

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

Лабораторна робота №4  
з дисципліни «Декларативне програмування графічних  
інтерфейсів»  
«Організація взаємодії з базою даних за допомогою  
ADO.NET»  
Варіант №19

Виконав: Студент групи ТР-12  
Ковальов Олександр  
Перевірів: доцент, к.ф.-м.н.  
Тарнавський Ю. А.

**Мета роботи.** Розробити програму для перегляду даних, що зберігаються в SQL-базі даних.

**Завдання 1. Створення простої однотабличної бази даних.**

Створіть нову SQL базу даних, що складається з однієї таблиці, відповідно до свого варіанта завдання (Книги). Поля: ISBN, назва, автори, видавництво, рік видання.

Скрипт:

```
USE DPGI
GO;

CREATE TABLE dbo.Books
(
    ISBN          VARCHAR(30),
    Title          VARCHAR(255),
    Authors        VARCHAR(255),
    Publisher       VARCHAR(255),
    PublicationYear SMALLINT
)
GO;
```

Додайте дані до бази даних:

```
USE DPGI
GO;

CREATE TABLE #BulkBooks
(
    isbn          VARCHAR(30),
    title          VARCHAR(255),
    authors        VARCHAR(255),
    publisher       VARCHAR(255),
    publication_year SMALLINT
)
BULK INSERT #BulkBooks
FROM 'D:\Programming\KPI-DPGI\Lab4\Database\Data\Data.csv'
WITH (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2
);
INSERT INTO Books
SELECT * FROM #BulkBooks;
DROP TABLE IF EXISTS #BulkBooks;
GO;
```

**Завдання 2. Використання ADO.net та розробка класу доступу до бази даних.**

Додайте в проект файл класу доступу до бази даних і опишіть метод для отримання даних з бази даних за допомогою DataTable.

```
private static string? _connectionString;

public AdoWrapper()
{
    try
    {
        if (_connectionString is null)
            throw new SettingsPropertyNotFoundException("Connection string is uninitialized");

        var connection = new SqlConnection(connectionString: _connectionString);

        connection.Open();
    }
    catch (SqlException)
    {
        MessageBox.Show(messageBoxText: "No connection with SQL Database.",
            caption: "Error!",
            button: MessageBoxButton.OK,
            icon: MessageBoxImage.Error,
            defaultResult: MessageBoxResult.OK);
    }
}
```

```

        Application.Current.Shutdown();
    }
}

public async Task<DataTable> ExecuteReader(string query)
{
    await using var connection = new SqlConnection(_connectionString);
    await connection.OpenAsync();

    var adapter = new SqlDataAdapter(query, connection);
    var table = new DataTable();
    adapter.Fill(table);

    return table;
}

```

### Завдання 3. Розробка інтерфейсу програми та прив'язування даних.

Відкрийте XAML-файл програми та опишіть у ньому інтерфейс програми для перегляду бази даних відповідно до свого варіанта завдання. У XAML-кодi забезпечте прив'язку даних.

```

<Window x:Class="Lab4.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    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:local="clr-namespace:Lab4"
    mc:Ignorable="d"
    Loaded="MainWindow_OnLoaded"
    Title="Lab4" Height="450" Width="800">
    <Grid
        DataContext="{Binding ElementName=listFullName, Path=SelectedItem}">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="Auto" />
            <ColumnDefinition />
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
        </Grid.RowDefinitions>

        <ListBox Grid.Row="0"
            Grid.Column="0"
            Grid.ColumnSpan="2"
            Margin="0,0,0,3"
            ScrollViewer.VerticalScrollBarVisibility="Auto"
            Name="MainListBox"
            ItemsSource="{Binding}"
            SelectedItem="{Binding SelectedBook}"
            DisplayMemberPath="title" />

        <TextBlock Grid.Column="0" Grid.Row="1" Margin="5">ISBN:</TextBlock>
        <TextBox Grid.Row="1" Grid.Column="1"
            Text="{Binding SelectedItem.isbn, ElementName=MainListBox, Mode=OneWay}"
            Focusable="True" />

        <TextBlock Grid.Column="0" Grid.Row="2" Margin="5">Title:</TextBlock>
        <TextBox Grid.Row="2" Grid.Column="1"
            Text="{Binding SelectedItem.title, ElementName=MainListBox, Mode=OneWay}"
            Focusable="True" />

        <TextBlock Grid.Column="0" Grid.Row="3" Margin="5">Authors:</TextBlock>
        <TextBox Grid.Row="3" Grid.Column="1"
            Text="{Binding SelectedItem.authors, ElementName=MainListBox, Mode=OneWay}"
            Focusable="True" />

        <TextBlock Grid.Column="0" Grid.Row="4" Margin="5">Publisher:</TextBlock>
        <TextBox Grid.Row="4" Grid.Column="1"
            Text="{Binding SelectedItem.publisher, ElementName=MainListBox, Mode=OneWay}"
            Focusable="True" />
    </Grid>

```

```

        <TextBlock Grid.Column="0" Grid.Row="5" Margin="5">Publication Year:</TextBlock>
        <TextBox Grid.Row="5" Grid.Column="1"
            Text="{Binding SelectedItem.publication_year, ElementName=MainListBox, Mode=OneWay}"
            Focusable="True"
        />
    </Grid>

</Window>

```

У файлі приєднаного коду напишіть метод для отримання контексту даних.

```

private async void MainWindow_OnLoaded(object sender, RoutedEventArgs e)
{
    var wrapper = new AdoWrapper();

    const string query = "SELECT * FROM Books";

    var table = await wrapper.ExecuteReader(query);

    MainListBox.SelectedIndex = 0;
    MainListBox.Focus();
    MainListBox.DataContext = table;
}

```

#### Завдання 4. Реалізація операцій додавання, оновлення і видалення записів бази даних.

Доповніть клас доступу до бази даних (AdoAssistant.class з завдання 2) методами для додавання, оновлення і видалення записів бази даних на основі відповідних SQL-запитів типу «NonQuery».

```

public async Task<int> ExecuteNonQuery(string query)
{
    int rowsAffected;

    await using (var connection = new SqlConnection(_connectionString))
    {
        await connection.OpenAsync();

        var command = new SqlCommand(cmdText: query,
            connection: connection);

        rowsAffected = await command.ExecuteNonQueryAsync();
    }

    return rowsAffected;
}

```

В кодї форми застосунку (XAML-файлі розроблений в завданні 3) весь вміст контейнера Grid перемістіть в створений для цього контейнер DockPanel. Додайте в DockPanel панель інструментів ToolBar з трьома кнопками: Create, Update та Delete.

```

<Window x:Class="Lab4.Views.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    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:local="clr-namespace:Lab4.Views"
    mc:Ignorable="d"
    Loaded="MainWindow_OnLoaded"
    WindowStartupLocation="CenterScreen"
    Title="Lab4" Height="450" Width="800">
    <DockPanel>
        <ToolBar Height="25" VerticalAlignment="Top" DockPanel.Dock="Top">
            <Button Click="ButtonCreate_OnClick">Create</Button>
            <Separator />
            <Button Click="ButtonUpdate_OnClick">Update</Button>
            <Separator />
            <Button Click="ButtonDelete_OnClick">Delete</Button>
        </ToolBar>
    </DockPanel>

```

```

</ToolBar>

<Grid
    DataContext="{Binding ElementName=listFullName, Path=SelectedItem}">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="Auto" />
        <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="*" />
        <RowDefinition Height="Auto" />
        <RowDefinition Height="Auto" />
        <RowDefinition Height="Auto" />
        <RowDefinition Height="Auto" />
        <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>

    <ListBox Grid.Row="0"
        Grid.Column="0"
        Grid.ColumnSpan="2"
        Margin="0,0,0,3"
        ScrollViewer.VerticalScrollBarVisibility="Auto"
        Name="MainListBox"
        ItemsSource="{Binding}"
        SelectedItem="{Binding SelectedBook}"
        DisplayMemberPath="title" />

    <TextBlock Grid.Column="0" Grid.Row="1" Margin="5">ISBN:</TextBlock>
    <TextBox Grid.Row="1" Grid.Column="1"
        Text="{Binding SelectedItem.isbn, ElementName=MainListBox, Mode=OneWay}"
        IsReadOnly="True"
        Focusable="True" />

    <TextBlock Grid.Column="0" Grid.Row="2" Margin="5">Title:</TextBlock>
    <TextBox Grid.Row="2" Grid.Column="1"
        Text="{Binding SelectedItem.title, ElementName=MainListBox, Mode=OneWay}"
        IsReadOnly="True"
        Focusable="True" />

    <TextBlock Grid.Column="0" Grid.Row="3" Margin="5">Authors:</TextBlock>
    <TextBox Grid.Row="3" Grid.Column="1"
        Text="{Binding SelectedItem.authors, ElementName=MainListBox, Mode=OneWay}"
        IsReadOnly="True"
        Focusable="True" />

    <TextBlock Grid.Column="0" Grid.Row="4" Margin="5">Publisher:</TextBlock>
    <TextBox Grid.Row="4" Grid.Column="1"
        Text="{Binding SelectedItem.publisher, ElementName=MainListBox, Mode=OneWay}"
        IsReadOnly="True"
        Focusable="True" />

    <TextBlock Grid.Column="0" Grid.Row="5" Margin="5">Publication Year:</TextBlock>
    <TextBox Grid.Row="5" Grid.Column="1"
        Text="{Binding SelectedItem.publication_year, ElementName=MainListBox,
Mode=OneWay}"
        IsReadOnly="True"
        Focusable="True" />

    </Grid>
</DockPanel>
</Window>

```

До кожної з кнопок панелі інструментів додайте обробник події Click для виконання відповідної операції. Сутність операції полягає у виконанні SQL-запиту типу «NonQuery», необхідні параметри якого беруться з форми застосунку, та зчитування отриманої таблиці з бази даних.

```

private void ButtonCreate_OnClick(object sender, RoutedEventArgs e)
{
    new CreateForm(this).Show();
}

private void ButtonUpdate_OnClick(object sender, RoutedEventArgs e)
{
    var selectedItem = MainListBox.SelectedItem as DataRowView;

```

```

        if (selectedItem is null)
        {
            MessageBox.Show(messageBoxText: "Select item if you want update it, please.",
                caption: "Error!",
                button: MessageBoxButton.OK,
                icon: MessageBoxImage.Error,
                defaultResult: MessageBoxResult.OK);

            return;
        }

        new UpdateForm(selectedItem, this).Show();
    }

    private async void ButtonDelete_OnClick(object sender, RoutedEventArgs e)
    {
        var selectedItem = MainListBox.SelectedItem as DataRowView;

        if (selectedItem is null)
        {
            MessageBox.Show(messageBoxText: "Select item if you want delete it, please.",
                caption: "Error!",
                button: MessageBoxButton.OK,
                icon: MessageBoxImage.Error,
                defaultResult: MessageBoxResult.OK);

            return;
        }

        var isbn = selectedItem["isbn"].ToString();

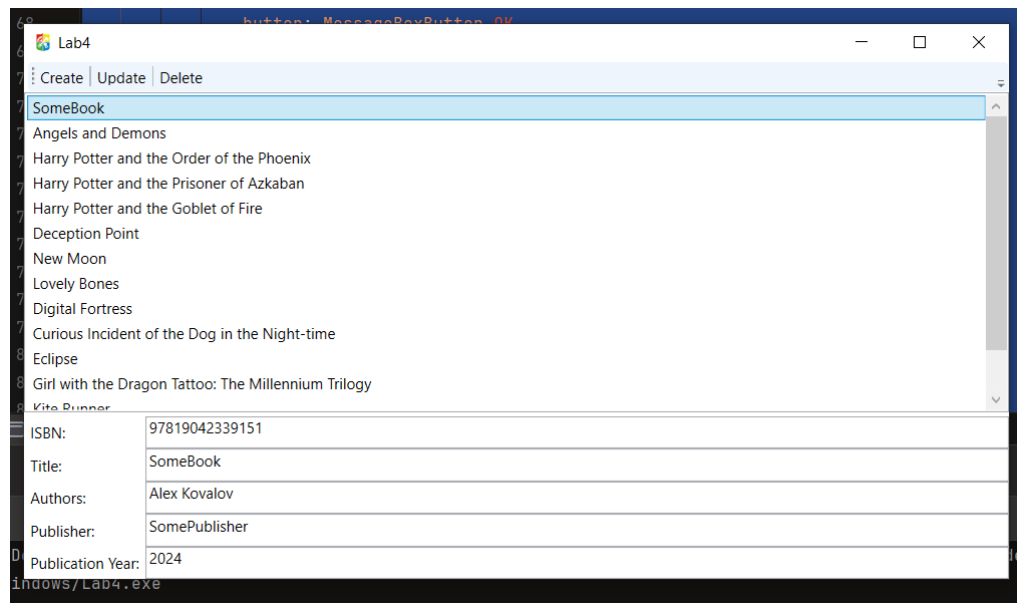
        var query = $"DELETE FROM Books WHERE isbn = {isbn}";
        var affectedRows = await _wrapper.ExecuteNonQuery(query);

        MessageBox.Show(messageBoxText: $"Affected rows: {affectedRows}",
            caption: "Delete operation results",
            button: MessageBoxButton.OK,
            icon: MessageBoxImage.Information,
            defaultResult: MessageBoxResult.OK);

        await LoadListBox();
    }
}

```

Перевірте працездатність застосунку:



**Висновок:** за результатами виконання цієї лабораторної роботи були набуті практичні навички роботи з елементом ListBox та фреймворком ADO.NET. Була проведена робота зі зв'язування додатку з базою даних SQL Server.