 = ((W_3 * S_3) / V_3);
double r_4 = r_3 - S_3;
W_4 = W_3 - V_3;
N_4 = ((W_4 * S_4) / V_4);
double r_5 = r_4 - S_4;
W_5 = W_4 - V_4;
N_5 = ((W_5 * S_5) / V_5);
double r_6 = r_5 - S_5;
W_6 = W_5 - V_5;
N_6 = ((W_6 * S_6) / V_6);
double C = (W_6 / (W_6 + r_6 + 1));

// Показати графік
var barData = new List<(string label, double value)>
{
    ("W1", W_1),
    ("W2", W_2),
    ("W3", W_3),
    ("W4", W_4),
    ("W5", W_5),
    ("W6", W_6)
};

ShowBarChart("Залежність W1-W6", barData);

string final = $"C = {C}.";
ShowFinalInfoMessage(final);
}

void Main_NSAV3(double k, double N, double t1, double t2,
double t3, double t4, double i1, double i2, double i3, double i4,
double i)
{
    double nm1p1, nm2p2, nm3p3, nm4p4;
    nm1p1 = 1 / (N - i1 + 1);
    nm2p2 = 1 / (N - i2 + 1);
    nm3p3 = 1 / (N - i3 + 1);
    nm4p4 = 1 / (N - i4 + 1);
    if (i1 == N + 1) { nm1p1 = 0; }
    if (i2 == N + 1) { nm2p2 = 0; }
    if (i3 == N + 1) { nm3p3 = 0; }
    if (i4 == N + 1) { nm4p4 = 0; }
    double JM1, JM2, CD, lambdaI, t;
    JM1 = ((k - 1) * ((t1 + t2 + t3 + t4) / (nm1p1 + nm2p2 +
nm3p3 + nm4p4)));
}

```

```

        JM2 = (((N - i1 + 1) * t1) + ((N - i2 + 1) * t2) + ((N -
i3 + 1) * t3) + ((N - i4 + 1) * t4));
        CD = ((nm1p1 + nm2p2 + nm3p3 + nm4p4) / (t1 + t2 + t3 +
t4));
        lambdaI = (CD * (N - (i - 1)));
        t = (1 / (lambdaI));
        string final = ("JM1 = " + JM1 + "\nJM2 = " + JM2 + "\nCD
= " + CD + "\nlambdaI = " + lambdaI + "\nt = " + t + ".");
        ShowFinalInfoMessage(final);
    }
    void Main_NSAV4(
        double p1, double p2, double p3, double p4, double p5, double p6,
double p7, double p8, double p9, double p10,
        double p11, double p12, double p13, double p14, double p15, double
p16, double p17, double p18, double p19, double p20,
        double p21, double p22, double p23, double p24, double p25, double
p26, double p27, double p28, double p29, double p30,
        double y1, double y2, double y3, double y4, double y5, double y6,
double y7, double y8, double y9, double y10,
        double y11, double y12, double y13, double y14, double y15, double
y16, double y17, double y18, double y19, double y20,
        double y21, double y22, double y23, double y24, double y25, double
y26, double y27, double y28, double y29, double y30)
    {
        double R = 1 - (
            p1 * y1 + p2 * y2 + p3 * y3 + p4 * y4 + p5 * y5 + p6 *
y6 + p7 * y7 + p8 * y8 + p9 * y9 + p10 * y10 +
            p11 * y11 + p12 * y12 + p13 * y13 + p14 * y14 + p15 *
y15 + p16 * y16 + p17 * y17 + p18 * y18 + p19 * y19 + p20 * y20 +
            p21 * y21 + p22 * y22 + p23 * y23 + p24 * y24 + p25 *
y25 + p26 * y26 + p27 * y27 + p28 * y28 + p29 * y29 + p30 * y30
        );

        var pValues = new[]
        {
            p1, p2, p3, p4, p5, p6, p7, p8, p9, p10,
            p11, p12, p13, p14, p15, p16, p17, p18, p19, p20,
            p21, p22, p23, p24, p25, p26, p27, p28, p29, p30
        };

        var graphPoints = new List<(double x, double y)>();
        for (int i = 0; i < pValues.Length; i++)
        {
            graphPoints.Add((i + 1, pValues[i]));
        }

        ShowGraph(graphPoints);

        string final = $"R = {R}.";
        ShowFinalInfoMessage(final);
    }
}

```

```

static TextBox CreateInput(string placeholder) => new()
{
    Watermark = placeholder,
    Width = 250,
    Margin = new Thickness(10),
    HorizontalAlignment =
Avalonia.Layout.HorizontalAlignment.Center
};
private void ShowInputDialog(string title, string[]
placeholders, Action<TextBox[]> onCalculate)
{
    var inputs = placeholders.Select(ph => new TextBox
    {
        Watermark = ph,
        Width = 250,
        Margin = new Thickness(10),
        HorizontalAlignment =
Avalonia.Layout.HorizontalAlignment.Center
    }).ToArray();

    var calculateButton = new Button
    {
        Content = "Розрахувати!",
        Margin = new Thickness(10),
        HorizontalAlignment =
Avalonia.Layout.HorizontalAlignment.Center
    };

    var children = new List<Control>
    {
        new TextBlock
        {
            Text = title,
            Margin = new Thickness(10),
            FontWeight = Avalonia.Media.FontWeight.Bold,
            HorizontalAlignment =
Avalonia.Layout.HorizontalAlignment.Center
        }
    };

    children.AddRange(inputs);
    children.Add(calculateButton);

    var stackPanel = new StackPanel
    {
        Margin = new Thickness(10),
        Orientation = Orientation.Vertical,
        HorizontalAlignment =
Avalonia.Layout.HorizontalAlignment.Center
    };

```

```

foreach (var child in children)
{
    stackPanel.Children.Add(child);
}

// Обгортаємо StackPanel у ScrollViewer
var scrollView = new ScrollViewer
{
    Content = stackPanel,
    VerticalScrollBarVisibility =
ScrollBarVisibility.Auto,
    HorizontalScrollBarVisibility =
ScrollBarVisibility.Disabled
};
var uri = new
Uri("avares://CITAR_Lab_Organizer_Beta/Assets/logo-final.ico");
var icon = new WindowIcon(AssetLoader.Open(uri));
var dialog = new Window
{
    Title = "Інформація",
    Icon = icon,
    Width = 640,
    Height = 360,
    MinWidth = 640,
    MinHeight = 360,
    MaxWidth = 640,
    MaxHeight = 360,
    WindowStartupLocation =
WindowStartupLocation.CenterOwner,
    Content = scrollView
};

calculateButton.Click += (s, e) => onCalculate(inputs);

dialog.Show();
}

private void Button_Click(object? sender, RoutedEventArgs e)
{
    var dataBuilder = new StringBuilder();

    if (sender is Button button && button.Content is string
taskName)
    {
        if (button.Content == "Густина полого циліндра")
        {
            ShowInputDialog("Густина полого циліндра", ["m",
"h", "D", "d"], inputs =>
            {
                double m;

```

```

        if (!TryGetValidatedInput(inputs[0].Text, "m",
out m)) return;
        double h;
        if (!TryGetValidatedInput(inputs[1].Text, "h",
out h)) return;
        double D;
        if (!TryGetValidatedInput(inputs[2].Text, "D",
out D)) return;
        double d;
        if (!TryGetValidatedInput(inputs[3].Text, "d",
out d)) return;
        Main_HollowCylinder(m, h, D, d);
    });
}
else if (button.Content == "Густина суцільного
циліндра")
{
    ShowInputDialog("Густина суцільного циліндра",
["m", "h", "d"], inputs =>
    {
        double m;
        if (!TryGetValidatedInput(inputs[0].Text, "m",
out m)) return;
        double h;
        if (!TryGetValidatedInput(inputs[1].Text, "h",
out h)) return;
        double d;
        if (!TryGetValidatedInput(inputs[2].Text, "d",
out d)) return;
        Main_SolidCylinder(m, h, d);
    });
}
else if (button.Content == "Густина пластинки")
{
    ShowInputDialog("Густина пластинки", ["m", "a",
"b", "c"], inputs =>
    {
        double m;
        if (!TryGetValidatedInput(inputs[0].Text, "m",
out m)) return;
        double a;
        if (!TryGetValidatedInput(inputs[1].Text, "a",
out a)) return;
        double b;
        if (!TryGetValidatedInput(inputs[2].Text, "b",
out b)) return;
        double c;
        if (!TryGetValidatedInput(inputs[3].Text, "c",
out c)) return;
        Main_PlateDensity(m, a, b, c);
    });
}
}

```

```

    }
    else if (button.Content == "Густина кулі")
    {
        ShowInputDialog("Густина кулі", ["m", "D"], inputs
=>
        {
            double m;
            if (!TryGetValidatedInput(inputs[0].Text, "m",
out m)) return;

            double D;
            if (!TryGetValidatedInput(inputs[1].Text, "D",
out D)) return;

            Main_BallDensity(m, D);
        });
    }
    else if (button.Content == "Модель максимальної
узагальненої корисності")
    {
        ShowInputDialog("Введення коефіцієнтів", [
            "k+1", "k-1", "k+2", "k-2", "k+3", "k-3", "k1",
"k2", "k3", "k+-1", "k+-2", "k+-3", "k11", "k21", "k31", "k12", "k22",
"k32", "k13", "k23", "k33", "k14", "k24", "k34", "k15", "k25", "k35",
"k16", "k26", "k36", "k17", "k27", "k37", "k18", "k28", "k38", "k19",
"k29", "k39", "k110", "k210", "k310"
        ], inputs =>
        {
            double k_plus_1;
            if (!TryGetValidatedInput(inputs[0].Text,
"k+1", out k_plus_1)) return;
            double k_minus_1;
            if (!TryGetValidatedInput(inputs[1].Text, "k-
1", out k_minus_1)) return;
            double k_plus_2;
            if (!TryGetValidatedInput(inputs[2].Text,
"k+2", out k_plus_2)) return;
            double k_minus_2;
            if (!TryGetValidatedInput(inputs[3].Text, "k-
2", out k_minus_2)) return;
            double k_plus_3;
            if (!TryGetValidatedInput(inputs[4].Text,
"k+3", out k_plus_3)) return;
            double k_minus_3;
            if (!TryGetValidatedInput(inputs[5].Text, "k-
3", out k_minus_3)) return;
            double k_1;
            if (!TryGetValidatedInput(inputs[6].Text,
"k1", out k_1)) return;
            double k_2;
            if (!TryGetValidatedInput(inputs[7].Text,
"k2", out k_2)) return;
            double k_3;

```

```

        if (!TryGetValidatedInput(inputs[8].Text,
"k3", out k3)) return;
        double k_plus_minus_1;
        if (!TryGetValidatedInput(inputs[9].Text,
"k+-1", out k_plus_minus_1)) return;
        double k_plus_minus_2;
        if (!TryGetValidatedInput(inputs[10].Text,
"k+-2", out k_plus_minus_2)) return;
        double k_plus_minus_3;
        if (!TryGetValidatedInput(inputs[11].Text,
"k+-3", out k_plus_minus_3)) return;
        double k11;
        if (!TryGetValidatedInput(inputs[12].Text,
"k11", out k11)) return;
        double k21;
        if (!TryGetValidatedInput(inputs[13].Text,
"k21", out k21)) return;
        double k31;
        if (!TryGetValidatedInput(inputs[14].Text,
"k31", out k31)) return;
        double k12;
        if (!TryGetValidatedInput(inputs[15].Text,
"k12", out k12)) return;
        double k22;
        if (!TryGetValidatedInput(inputs[16].Text,
"k22", out k22)) return;
        double k32;
        if (!TryGetValidatedInput(inputs[17].Text,
"k32", out k32)) return;
        double k13;
        if (!TryGetValidatedInput(inputs[18].Text,
"k13", out k13)) return;
        double k23;
        if (!TryGetValidatedInput(inputs[19].Text,
"k23", out k23)) return;
        double k33;
        if (!TryGetValidatedInput(inputs[20].Text,
"k33", out k33)) return;
        double k14;
        if (!TryGetValidatedInput(inputs[21].Text,
"k14", out k14)) return;
        double k24;
        if (!TryGetValidatedInput(inputs[22].Text,
"k24", out k24)) return;
        double k34;
        if (!TryGetValidatedInput(inputs[23].Text,
"k34", out k34)) return;
        double k15;
        if (!TryGetValidatedInput(inputs[24].Text,
"k15", out k15)) return;
        double k25;

```

```

        if (!TryGetValidatedInput(inputs[25].Text,
"k25", out k_25)) return;
        double k_35;
        if (!TryGetValidatedInput(inputs[26].Text,
"k35", out k_35)) return;
        double k_16;
        if (!TryGetValidatedInput(inputs[27].Text,
"k16", out k_16)) return;
        double k_26;
        if (!TryGetValidatedInput(inputs[28].Text,
"k26", out k_26)) return;
        double k_36;
        if (!TryGetValidatedInput(inputs[29].Text,
"k36", out k_36)) return;
        double k_17;
        if (!TryGetValidatedInput(inputs[30].Text,
"k17", out k_17)) return;
        double k_27;
        if (!TryGetValidatedInput(inputs[31].Text,
"k27", out k_27)) return;
        double k_37;
        if (!TryGetValidatedInput(inputs[32].Text,
"k37", out k_37)) return;
        double k_18;
        if (!TryGetValidatedInput(inputs[33].Text,
"k18", out k_18)) return;
        double k_28;
        if (!TryGetValidatedInput(inputs[34].Text,
"k28", out k_28)) return;
        double k_38;
        if (!TryGetValidatedInput(inputs[35].Text,
"k38", out k_38)) return;
        double k_19;
        if (!TryGetValidatedInput(inputs[36].Text,
"k19", out k_19)) return;
        double k_29;
        if (!TryGetValidatedInput(inputs[37].Text,
"k29", out k_29)) return;
        double k_39;
        if (!TryGetValidatedInput(inputs[38].Text,
"k39", out k_39)) return;
        double k_110;
        if (!TryGetValidatedInput(inputs[39].Text,
"k110", out k_110)) return;
        double k_210;
        if (!TryGetValidatedInput(inputs[40].Text,
"k210", out k_210)) return;
        double k_310;
        if (!TryGetValidatedInput(inputs[41].Text,
"k310", out k_310)) return;

```

```

        Main_MaximumGeneralizedUtilityModel(
            k_plus_1, k_minus_1, k_plus_2, k_minus_2,
k_plus_3, k_minus_3,
            k_1, k_2, k_3, k_plus_minus_1,
k_plus_minus_2, k_plus_minus_3,
            k_11, k_21, k_31, k_12, k_22, k_32, k_13,
k_23, k_33,
            k_14, k_24, k_34, k_15, k_25, k_35, k_16,
k_26, k_36,
            k_17, k_27, k_37, k_18, k_28, k_38, k_19,
k_29, k_39,
            k_110, k_210, k_310);
    });
}
else if (button.Content == "Критерій максимального
математичного сподівання")
{
    ShowInputDialog("Критерій максимального
математичного сподівання", ["Mx1", "Mx2", "Mx3", "Mx4"], inputs =>
    {
        double M_x1;
        if (!TryGetValidatedInput(inputs[0].Text,
"Mx1", out M_x1)) return;
        double M_x2;
        if (!TryGetValidatedInput(inputs[1].Text,
"Mx2", out M_x2)) return;
        double M_x3;
        if (!TryGetValidatedInput(inputs[2].Text,
"Mx3", out M_x3)) return;
        double M_x4;
        if (!TryGetValidatedInput(inputs[3].Text,
"Mx4", out M_x4)) return;

Main_CriterionOfMaximumMathematicalExpectation(M_x1, M_x2, M_x3,
M_x4);
    });
}
else if (button.Content == "Критерій мінімальної
дисперсії")
{
    ShowInputDialog("Критерій мінімальної дисперсії",
["Dx1", "Dx2", "Dx3", "Dx4"], inputs =>
    {
        double D_x1;
        if (!TryGetValidatedInput(inputs[0].Text,
"Dx1", out D_x1)) return;
        double D_x2;
        if (!TryGetValidatedInput(inputs[1].Text,
"Dx2", out D_x2)) return;
        double D_x3;

```

```

        if (!TryGetValidatedInput(inputs[2].Text,
"Dx3", out D_x3)) return;
        double D_x4;
        if (!TryGetValidatedInput(inputs[3].Text,
"Dx4", out D_x4)) return;
        Main_CriterionOfMinimumDispersion(D_x1, D_x2,
D_x3, D_x4);
    });
}
else if (button.Content == "Критерій «очікуване
значення - дисперсія»")
{
    ShowInputDialog("Критерій «очікуване значення -
дисперсія», ["K", "Mx1", "Mx2", "Mx3", "Mx4", "Dx1", "Dx2", "Dx3",
"Dx4"], inputs =>
    {
        double K;
        if (!TryGetValidatedInput(inputs[0].Text, "K",
out K)) return;
        double M_x1;
        if (!TryGetValidatedInput(inputs[1].Text,
"Mx1", out M_x1)) return;
        double M_x2;
        if (!TryGetValidatedInput(inputs[2].Text,
"Mx2", out M_x2)) return;
        double M_x3;
        if (!TryGetValidatedInput(inputs[3].Text,
"Mx3", out M_x3)) return;
        double M_x4;
        if (!TryGetValidatedInput(inputs[4].Text,
"Mx4", out M_x4)) return;
        double D_x1;
        if (!TryGetValidatedInput(inputs[5].Text,
"Dx1", out D_x1)) return;
        double D_x2;
        if (!TryGetValidatedInput(inputs[6].Text,
"Dx2", out D_x2)) return;
        double D_x3;
        if (!TryGetValidatedInput(inputs[7].Text,
"Dx3", out D_x3)) return;
        double D_x4;
        if (!TryGetValidatedInput(inputs[8].Text,
"Dx4", out D_x4)) return;
        Main_ExpectationDispersion(K, M_x1, M_x2,
M_x3, M_x4, D_x1, D_x2, D_x3, D_x4);
    });
}
else if (button.Content == "Асимптотична частотна
характеристика")
{

```

```

        ShowInputDialog("Асимптотична частотна
характеристика", ["k", "T1", "T2", "T3", "T4", "s", "v"], inputs =>
        {
            double k;
            if (!TryGetValidatedInput(inputs[0].Text, "k",
out k)) return;
            double T1;
            if (!TryGetValidatedInput(inputs[1].Text,
"T1", out T1)) return;
            double T2;
            if (!TryGetValidatedInput(inputs[2].Text,
"T2", out T2)) return;
            double T3;
            if (!TryGetValidatedInput(inputs[3].Text,
"T3", out T3)) return;
            double T4;
            if (!TryGetValidatedInput(inputs[1].Text,
"T4", out T4)) return;
            double s;
            if (!TryGetValidatedInput(inputs[2].Text, "s",
out s)) return;
            double v;
            if (!TryGetValidatedInput(inputs[3].Text, "v",
out v)) return;
            Main_AsymptoticFrequencyResponse(k, T1, T2,
T3, T4, s, v);
        });
    }
    else if (button.Content == "Кореневий критерій та
критерій Лапласа")
    {
        ShowInputDialog("Кореневий критерій та критерій
Лапласа", ["a", "b", "c", "s"], inputs =>
        {
            double a;
            if (!TryGetValidatedInput(inputs[0].Text, "a",
out a)) return;
            double b;
            if (!TryGetValidatedInput(inputs[1].Text, "b",
out b)) return;
            double c;
            if (!TryGetValidatedInput(inputs[2].Text, "c",
out c)) return;
            double s;
            if (!TryGetValidatedInput(inputs[3].Text, "s",
out s)) return;
            Main_RootAndLaplasCriterion(a, b, c, s);
        });
    }
    else if (button.Content == "Коефіцієнт серійності")
    {

```



```

        });
    }
    else if (button.Content == "HCAB за методом Шумана та
Лямбда")
    {
        ShowInputDialog("HCAB за методом Шумана та
Лямбда", ["x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10",
"Na", "Ta", "Nβ", "Tβ", "I", "t"], inputs =>
        {
            double x1;
            if (!TryGetValidatedInput(inputs[0].Text,
"x1", out x1)) return;
            double x2;
            if (!TryGetValidatedInput(inputs[1].Text,
"x2", out x2)) return;
            double x3;
            if (!TryGetValidatedInput(inputs[2].Text,
"x3", out x3)) return;
            double x4;
            if (!TryGetValidatedInput(inputs[3].Text,
"x4", out x4)) return;
            double x5;
            if (!TryGetValidatedInput(inputs[4].Text,
"x5", out x5)) return;
            double x6;
            if (!TryGetValidatedInput(inputs[5].Text,
"x6", out x6)) return;
            double x7;
            if (!TryGetValidatedInput(inputs[6].Text,
"x7", out x7)) return;
            double x8;
            if (!TryGetValidatedInput(inputs[7].Text,
"x8", out x8)) return;
            double x9;
            if (!TryGetValidatedInput(inputs[8].Text,
"x9", out x9)) return;
            double x10;
            if (!TryGetValidatedInput(inputs[9].Text,
"x10", out x10)) return;
            double N_A;
            if (!TryGetValidatedInput(inputs[10].Text,
"Na", out N_A)) return;
            double T_A;
            if (!TryGetValidatedInput(inputs[11].Text,
"Ta", out T_A)) return;
            double N_B;
            if (!TryGetValidatedInput(inputs[12].Text,
"Nβ", out N_B)) return;
            double T_B;
            if (!TryGetValidatedInput(inputs[13].Text,
"Tβ", out T_B)) return;
        }
    }
}

```

```

        double I;
        if (!TryGetValidatedInput(inputs[14].Text,
"I", out I)) return;
        double t;
        if (!TryGetValidatedInput(inputs[15].Text,
"t", out t)) return;

        Main_NSAV1(x1, x2, x3, x4, x5, x6, x7, x8, x9,
x10, N_A, T_A, N_B, T_B, I, t);
    });
}
else if (button.Content == "HCAB за методом Мілса")
{
    ShowInputDialog("HCAB за методом Мілса", [
        "W1", "S1", "V1",
        "W2", "S2", "V2",
        "W3", "S3", "V3",
        "W4", "S4", "V4",
        "W5", "S5", "V5",
        "W6", "S6", "V6"
    ], inputs =>
    {
        double W_1;
        if (!TryGetValidatedInput(inputs[0].Text,
"W1", out W_1)) return;
        double S_1;
        if (!TryGetValidatedInput(inputs[1].Text,
"S1", out S_1)) return;
        double V_1;
        if (!TryGetValidatedInput(inputs[2].Text,
"V1", out V_1)) return;

        double W_2;
        if (!TryGetValidatedInput(inputs[3].Text,
"W2", out W_2)) return;
        double S_2;
        if (!TryGetValidatedInput(inputs[4].Text,
"S2", out S_2)) return;
        double V_2;
        if (!TryGetValidatedInput(inputs[5].Text,
"V2", out V_2)) return;

        double W_3;
        if (!TryGetValidatedInput(inputs[6].Text,
"W3", out W_3)) return;
        double S_3;
        if (!TryGetValidatedInput(inputs[7].Text,
"S3", out S_3)) return;
        double V_3;
        if (!TryGetValidatedInput(inputs[8].Text,
"V3", out V_3)) return;
    }
}

```

```

        double W_4;
        if (!TryGetValidatedInput(inputs[9].Text,
"W4", out W_4)) return;
        double S_4;
        if (!TryGetValidatedInput(inputs[10].Text,
"S4", out S_4)) return;
        double V_4;
        if (!TryGetValidatedInput(inputs[11].Text,
"V4", out V_4)) return;

        double W_5;
        if (!TryGetValidatedInput(inputs[12].Text,
"W5", out W_5)) return;
        double S_5;
        if (!TryGetValidatedInput(inputs[13].Text,
"S5", out S_5)) return;
        double V_5;
        if (!TryGetValidatedInput(inputs[14].Text,
"V5", out V_5)) return;

        double W_6;
        if (!TryGetValidatedInput(inputs[15].Text,
"W6", out W_6)) return;
        double S_6;
        if (!TryGetValidatedInput(inputs[16].Text,
"S6", out S_6)) return;
        double V_6;
        if (!TryGetValidatedInput(inputs[17].Text,
"V6", out V_6)) return;

        Main_NSAV2(W_1, S_1, V_1, W_2, S_2, V_2, W_3,
S_3, V_3, W_4, S_4, V_4, W_5, S_5, V_5, W_6, S_6, V_6);
    });
}
else if (button.Content == "HCAB за моделлю Джелінські-
Моранді")
{
    ShowInputDialog("HCAB за моделлю Джелінські-
Моранді", [
        "k", "N",
        "t1", "t2", "t3", "t4",
        "i1", "i2", "i3", "i4",
        "i"
    ], inputs =>
    {
        double k;
        if (!TryGetValidatedInput(inputs[0].Text, "k",
out k)) return;

        double N;

```

```

        if (!TryGetValidatedInput(inputs[1].Text, "N",
out N)) return;

        double t_1;
        if (!TryGetValidatedInput(inputs[2].Text,
"t_1", out t_1)) return;
        double t_2;
        if (!TryGetValidatedInput(inputs[3].Text,
"t_2", out t_2)) return;
        double t_3;
        if (!TryGetValidatedInput(inputs[4].Text,
"t_3", out t_3)) return;
        double t_4;
        if (!TryGetValidatedInput(inputs[5].Text,
"t_4", out t_4)) return;

        double i_1;
        if (!TryGetValidatedInput(inputs[6].Text,
"i_1", out i_1)) return;
        double i_2;
        if (!TryGetValidatedInput(inputs[7].Text,
"i_2", out i_2)) return;
        double i_3;
        if (!TryGetValidatedInput(inputs[8].Text,
"i_3", out i_3)) return;
        double i_4;
        if (!TryGetValidatedInput(inputs[9].Text,
"i_4", out i_4)) return;

        double i;
        if (!TryGetValidatedInput(inputs[10].Text,
"i", out i)) return;

        Main_NSAB3(k, N, t_1, t_2, t_3, t_4, i_1, i_2,
i_3, i_4, i);
    });
}
else if (button.Content == "НСАВ за моделлю Нельсона")
{
    ShowInputDialog("НСАВ за моделлю Нельсона", [
        "p_1", "p_2", "p_3", "p_4", "p_5", "p_6", "p_7",
"p_8", "p_9", "p_10",
        "p_11", "p_12", "p_13", "p_14", "p_15", "p_16",
"p_17", "p_18", "p_19", "p_20",
        "p_21", "p_22", "p_23", "p_24", "p_25", "p_26",
"p_27", "p_28", "p_29", "p_30",
        "y_1", "y_2", "y_3", "y_4", "y_5", "y_6", "y_7",
"y_8", "y_9", "y_10",
        "y_11", "y_12", "y_13", "y_14", "y_15", "y_16",
"y_17", "y_18", "y_19", "y_20",

```

```

        "y21", "y22", "y23", "y24", "y25", "y26",
"y27", "y28", "y29", "y30"
    ], inputs =>
    {
        double[] p = new double[30];
        double[] y = new double[30];

        for (int i = 0; i < 30; i++)
        {
            if (!TryGetValidatedInput(inputs[i].Text,
$"p{i + 1}", out p[i])) return;
        }

        for (int i = 0; i < 30; i++)
        {
            if (!TryGetValidatedInput(inputs[30 +
i].Text, $"y{i + 1}", out y[i])) return;
        }

        Main_NSAV4(
            p[0], p[1], p[2], p[3], p[4], p[5], p[6],
p[7], p[8], p[9],
            p[10], p[11], p[12], p[13], p[14], p[15],
p[16], p[17], p[18], p[19],
            p[20], p[21], p[22], p[23], p[24], p[25],
p[26], p[27], p[28], p[29],
            y[0], y[1], y[2], y[3], y[4], y[5], y[6],
y[7], y[8], y[9],
            y[10], y[11], y[12], y[13], y[14], y[15],
y[16], y[17], y[18], y[19],
            y[20], y[21], y[22], y[23], y[24], y[25],
y[26], y[27], y[28], y[29]
        );
    });
}

bool TryGetValidatedInput(string input, string
variableName, out double value)
{
    if (!double.TryParse(input, out value) || value <
0)
    {
        ShowFinalInfoMessage($"Помилка: значення
'{variableName}' має бути невід'ємним.\nПідказка: Для вказання
дробових чисел,\nдробову частину від цілої варто відокремлювати комою,
а не крапкою.");
        return false;
    }
    return true;
}
}

```

```
    }  
  }  
}
```

```
#pragma warning restore CS0219  
#pragma warning restore CS0252  
#pragma warning restore CS8604  
#pragma warning restore CS8602
```

## ДОДАТОК Г

### Налаштування C#-проєкту

```

<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>WinExe</OutputType>
    <TargetFramework>net8.0</TargetFramework>
    <Nullable>enable</Nullable>
    <BuiltInComInteropSupport>true</BuiltInComInteropSupport>
    <ApplicationManifest>app.manifest</ApplicationManifest>

    <AvaloniaUseCompiledBindingsByDefault>true</AvaloniaUseCompiledBinding
sByDefault>
  </PropertyGroup>

  <ItemGroup>
    <Folder Include="Models\" />
    <AvaloniaResource Include="Assets\*" />
  </ItemGroup>

  <ItemGroup>
    <PackageReference Include="Avalonia" Version="11.3.0" />
    <PackageReference Include="Avalonia.Desktop" Version="11.3.0" />
    <PackageReference Include="Avalonia.Themes.Fluent"
Version="11.3.0" />
    <PackageReference Include="Avalonia.Fonts.Inter" Version="11.3.0"
/>
    <PackageReference Include="Avalonia.Xaml.Behaviors"
Version="11.3.0" />
    <!--Condition below is needed to remove Avalonia.Diagnostics
package from build output in Release configuration.-->
    <PackageReference Condition="'$(Configuration)' == 'Debug'"
Include="Avalonia.Diagnostics" Version="11.3.0" />
    <PackageReference Include="Avalonia.ReactiveUI" Version="11.3.0"
/>
    <PackageReference Include="LiveChartsCore.SkiaSharpView.Avalonia"
Version="2.0.0-rc5.4" />
  </ItemGroup>

  <ItemGroup>
    <AvaloniaXaml Update="Views\CourseContentView.axaml">
      <CopyToOutputDirectory>Never</CopyToOutputDirectory>
    </AvaloniaXaml>
    <AvaloniaXaml Update="Views\MainWindow.axaml">
      <CopyToOutputDirectory>Never</CopyToOutputDirectory>
    </AvaloniaXaml>
  </ItemGroup>
</Project>

```

ДОДАТОК Д  
Демонстраційний матеріал

