Лабораторная работа №33
Разработка приложения для отображения данных в табличном виде

1Цель работы

1.1 Изучить свойства и процесс настройки внешнего вида элемента DataGrid в приложениях WPF.

2Литература

2.1 https://metanit.com/sharp/wpf/5.14.php - DataGrid

3Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4Основное оборудование

4.1 Персональный компьютер.

5Задание

DataGrid – отображает данные в табличном виде с возможностью редактирования содержимого таблицы.

Данные отображаются в столбцах, которые описывает разработчик (если отключено свойство AutoGenerateColumns)

Столбцы указываются между тэгами <DataGrid.Columns> и </DataGrid.Columns>:

Типы столбцов:

DataGridTextColumn — TextBlock (при чтении) или TextBox (при редактировании)

DataGridHyperlinkColumn — гиперссылка с возможностью перехода

DataGridCheckBoxColumn — CheckBox

DataGridComboBoxColumn — ComboBox (список будет отображаться при редактировании)

DataGridTemplateColumn — позволяет задать шаблон для отображения столбца (например, кнопку с обработчиком события)

Свойства DataGrid:

AutoGenerateColumns — автогенерация столбцов на основе свойств источника данных

IsReadOnly — запрет редактирования данных

CanUserAddRows — разрешение пользователю добавлять строки

RowBackground (для всех строк) и **AlternatingRowBackground** (для четных строк) — фон строк. Если установлены оба свойства, цветовой фон чередуется.

RowHeight — высота строк.

ColumnHeaderHeight — высота строки названий столбцов.

ColumnWidth — ширина столбцов.

GridLinesVisibility — видимость линий, разделяющих столбцы и строки: All (все), Horizontal (только горизонтальные), Vertical (только вертикальные), None (без линий).

HeadersVisibility — видимость заголовков строк и столбцов

HorizontalGridLinesBrush и **VerticalGridLinesBrush** — цвет горизонтальных и вертикальных линий

Создание шаблона столбца DataGridTemplateColumn:

```
<DataGridTemplateColumn>
        <DataGridTemplateColumn.CellTemplate>
          <DataTemplate>
            <ТипЭлемента Атрибут="значение" />
          </DataTemplate>
        </DataGridTemplateColumn.CellTemplate>
</DataGridTemplateColumn>
```

Привязка данных к DataGrid

контрол.ItemsSource = список;

// **привязка свойства объекта к столбцу DataGrid в разметке XAML**

<DataGridTextColumn Header="**ЗаголовокСтолбца**"
Binding="{Binding **СвойствоОбъекта** }"/>

В атрибуте Binding кроме привязки можно указать формат отображения данных:

Binding="{Binding **СвойствоОбъекта**, StringFormat={}{**0:формат**}"

Пример денежного формата: C

5.1 Отображение данных в ListView

5.1.1 Создать WPF-приложение.

5.1.2 Добавить в приложение класс Game:

```
class Game
{
   public int IdGame { get; set; }
   public string Name { get; set;  }
   public string Site { get; set;  }
   public string Category { get; set; }
   public double Price { get; set;  }
}
```

5.1.3 В конструкторе окна создать список игр games с пятью объектами.

5.1.4 Добавить на форму ListView и указать для него в качестве источника данных список games.

5.1.5 Реализовать отображение информации о пользователях в табличном виде, настроив View и указав DisplayMemberBinding у каждого столбца GridViewColumn

5.2 Отображение данных в DataGrid

5.2.1 Создать WPF-приложение.

5.2.2 Добавить в приложение класс Game

5.2.3 В конструкторе окна создать список игр games с пятью объектами.

5.2.4 Добавить на форму DataGrid и указать для него в качестве источника данных список games.

5.2.5 Запретить вставку и удаление строк в DataGrid.

5.2.6 Указать различные цвета для четных и нечетных строк DataGrid

5.3 Настройка привязки столбцов DataGrid

5.3.1 Создать WPF-приложение.

5.3.2 Добавить в приложение класс Game

5.3.3 В конструкторе окна создать список игр games с пятью объектами.

5.3.4 Отключить автогенерацию стобцов в DataGrid

5.3.5 Реализовать отображение следующих привязанных данных в столбцах DataGrid:

- название (DataGridTextColumn)
- цена (DataGridTextColumn) с указанием денежного формата отображения
- сайт (DataGridHyperlinkColumn)
- кнопка «Подробнее» (DataGridTemplateColumn с типом элемент Button, у которой указать подпись кнопки)
- кнопка «Редактировать» (DataGridTemplateColumn с типом элемент Button, у которой указать подпись кнопки)
- кнопка «Удалить» (DataGridTemplateColumn с типом элемент Button, у которой указать подпись кнопки)

5.4 Обработка нажатий кнопки в DataGrid

5.4.1 Добавить кнопке «Подробнее» обработчик нажатия

5.4.2 В обработчике нажатия на кнопку реализовать приведение отправителя (sender) к типу кнопки, свойства кнопки DataContext к типу Game и вывод всех данных о выбранном объекте в MessageBox.

5.5 Привязка выбранного элемента к выпадающему списку

5.5.1 В конструкторе окна создать список категорий игр с тремя строковыми значениями.

5.5.2 Добавить в DataGrid столбец типа DataGridComboBoxColumn, который связать с категорией игры и заполнить значениями из коллекции строк с категориями игр.

5.5.3 Настроить выпадающий список, чтобы при смене значения в выпадающем списке изменялась категория игры.

6Порядок выполнения работы

6.1 Выполнить все задания из п.5 в одном решении LabWork33. 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 DataGrid и для чего он используется?
8.2 Какие типы столбцов поддерживаются в DataGrid?
8.3 Как добавить кнопку в строки DataGrid?
8.4 Как указать источник данных для DataGrid?
8.5 Как указать источник данных для выпадающего списка DataGrid?