

Упражнение №5 по ПС WPF

Работа с XAML, WPF, Bindings.

Целта на това упражнение

Да се запознаем с WPF и неговите опции за потребителски интерфейс

Задачите в упражнението изграждат:

Малка студентска информационна система

В това упражнение:

Десктоп приложение, което работи с база данни.

В края на упражнението:

Ще създадем интерфейс който да визуализира списъка с потребителите създадени в предишните упражнения.

За домашно:

Да се промени кода така, че:

1. Да се добави опция за скриване на паролата от списъка, чрез употребата на **Converter**. Може да потърсите **Binding Converter**.
2. Да се добави нов прозорец и контрола която да визуализират данните запаметени от Logger-a, по аналог на екраните по-горе.
3. Да се промени контролата за Logger-a, така че в списъка да се вижда само датата на събитието а при двоен клик, с MessageBox, да се визуализира и събитието, форматирано.

Важни знания от упражнението: WPF, Bindings, XAML

Зареждане на проект

1. Отворете **Visual Studio**
2. Заредете **Solution**-а създаден в предишните упражнения.
3. Създаваме нов проект, който кръщаваме **UI**. Типа на проекта трябва да бъде **WPF**, задължително **C#**
4. Добавяме референция към проекта **DataLayer**
 - a. С десен бутон кликаме върху **Dependencies** на проекта **UI** и избираме **Add Project Reference ...**
 - b. От отвореният прозорец избираме **DataLayer** Проекта.

Създаване на папки

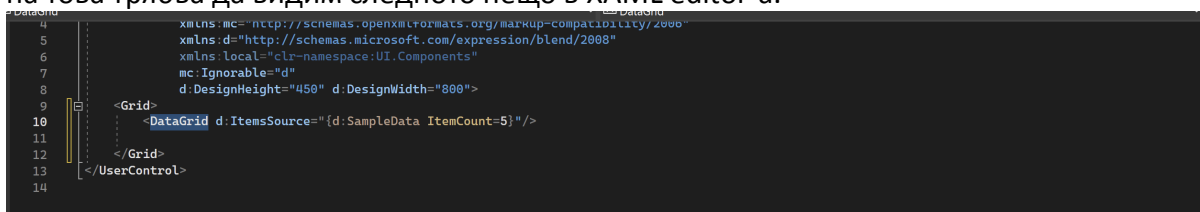
1. Създаваме следните папки: **Windows, Extras, Components**

Подреждаме съществуващият код

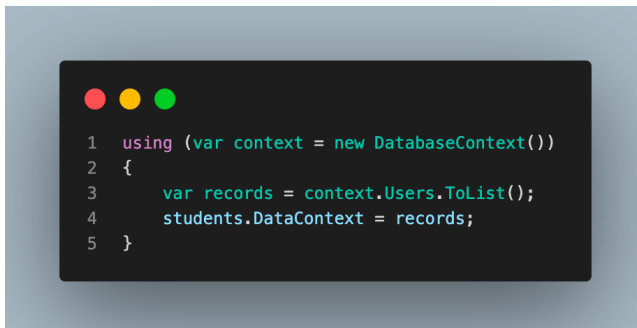
1. Преместваме файла **MainWindow.xaml** в папката **Windows**, където ще стоят всички прозорци които ще имаме в приложението
2. Отваряме файла **App.xaml**, което е нашата входна точка и на мястото на което се намира нашият **StartupUri** трябва да добавим **"Windows/"** пред **MainWindow.xaml**, за да окажем че нашият основен екран се намира в папката **Windows**.

Създаване на Component **StudentsList.xaml**

1. Кликаме с десен бутон на папката **Components** и избираме **Add**.
2. От отвореното меню избираме **UserControl (WPF)**
3. В новоотвореният прозорец кръщаваме файла **StudentsList**
4. От прозореца **Toolbox**, избираме **DataGrid** и го влачим върху екрана. В резултат на това трябва да видим следното нещо в **XAML editor**-а.



2. В конструктора на нашият компонент виждаме InitializeComponent, което е автоматичен метод който се извиква за да свърже XAML потребителският интерфейс с кода зад страницата. След този InitializeComponent, добавяме следният код по аналог от миналото упражнение:



Последният ред от този using е students.DataContext = records, students идва от името на компонента чиито Context трябва да приеме данните от базата данни.

Връзване на нашият нов компонент към основният UI

1. Отворете файла MainWindow.xaml
2. В най-горната част където се намира Title="MainWindow" променете заглавието на страницата на "List of all students"
3. Добавете нов атрибут WindowStartupLocation и му задайте стойност „CenterScreen”
Това ще зададе на екрана да се покаже в центъра на екрана с неговото отваряне
4. За да добавим компонента който създадохме и той да бъде достъпен тук, трябва да добавим неговият namespace като в същият таг Window, трябва да добавим следният атрибут **xmlns:local2="clr-namespace:UI.Components"**
По този начин създаваме локален namespace local2, който сочи към UI.Components namespace-a
5. В <Grid> тага, вече имаме достъп да добавим таг за листа с нашите потребители, него добавяме по следният начин:

`<local2:StudentsList x:Name="studentsList" Padding="2" />`

6. Стартираме проекта и би трябвало да видим данните за всички потребители които имаме в базата данни.

Настройка на визуализацията на нашата таблица с потребители

1. Отворете файла StudentsList.xaml
2. Тъй като DataGrid-a е таг, който е самозатворен, трябва да го променим да има отварящ и затварящ таг. След това трябва да променим AutoGenerateColumns да бъде False

`<DataGrid .../>`, трябва да стане `<DataGrid ...></DataGrid>`

3. В този отворен таг, трябва да добавим описание на колоните които искаме да имаме:

```
1 <DataGrid.Columns>
2   <DataGridTextColumn Header="#" IsReadOnly="True" Binding="{Binding Id}" />
3   <DataGridTextColumn Header="Names" IsReadOnly="True" Binding="{Binding Name}" />
4   <DataGridTextColumn Header="Role" IsReadOnly="True" Binding="{Binding Role}" />
5   <DataGridTextColumn Header="Expires In" Binding="{Binding Expires}"/>
6 </DataGrid.Columns>
```

Както виждате, *DataGrid.Columns* е мястото на което дефинираме всички колони които искаме да дефинираме, за целта на упражнението за всяка от колоните сме се спрели на текстови колони *DataGridTextColumn*, на които сме задали с *Header* името на колоната, след което *IsReadOnly* за да нямаме опция за корекция на данните и *Binding* който сочи към определеното свойство което искаме да визуализираме. Тъй като ние вече сме заредили колекцията с потребителите на *DataGrid*-а, тук в *Binding*-а трябва да посочим единствено свойството на обекта което искаме да визуализираме.