

Лабораторная работа №33

Разработка приложения для отображения данных в табличном виде

1 Цель работы

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

2 Литература

2.1 <https://metanit.com/sharp/wpf/5.9.php> - ListView

2.2 <https://metanit.com/sharp/wpf/5.14.php> - DataGrid

3 Подготовка к работе

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

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Отображение данных в ListView

5.1.1 Создать WPF-приложение.

5.1.2 В конструкторе окна получить список файлов.

5.1.3 Добавить на форму ListView и указать для него в качестве источника данных список полученных файлов.

5.1.4 Реализовать отображение информации о файлах в табличном виде, настроив View и указав DisplayMemberBinding у каждого столбца GridViewColumn. Требуется указать столбцы со следующими заголовками:

- Имя файла
- Дата изменения
- Расширение
- Размер

5.2 Отображение данных в DataGrid

5.2.1 Создать WPF-приложение.

5.2.2 Добавить в приложение класс User со свойствами: адрес, логин, пароль, категория (все данные строковые, категория: email / БД / сайт), архивный (логическое значение).

5.2.3 В конструкторе окна создать список пользователей с пятью объектами.

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

5.2.5 Настроить свойства DataGrid:

- запретить вставку и удаление строк
- указать цвета для четных строк
- отключить отображение вертикальных линий

5.3 Настройка шаблонов столбцов

5.3.1 В приложении п.5.1 столбец реализовать отображение:

- единицы размера файлов, настроив у столбца формат StringFormat={ } {0:0 Б}
- кнопки «Подробнее», изменив шаблон столбца GridViewColumn.CellTemplate.

5.3.2 В приложении п.5.2 отключить автогенерацию столбцов в DataGrid и реализовать отображение следующих привязанных данных в столбцах DataGrid:

- адрес (DataGridHyperlinkColumn)
- логин (DataGridTextColumn)
- пароль (DataGridTextColumn)
- архивный (DataGridCheckBoxColumn)
- кнопка «Скопировать пароль» (DataGridTemplateColumn с типом элемента Button, у которой указать подпись кнопки)

5.3.3 Для того, чтобы скрыть пароль и запретить его редактирование в ячейке:

- добавить в класс User свойство на чтение скрытыйПароль:
public string HiddenPassword => new('.', длина пароля);
- реализовать отображение вместо пароля свойства скрытыйПароль и сделать столбец с паролем только на чтение.

5.4 Обработка нажатий кнопки

5.4.1 Добавить кнопке «Подробнее» обработчик нажатия, при нажатии на кнопку вывести полное имя файла.

5.4.2 Добавить кнопке «Скопировать пароль» обработчик нажатия, при нажатии на кнопку скопировать пароль пользователя:

Clipboard.SetText(пароль);

5.5 Привязка выбранного элемента к выпадающему списку

5.5.1 В конструкторе окна из п.5.2 создать список категорий с тремя строковыми значениями (email / БД / сайт).

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

5.5.3 Настроить выпадающий список, чтобы при смене значения в выпадающем списке изменялась категория учетных данных пользователя.

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?

9 Приложение

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 отображается при редактировании)

DataGridTemplateColumn – позволяет задать шаблон для отображения столбца (например, кнопку с обработчиком события, изображение, контейнер)

Создание шаблона столбца **DataGridTemplateColumn**:

```
<DataGridTemplateColumn>
  <DataGridTemplateColumn.CellTemplate>
    <DataTemplate>Шаблон</DataTemplate>
  </DataGridTemplateColumn.CellTemplate>
</DataGridTemplateColumn>
```

Привязка данных к **DataGrid**

контроль. `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`

Свойства **DataGrid**:

AutoGenerateColumns – автогенерация столбцов на основе источника данных

IsReadOnly – запрет редактирования данных

CanUserAddRows – разрешение пользователю добавлять строки

CanUserDeleteRows – разрешение пользователю удалять 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).

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

HeadersVisibility – видимость заголовков строк и столбцов

FrozenColumnCount – количество зафиксированных для прокрутки столбцов