

Міністерство освіти і науки України
НТУУ «КПІ ім. Ігоря Сікорського»
Навчально-науковий інститут атомної та теплової енергетики
Кафедра цифрових технологій в енергетиці

Лабораторна робота №7
з дисципліни «Декларативне програмування графічних
інтерфейсів»
«Розробка сторінкового/віконного WPF-застосунка»
Варіант №19

Виконав: Студент групи ТР-12

Ковальов Олександр

Перевірив: доцент, к.ф.-м.н.

Тарнавський Ю. А.

Мета роботи. Ознайомитися з основами розробки сторінки WPF-застосунку, що використовує базу даних.

Завдання 1. Освоїти основи розробки сторінкового інтерфейсу.

MainPage.xaml:

```
<Page x:Class="Lab7.Views.Pages.MainPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:Lab7.Views.Pages"
      xmlns:commands="clr-namespace:Lab7.Commands"
      mc:Ignorable="d"
      Title="MainPage" Height="550" MaxHeight="550" Width="800">
  <Page.CommandBindings>
    <CommandBinding Command="{x:Static commands:DataCommands.Undo}"
      Executed="UndoCommandBinding_Executed"
      CanExecute="UndoCommandBinding_CanExecute"/>
    <CommandBinding Command="{x:Static commands:DataCommands.Create}"
      Executed="CreateCommandBinding_Executed"
      CanExecute="CreateCommandBinding_CanExecute"/>
    <CommandBinding Command="{x:Static commands:DataCommands.Edit}"
      Executed="EditCommandBinding_Executed"
      CanExecute="EditCommandBinding_CanExecute"/>
    <CommandBinding Command="{x:Static commands:DataCommands.Save}"
      Executed="SaveCommandBinding_Executed"
      CanExecute="SaveCommandBinding_CanExecute"/>
    <CommandBinding Command="{x:Static commands:DataCommands.Find}"
      Executed="FindCommandBinding_Executed"
      CanExecute="FindCommandBinding_CanExecute"/>
    <CommandBinding Command="{x:Static commands:DataCommands.Reload}"
      Executed="ReloadCommandBinding_Executed"
      CanExecute="ReloadCommandBinding_CanExecute"/>
    <CommandBinding Command="{x:Static commands:DataCommands.Delete}"
      Executed="DeleteCommandBinding_Executed"
      CanExecute="DeleteCommandBinding_CanExecute"/>

    <CommandBinding Command="{x:Static commands:DataCommands.Report}"
      Executed="ReportCommandBinding_Executed"
      CanExecute="ReportCommandBinding_CanExecute"/>
  </Page.CommandBindings>

  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition Height="Auto" />
      <RowDefinition Height="*" />
    </Grid.RowDefinitions>

    <StackPanel Grid.Row="0">
      <Menu>
        <MenuItem Header="Action" >
          <MenuItem Header="Undo" Command="{x:Static commands:DataCommands.Undo}"/>

          <Separator />

          <MenuItem Header="Create" Command="{x:Static commands:DataCommands.Create}" />
          <MenuItem Header="Edit" Command="{x:Static commands:DataCommands.Edit}" />
          <MenuItem Header="Save" Command="{x:Static commands:DataCommands.Save}" />
          <MenuItem Header="Find" Command="{x:Static commands:DataCommands.Find}" />

          <Separator />

          <MenuItem Header="Delete" Command="{x:Static commands:DataCommands.Delete}" />
        </MenuItem>

        <MenuItem Header="Report" Command="{x:Static commands:DataCommands.Report}"/>
      </Menu>

      <ToolBar>
        <Button Command="{x:Static commands:DataCommands.Undo}" ToolTip="Undo" Margin="5">
          <Image Width="32" Height="32" Source="../../Resources/Icons/UndoIcon.png" />
        </Button>

        <Button Command="{x:Static commands:DataCommands.Create}" ToolTip="Create"
          Margin="5">
          <Image Width="32" Height="32" Source="../../Resources/Icons/CreateIcon.png" />
        </Button>
      </ToolBar>
    </StackPanel>
  </Grid>
</Page>
```

```

        <Button Command="{x:Static commands:DataCommands.Edit}" ToolTip="Edit" Margin="5">
            <Image Width="32" Height="32" Source="../../Resources/Icons/EditIcon.png" />
        </Button>

        <Button Command="{x:Static commands:DataCommands.Save}" ToolTip="Save" Margin="5">
            <Image Width="32" Height="32" Source="../../Resources/Icons/SaveIcon.png" />
        </Button>

        <Button Command="{x:Static commands:DataCommands.Find}" ToolTip="Find" Margin="5">
            <Image Width="32" Height="32" Source="../../Resources/Icons/FindIcon.png" />
        </Button>

        <Button Command="{x:Static commands:DataCommands.Reload}" ToolTip="Reload"
Margin="5">
            <Image Width="32" Height="32" Source="../../Resources/Icons/ReloadIcon.png" />
        </Button>

        <Button Command="{x:Static commands:DataCommands.Delete}" ToolTip="Delete"
Margin="5">
            <Image Width="32" Height="32" Source="../../Resources/Icons/DeleteIcon.png" />
        </Button>

        <Separator />

        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="*" />
                <ColumnDefinition Width="*" />
            </Grid.ColumnDefinitions>
            <Grid.RowDefinitions>
                <RowDefinition Height="*" />
                <RowDefinition Height="*" />
            </Grid.RowDefinitions>

            <Label Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2"
                VerticalAlignment="Center" Content="Hyperlinks:" />

            <Button Grid.Row="1" Grid.Column="0"
                ToolTip="Navigate to Books.." Click="HyperlinkBooks_OnClick" Margin="5"
                Background="Transparent" BorderThickness="0">
                <Image Width="20" Height="20" Source="../../Resources/Icons/BooksIcon.png" />
            </Button>

            <Button Grid.Row="1" Grid.Column="1"
                ToolTip="Navigate to Publishers.." Click="HyperlinkPublishers_OnClick"
Margin="5"
                Background="Transparent" BorderThickness="0">
                <Image Width="20" Height="20" Source="../../Resources/Icons/PublishersIcon.png" />
            </Button>
        </Grid>

        <Separator />
    </ToolBar>
</StackPanel>

    <DataGrid Grid.Row="1" x:Name="JoinedGrid"
        AutoGenerateColumns="True"
        ItemsSource="{Binding}" />
</DataGrid>
</Grid>
</Page>

```

Завдання 2. Освоїти основи розробки бізнес-логіки програми.

MainPage.xaml.cs:

```

using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;
using Lab7.Context;
using Lab7.Repositories;
using Lab7.Services;
using Lab7.Views.Forms;

namespace Lab7.Views.Pages;

public partial class MainPage : Page, IDisposable
{

```

```

private bool _canUndo;

private readonly Frame _frame;
private readonly Dictionary<string, Page> _pages;
private readonly LibraryDbContext _dbContext;

public MainPage(Frame frame, Dictionary<string, Page> pages)
{
    InitializeComponent();

    _frame = frame;
    _pages = pages;
    _dbContext = new LibraryDbContext();

    InitializeJoinedTable();
}

private void InitializeJoinedTable()
{
    var joined = _dbContext.Books.Join(_dbContext.Publishers,
        x => x.PublisherCode,
        y => y.Id,
        (x, y) => new
        {
            x.Isbn,
            x.Title,
            x.Authors,
            Publisher = y.Name,
            x.PublicationYear
        });

    JoinedGrid.DataContext = joined.ToList();
}

private void UndoCommandBinding_CanExecute(object sender, CanExecuteRoutedEventArgs e) {
    e.CanExecute = _canUndo;
}

private void UndoCommandBinding_Executed(object sender, ExecutedRoutedEventArgs e)
{
    OperationContext.Pop();
    _canUndo = OperationContext.Length > 0;
}

private void CreateCommandBinding_CanExecute(object sender, CanExecuteRoutedEventArgs e) {
    e.CanExecute = true;
}

private void CreateCommandBinding_Executed(object sender, ExecutedRoutedEventArgs e) {
    new CreateForm(_dbContext).Show();
}

private void EditCommandBinding_CanExecute(object sender, CanExecuteRoutedEventArgs e)
{
    e.CanExecute = true;
}

private void EditCommandBinding_Executed(object sender, ExecutedRoutedEventArgs e)
{
    var item = JoinedGrid.SelectedItem;

    if (item is null)
    {
        MessageBox.Show(messageBoxText: "Choose some row if you want to edit it!",
            caption: "Error",
            button: MessageBoxButton.OK,
            icon: MessageBoxImage.Error,
            defaultResult: MessageBoxResult.OK);

        return;
    }

    dynamic dynamicItem = item;
    var isbn = (string)dynamicItem.Isbn;

    new EditForm(_dbContext, isbn).Show();
}

private void SaveCommandBinding_CanExecute(object sender, CanExecuteRoutedEventArgs e) {
    e.CanExecute = true;
}

private void SaveCommandBinding_Executed(object sender, ExecutedRoutedEventArgs e)
{
    _dbContext.SaveChanges();
}

```

```

        MessageBox.Show(messageBoxText: "Changes saved in database",
            caption: "Success!",
            button: MessageBoxButton.OK,
            icon: MessageBoxImage.Information,
            defaultResult: MessageBoxResult.OK);
    }

    private void FindCommandBinding_CanExecute(object sender, CanExecuteRoutedEventArgs e) {
        e.CanExecute = true;
    }
    private void FindCommandBinding_Executed(object sender, ExecutedRoutedEventArgs e) {
        new FindForm(_dbContext, JoinedGrid).Show();
    }

    private void ReloadCommandBinding_CanExecute(object sender, CanExecuteRoutedEventArgs e)
    {
        e.CanExecute = true;
    }
    private void ReloadCommandBinding_Executed(object sender, ExecutedRoutedEventArgs e)
    {
        InitializeJoinedTable();
        _canUndo = OperationContext.Length > 0;
    }

    private void DeleteCommandBinding_CanExecute(object sender, CanExecuteRoutedEventArgs e) {
        e.CanExecute = true;
    }
    private void DeleteCommandBinding_Executed(object sender, ExecutedRoutedEventArgs e) {
        var item = JoinedGrid.SelectedItem;

        if (item is null)
        {
            MessageBox.Show(messageBoxText: "Choose some row if you want to edit it!",
                caption: "Error",
                button: MessageBoxButton.OK,
                icon: MessageBoxImage.Error,
                defaultResult: MessageBoxResult.OK);

            return;
        }

        dynamic dynamicItem = item;
        var isbn = (string)dynamicItem.Isbn;

        try
        {
            new BooksRepository(_dbContext).Delete(isbn);
        }
        catch (Exception exception)
        {
            MessageBox.Show(messageBoxText: exception.Message,
                caption: "Error",
                button: MessageBoxButton.OK,
                icon: MessageBoxImage.Error,
                defaultResult: MessageBoxResult.OK);
        }

        ReloadCommandBinding_Executed(null!, null!);
    }

    private void ReportCommandBinding_CanExecute(object sender, CanExecuteRoutedEventArgs e)
    {
        e.CanExecute = true;
    }
    private void ReportCommandBinding_Executed(object sender, ExecutedRoutedEventArgs e)
    {
        const string filename = "report.txt";

        var reportService = new ReportService(filename);
        var text = reportService.ToCsvString(JoinedGrid);
        reportService.ToTextFile(text);

        MessageBox.Show(messageBoxText: $"Data written to file {filename}!",
            caption: "Success | Report",
            button: MessageBoxButton.OK,
            icon: MessageBoxImage.Information,
            defaultResult: MessageBoxResult.OK);
    }
}

```

```

private void HyperlinkBooks_OnClick(object sender, RoutedEventArgs e)
{
    _frame.Navigate(_pages["BooksPage"]);

    var page = (BooksPage) _pages["BooksPage"];
    page.ReloadGrid();
}

private void HyperlinkPublishers_OnClick(object sender, RoutedEventArgs e)
{
    _frame.Navigate(_pages["PublishersPage"]);

    var page = (PublishersPage) _pages["PublishersPage"];
    page.ReloadGrid();
}

public void Dispose()
{
    _dbContext.Dispose();
}
}

```

DataCommands.cs:

```

using System.Windows.Input;

namespace Lab7.Commands;

public static class DataCommands
{
    public static RoutedCommand Undo { get; }
    public static RoutedCommand Create { get; }
    public static RoutedCommand Edit { get; }
    public static RoutedCommand Save { get; }
    public static RoutedCommand Find { get; }
    public static RoutedCommand Reload { get; }
    public static RoutedCommand Delete { get; }
    public static RoutedCommand Report { get; }

    static DataCommands() {
        Undo = new RoutedCommand("Delete", typeof(DataCommands),
            [new KeyGesture(Key.Z, ModifierKeys.Control, "Ctrl+Z")]);

        Create = new RoutedCommand("Create", typeof(DataCommands),
            [new KeyGesture(Key.N, ModifierKeys.Control, "Ctrl+N")]);

        Edit = new RoutedCommand("Edit", typeof(DataCommands),
            [new KeyGesture(Key.E, ModifierKeys.Control, "Ctrl+E")]);

        Save = new RoutedCommand("Save", typeof(DataCommands),
            [new KeyGesture(Key.S, ModifierKeys.Control, "Ctrl+S")]);

        Find = new RoutedCommand("Find", typeof(DataCommands),
            [new KeyGesture(Key.F, ModifierKeys.Control, "Ctrl+F")]);

        Reload = new RoutedCommand("Reload", typeof(DataCommands),
            [new KeyGesture(Key.R, ModifierKeys.Control, "Ctrl+R")]);

        Delete = new RoutedCommand("Delete", typeof(DataCommands),
            [new KeyGesture(Key.D, ModifierKeys.Control, "Ctrl+D")]);

        Report = new RoutedCommand("Report", typeof(DataCommands),
            [new KeyGesture(Key.P, ModifierKeys.Control, "Ctrl+P")]);
    }
}

```

Github Repository: <https://github.com/xairaven/KPI-DPGI/tree/Lab7/Lab7>

Висновок: за результатами виконання цієї лабораторної роботи були набуті практичні навички розробки сторінкового/віконного додатку використовуючи фреймворк WPF.