МДК 01.03 «Разработка мобильных приложений»

Практическая работа №4

Учебная группа **32\1 ИСД-21К**

Студент **Юлдашев Иброхим Камалидинович**

# Код программного приложения

MainWindow

|  |
| --- |
| <Window x:Class="Yuldashev.MainWindow"  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  xmlns:local="clr-namespace:Yuldashev"  xmlns:vm="clr-namespace:Yuldashev.ViewModels"  mc:Ignorable="d"  Title="MainWindow" Height="450" Width="800">  <Window.DataContext>  <vm:MainViewModel/>  <!--Запустить один раз-->  </Window.DataContext>  <StackPanel Orientation="Horizontal">  <ListView ItemsSource="{Binding directors}">  <ListView.View>  <GridView>  <GridViewColumn Header="directorid" DisplayMemberBinding="{Binding directorid}"></GridViewColumn>  <GridViewColumn Header="ФИО" DisplayMemberBinding="{Binding fullname}"></GridViewColumn>  <GridViewColumn Header="Количество фильмов" DisplayMemberBinding="{Binding NamberOfFilms }"></GridViewColumn>  <GridViewColumn Header="Возраст" DisplayMemberBinding="{Binding age}"></GridViewColumn>  </GridView>  </ListView.View>  </ListView>  <StackPanel>  <ListView x:Name="listView" ItemsSource="{Binding movies}">  <ListView.View>  <GridView>  <GridViewColumn Header="moveid" DisplayMemberBinding="{Binding movieid}"/>  <GridViewColumn Header="directorid" DisplayMemberBinding="{Binding directorid}"/>  <GridViewColumn Header="Название" DisplayMemberBinding="{Binding title}"/>  <GridViewColumn Header="Дата выхода" DisplayMemberBinding="{Binding release}"/>  <GridViewColumn Header="Страна" DisplayMemberBinding="{Binding country}"/>  <GridViewColumn>  <GridViewColumn.CellTemplate>  <DataTemplate>  <Button Width="50" Height="20" Command="{Binding ElementName=listView, Path=DataContext.Deletemovie}" CommandParameter="{Binding movieid}" Content="Удалить" FontSize="12" FontWeight="Bold" BorderBrush="#FFFF7979" Background="#FFFF7D7D" Foreground="White"></Button>  </DataTemplate>  </GridViewColumn.CellTemplate>  </GridViewColumn>  </GridView>  </ListView.View>  </ListView>  <GroupBox>  <StackPanel>  <StackPanel>  <Label Content="Директор"/>  <TextBox Text="{Binding newmovies.directorid}"/>  </StackPanel>  <StackPanel>  <Label Content="Название фильма"/>  <TextBox Text="{Binding newmovies.title}"/>  </StackPanel>  <StackPanel>  <Label Content="Дата выхода"/>  <TextBox Text="{Binding newmovies.release}"/>  </StackPanel>  <StackPanel>  <Label Content="Страна"/>  <TextBox Text="{Binding newmovies.country}"/>  </StackPanel>  <Button Command="{Binding Addnewmovie}" Content="Добавить фильм" Background="#FFA0E274" BorderBrush="#FFB6FF7C" Foreground="White" FontWeight="Bold"/>  </StackPanel>  </GroupBox>  </StackPanel>  </StackPanel>  </Window> |

Contexts

|  |
| --- |
| using Microsoft.EntityFrameworkCore;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using Yuldashev.Models;  namespace Yuldashev.Contexts  {  public class MovieContext : DbContext  {  protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)  {  optionsBuilder.UseSqlServer(@"Server=DESKTOP-PK64UE3\SQLEXPRESS;Database=Yuldashev;Trusted\_Connection=True;TrustServerCertificate=True;");  }  public DbSet<Movie> Movies { get; set; }  public DbSet<Director> Directors { get; set; }  }  } |

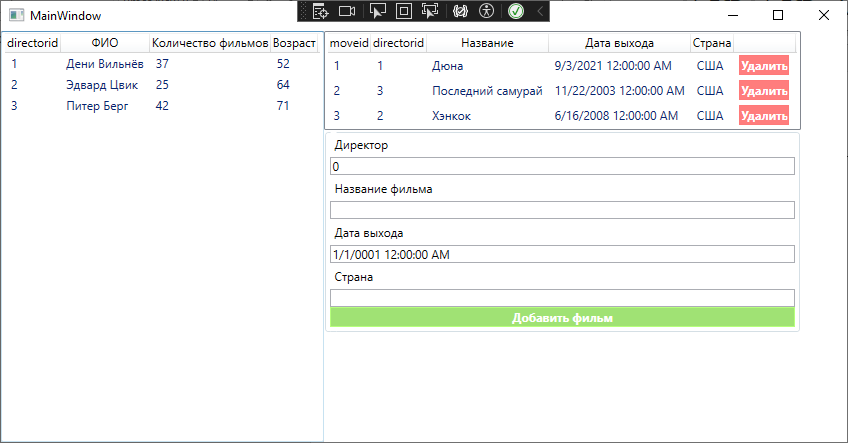
Models

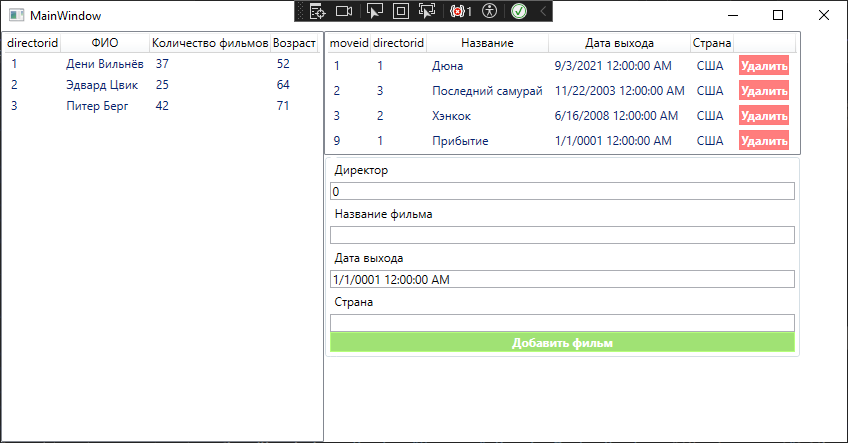
|  |
| --- |
| using System;  using System.Collections.Generic;  namespace Yuldashev.Models  {  public class Director  {  public int directorid { get; set; }  public string fullname { get; set; }  public int NamberOfFilms { get; set; }  public int age { get; set; }  }  }  namespace Yuldashev.Models  {  public class Movie  {  public int movieid { get; set; }  public int directorid { get; set; }  public string title { get; set; }  public DateTime release { get; set; }  public string country { get; set; }  }  } |

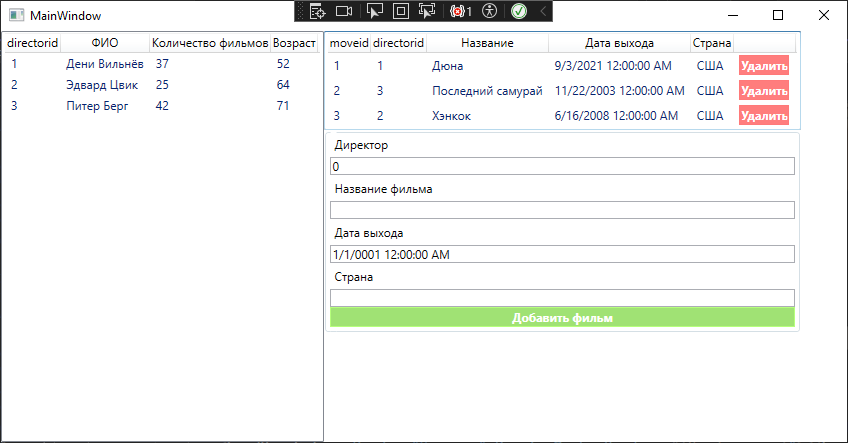
ViewModels

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Collections.Specialized;  using System.ComponentModel;  using System.Linq;  using System.Runtime.CompilerServices;  using System.Text;  using System.Threading.Tasks;  using Yuldashev.Contexts;  using Yuldashev.Models;  namespace Yuldashev.ViewModels  {  public class MainViewModel : INotifyPropertyChanged  {  public event PropertyChangedEventHandler? PropertyChanged;  public void OnPropertyChanged([CallerMemberName] string propertyName = "")  {  PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));  }  private IEnumerable<Director> \_directors;  public IEnumerable<Director> directors  {  get  {  return \_directors;  }  set  {  \_directors = value;  OnPropertyChanged();  }  }  private IEnumerable<Movie> \_movies;  public IEnumerable<Movie> movies  {  get  {  return \_movies;  }  set  {  \_movies = value;  OnPropertyChanged();  }  }  private void GetDirector()  {  using (var context = new Context())  {  directors = context.directors.ToList();  }  }  private void GetMovie()  {  using (var context = new Context())  {  movies = context.movies.ToList();  }  }  private Movie \_newmovies;  public Movie newmovies  {  get  {  return \_newmovies;  }  set  {  \_newmovies = value;  OnPropertyChanged();  }  }  public DelegateCommand Addnewmovie  {  get  {  return new DelegateCommand(o =>  {  AddNewMovie();  });    }  }  private void AddNewMovie()  {  using (var context = new Context())  {  context.movies.Add(newmovies);  context.SaveChanges();  movies = context.movies.ToList();  }  newmovies = new Movie();  }  public DelegateCommand Deletemovie  {  get  {  return new DelegateCommand(o =>  {  DeleteMovie((int)o);  });  }  }  private void DeleteMovie(int movieId)  {  using(var context = new Context())  {  Movie movieDelete = context.movies.Find(movieId);  context.movies.Remove(movieDelete);  context.SaveChanges();  movies = context.movies.ToList();  }  }  public MainViewModel()  {  newmovies = new Movie();  GetDirector();  GetMovie();  }  }  } |

Демонстрация работы приложения



Добавление новых данных

Удаление данных