

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк __ / 35

## Лабораторна робота №6. Обробка даних у WPF: створення списків і прив'язка до елементів інтерфейсу

**Мета роботи:** Навчитися працювати з даними у WPF-додатках, додавати їх у список із використанням `ObservableCollection`, забезпечувати валідацію введення, прив'язувати дані до таблиць (`DataGrid`, `ListView`) і налаштовувати обробку подій у графічному інтерфейсі.

### Теоретичні відомості

У WPF обробка даних є невід'ємною частиною створення інтерактивних додатків. Для цього використовується механізм прив'язки даних (`Data Binding`), який дозволяє автоматично синхронізувати інформацію між джерелом даних та елементами інтерфейсу. Прив'язка забезпечує легкість візуалізації динамічних даних у таких елементах, як таблиці (`DataGrid`) або списки (`ListView`), і дозволяє спростити обробку змін у даних. Зазвичай дані зберігаються у структурах або класах, які представляють об'єкт із певним набором полів. Для роботи зі списками об'єктів у WPF найчастіше використовується `ObservableCollection`. Ця колекція повідомляє інтерфейс про всі зміни, такі як додавання, видалення або оновлення елементів. Завдяки цьому, при зміні вмісту списку, елементи управління, прив'язані до цієї колекції, оновлюються автоматично. Приклад прив'язки до колекції:

```
<ListBox ItemsSource="{Binding Names}" />
```

#### С# код:

```
public partial class MainWindow : Window
{
    public ObservableCollection<string> Names { get; set; }

    public MainWindow()
    {
        InitializeComponent();
        Names = new ObservableCollection<string> { "Анна", "Іван", "Олег" };
        DataContext = this;
    }
}
```

Тут `ListBox` автоматично відображає елементи з колекції `Names`.

Події у WPF дозволяють реагувати на дії користувача, наприклад, натискання кнопок або введення тексту. Події можна підключити безпосередньо у XAML або в коді. Приклад обробки події натискання кнопки:

```
<Button Content="Зберегти" Click="SaveButton_Click" />
```

#### С# код:

```
private void SaveButton_Click(object sender, RoutedEventArgs e)
{
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк __/36

```
MessageBox.Show("Дані збережено!");  
}
```

Ще однією важливою складовою є валідація даних, яка забезпечує коректність введення. Валідація дозволяє перевіряти, чи всі обов'язкові поля заповнені, і чи відповідають введені значення заданим критеріям. У разі помилки валідації користувач отримує відповідне повідомлення, наприклад, через MessageBox.

## Зміст роботи

Згідно з обраним у лабораторній роботі 5 варіантом теми, необхідно виконати наступну дію. Всі дії відбуваються в репозиторії OOPWPFProject.

**Завдання 1:** Злити гілку feature/add-form в основну гілку (master або main)

1.1 Створити нові гілку

```
git checkout -b feature/connect-struct
```

**Завдання 2:** Перенести структуру з 4 роботи, внести зміни відповідно до завдання: визначити необов'язкові поля згідно з таблицею в лабораторній роботі 5. Необов'язкові поля зробити nullable.

**Завдання 3:** Додати обробку подій.

4.1 Додавати введені дані до списку. Зберігати дані в ObservableCollection.

4.2 Реалізувати видалення елемента

**Завдання 4:** Забезпечити валідацію введення. Перевіряти заповнення обов'язкових полів. Реалізувати виведення повідомлення про помилку, якщо введені некоректні дані або обов'язкове поле незаповнене.

**Завдання 5:** Реалізувати виведення даних у вигляді таблиці (наприклад DataGridView або ListView компоненті). Забезпечити прив'язку даних.

**Завдання 6:** Створити коміт та завантажити зміни в віддалений репозиторій

## Контрольні запитання

1. Що таке Windows Presentation Foundation (WPF), і для чого використовується ця технологія?
2. Які основні переваги використання WPF для розробки інтерфейсу користувача?
3. Що таке прив'язка даних (Data Binding) у WPF і які її основні механізми?
4. Що таке ObservableCollection, і чому вона використовується для роботи з даними в WPF?
5. Які компоненти WPF можна використовувати для виведення даних у вигляді таблиці? Чим вони відрізняються?

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк __ / 37

6. Що станеться, якщо спробувати змінити дані у звичайному списку (List<T>), прив'язаному до таблиці? Чому?
7. Що таке елементи управління (controls) у WPF, і яку роль вони виконують?
8. Як працює TextBox для введення тексту? Як перевірити, чи текст заповнений?
9. У чому перевага використання властивості ItemsSource для прив'язки даних до компонентів?
10. Як працює MessageBox, і як його можна використовувати для відображення повідомлень у WPF?