Лабораторная работа №10
Изучение особенностей элементов отображения списков в приложениях WPF

1Цель работы

1.1 Изучить свойства и процесс обработки событий элементов отображения списков в приложениях WPF.

2Литература

2.1 https://metanit.com/sharp/wpf/5.7.php
2.2 https://metanit.com/sharp/wpf/5.8.php

3Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).
3.2 Изучить описание лабораторной работы.

4Основное оборудование

4.1 Персональный компьютер.

5Задание

ListBox – простой список – предназначен для отображения раскрытого списка.
Особенности:
- содержит коллекцию элементов **Items** типа **ListBoxItem**
- вместо элементов ListBoxItem или внутри ListBoxItem могут быть указаны другие типы элементов управления (включая StackPanel)
- допускает множественный выбор, если свойству **SelectionMode** присвоить значение **Multiple** (для выбора используется нажатие) или **Extended** (для выбора используются нажатие и Shift и/или Ctrl)
- для выбора элемента нужно указать у него атрибут **IsSelected**="True"
- для получения выделенного элемента используется **SelectedItem**
- для получения всех выделенных элементов используется коллекция **SelectedItems**
- если нужно определить, с какого элемента был снят выбор, можно воспользоваться свойством **RemovedItems** объекта **SelectionChangedEventArgs**.

ComboBox – комбинированный или выпадающий список.
Особенности:
- содержит коллекцию элементов **Items** типа **ComboBoxItem**.
- вместо элементов ComboBoxItem или внутри ComboBoxItem могут быть указаны другие типы элементов управления (включая StackPanel)
- установка свойства **IsEditable**="True" позволяет вводить в поле списка начальные символы, а затем функция автозаполнения подставит подходящий результат
- для выбора элемента нужно указать у него атрибут **IsSelected**="True"
- для получения выделенного элемента используется **SelectedItem**.

Привязка данных к ComboBox и ListBox

контрол.ItemsSource = список; // ItemsSource – аналог DataSource
контрол.DisplayMemberPath = "Свойство объекта"; // отображаемое свойство

Пример приведения выбранного элемента к требуемому типу:

Тип объект = (Тип)**контрол**.SelectedItem;

Пример перебора всех элементов типа CheckBox в списке:

```
foreach (CheckBox item in контрол.Items)
{
    // …
}
```

5.1 Создание в WPF аналога CheckedListBox (раскрытого списка с флажками)

5.1.1 Создать WPF-приложение.

5.1.2 Добавить на форму ListBox и заполнить в дизайнере его коллекцию Items элементами типа CheckBox (в списке должно быть как минимум 5 элементов), у каждого CheckBox указать свое значение свойства Content.

5.1.3 Добавить на форму кнопку, при нажатии на которую выводить значения элементов списка, отмеченных флажком.

5.2 Привязка ListBox к списку пользовательского типа данных

5.2.1 Создать WPF-приложение.

5.2.2 Добавить в приложение класс User для хранения идентификатора, логина и пароля пользователя.

5.2.3 В конструкторе после инициализации компонентов:

- заполнить коллекцию List<User> данными
- у ListBox указать в качестве источника данных заполненный список
- у ListBox настроить DisplayMemberPath так, чтобы в списке отображался логин пользователя

5.2.4 Предоставить пользователю возможность выбора более одного пользователя. При выборе элемента отображать на форме логины всех выбранных пользователей.

5.3 Привязка ComboBox к списку пользовательского типа данных

5.3.1 Создать WPF-приложение.

5.3.2 Добавить в приложение класс User для хранения идентификатора, логина и пароля пользователя.

5.3.3 В конструкторе после инициализации компонентов:

- заполнить коллекцию List<User> данными
- у ComboBox указать в качестве источника данных заполненный список
- у ComboBox настроить DisplayMemberPath так, чтобы в списке отображался логин пользователя

5.3.4 При выборе элемента в выпадающем списке отображать в полях ввода или метках на форме значения идентификатора, логина и пароля пользователя.

5.4 Программное добавление элементов в ListBox

5.4.1 Создать WPF-приложение. list

5.4.2 Добавить на форму ListBox, поле ввода и кнопку «Добавить».

5.4.3 При нажатии на кнопку нужно добавлять элемент в коллекцию Items элемента ListBox:

```
список.Items.Add(значение);
```

5.5 Смена внешнего вида ComboBox

5.5.1 Создать WPF-приложение. list

5.5.2 Добавить на форму ComboBox, в котором разместить набор элементов ComboBoxItem:

- внутри первого указать TextBlock (нередактируемый текстовый блок)
- внутри второго указать StackPanel с CheckBox и изображением. Пример настройки изображения: <Image Source="имя файла" Width="ширина" />
- внутри третьего указать StackPanel с двумя элементами TextBlock, ориентация StackPanel – горизонтальная
- внутри четвертого указать StackPanel с двумя элементами TextBlock, ориентация StackPanel – вертикальная

5.5.3 В дизайнере выбрать последний элемент, используя IsSelected.

5.5.4 Сменить фон элементов ComboBox (у каждого указать свой).

6Порядок выполнения работы

6.1 Выполнить все задания из п.5 в одном решении LabWork9. Каждый проект – приложение WPF.

6.2 Ответить на контрольные вопросы.

7Содержание отчета

7.1 Титульный лист
7.2 Цель работы
7.3 Ответы на контрольные вопросы
7.4 Вывод

8Контрольные вопросы

8.1 Что такое ComboBox и для чего он используется?
8.2 Что такое ListBox и для чего он используется?
8.3 Какое событие срабатывает при выборе элемента в ComboBox и ListBox?
8.4 В каком свойстве хранятся элементы ComboBox и ListBox?
8.5 Какого типа элементы могут быть в ComboBox и ListBox?
8.6 Какое свойство позволяет привязать ComboBox и ListBox к набору данных?
8.7 Для чего используется свойство DisplayMemberPath в ComboBox и ListBox