XVII ЕЖЕГОДНЫЙ ВСЕРОССИЙСКИЙ КОНКУРС ДОСТИЖЕНИЙ ТАЛАНТЛИВОЙ МОЛОДЁЖИ

«НАЦИОНАЛЬНОЕ ДОСТОЯНИЕ РОССИИ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Секция:**

**Тема: «Реализация принципов объектно-ориентированного программирования с использованием технологии Entity Framework на базе IDE Visual Studio 2022»**

**Автор: Немтырёва Ксения Антоновна, студентка Ликино-Дулевского политехнического колледжа – филиала ГГТУ (ЛДПК – филиал ГГТУ)**

**Научный руководитель: Кузьмина Елена Евгеньевна**

**преподаватель общеобразовательных дисциплин Ликино-Дулевского политехнического колледжа – филиала ГГТУ (ЛДПК – филиал ГГТУ)**

**г. Ликино-Дулево**

**2023 г.**

# Аннотация

Аннотация (что сделано, что нового получено)

Оглавление

[Аннотация 2](#_Toc126918045)

[1. Постановка задачи 4](#_Toc126918046)

[2. Разработка структуры базы данных в среде SQL SERVER 4](#_Toc126918047)

[3. Формирование данных для импорта и выполнение импорта 4](#_Toc126918048)

[4. Создание настольного приложения 12](#_Toc126918049)

[5. Проектирование интерфейса 17](#_Toc126918050)

[6. Разработка интерфейса 18](#_Toc126918051)

[7. Реализация функций для работы со справочниками 20](#_Toc126918052)

[8. Реализация функций для работы с основной информацией 21](#_Toc126918053)

[9. Реализация функции вывода на печать 22](#_Toc126918054)

[10. Отладка 24](#_Toc126918055)

[Список литературы 25](#_Toc126918056)

[Приложения 26](#_Toc126918057)

1. **Постановка задачи**

Необходимо написать программу, которая будет хранить информацию о расписании поездов (например, такую как: название пункта назначения и отправления, номер поезда, время отправления, время в пути, станции на которых делается остановка (вложенный список)).

1. **Разработка структуры базы данных в среде SQL SERVER**

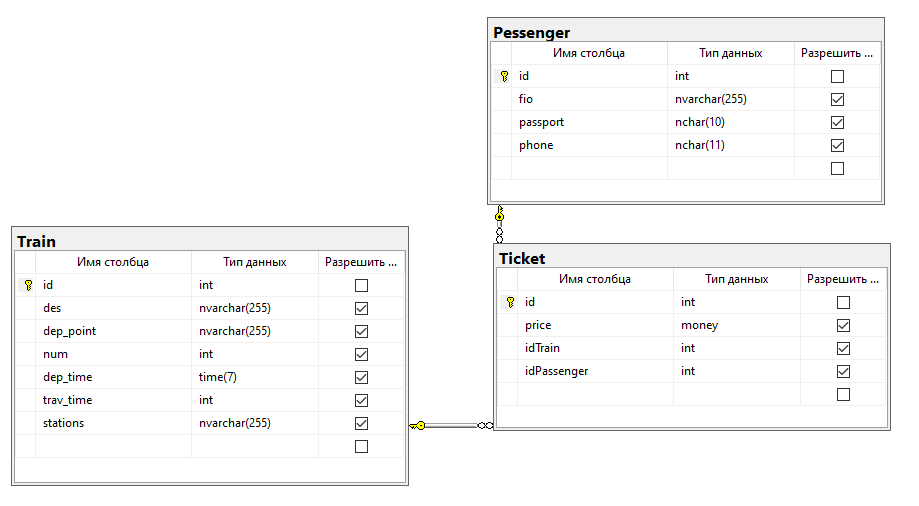
****

Рис.1 «Схема данных предметной области в среде SQL SERVER»

1. **Формирование данных для импорта и выполнение импорта**

Таблица «Train»:

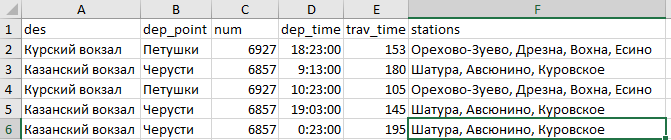


Рис.2 «Данные для импорта в Excel таблицы поездов»

Таблица «Pessenger»:

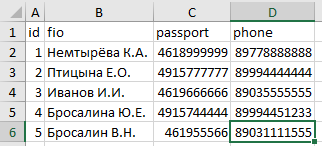


Рис.3 «Данные для импорта в Excel таблицы пассажиров»

Таблица «Ticket»:

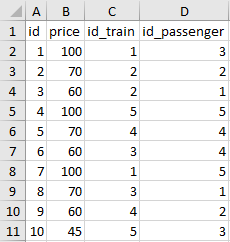


Рис.4 «Данные для импорта в Excel таблицы билеты»

Выполнение импорта данных в SQL SERVER:

* 1. Выбираем нашу БД 🡪 нажимаем ПКМ (правая кнопка мыши) 🡪 Выбираем «задачи» 🡪 Выбираем «Импорт данных»

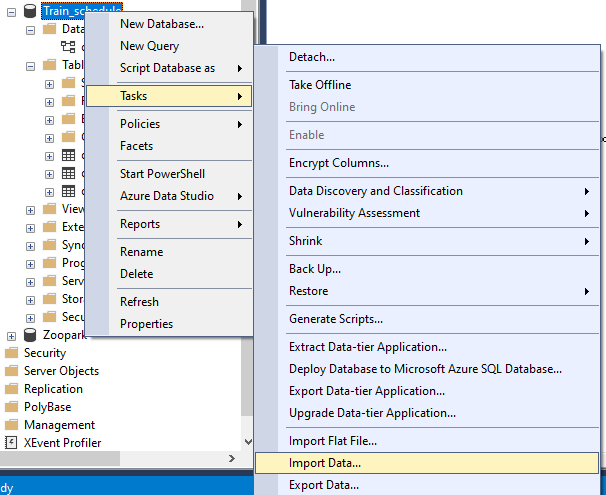


Рис.5 «Выполнение импорта данных»

* 1. Выполняем инструкции.

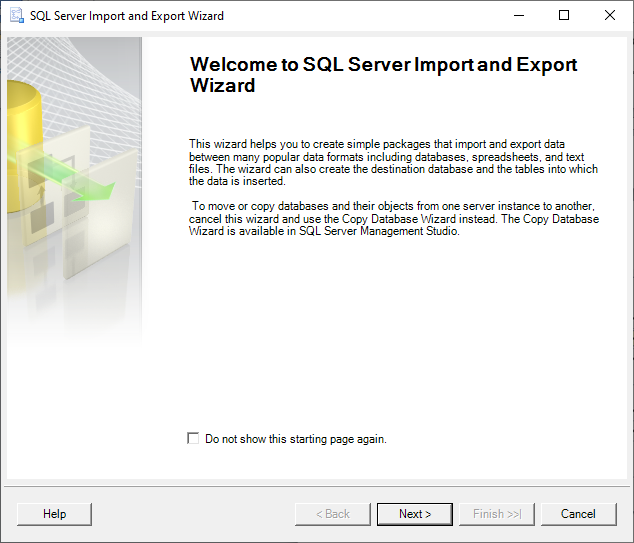


Рис.6 «Выполнение импорта данных»

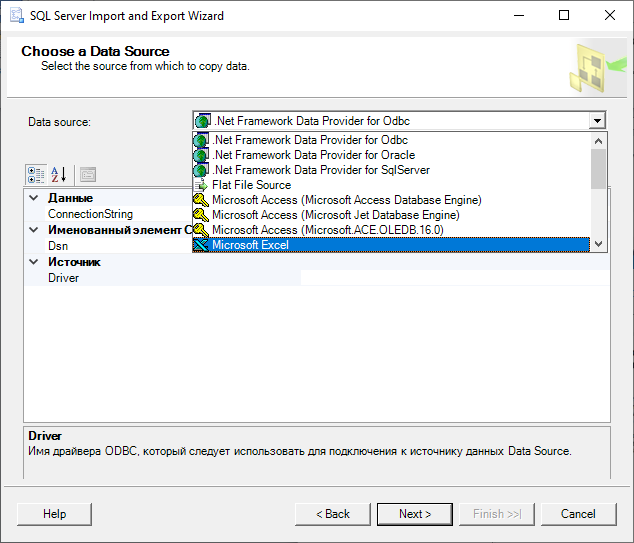


Рис.7 «Выполнение импорта данных»

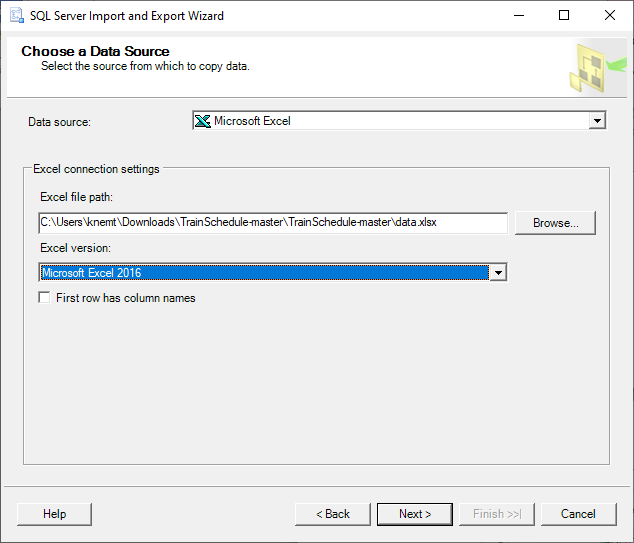
****

Рис.8 «Выполнение импорта данных»

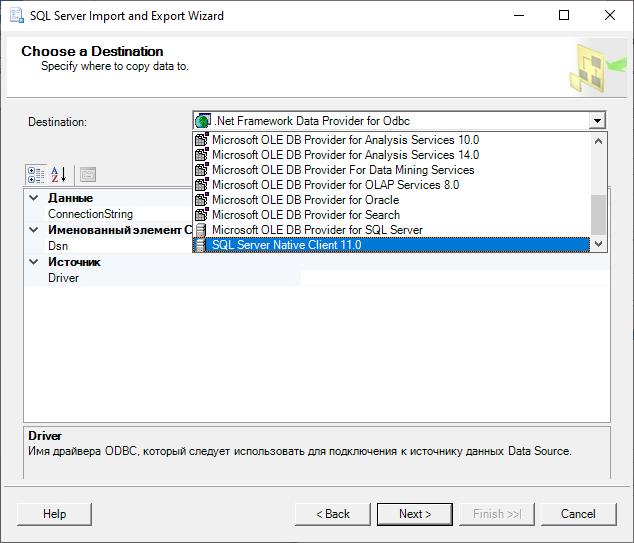
****

Рис.9 «Выполнение импорта данных»

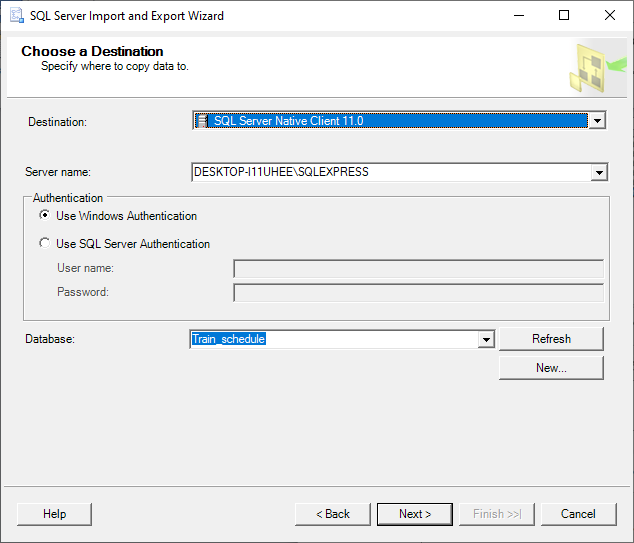
****

Рис.10 «Выполнение импорта данных»

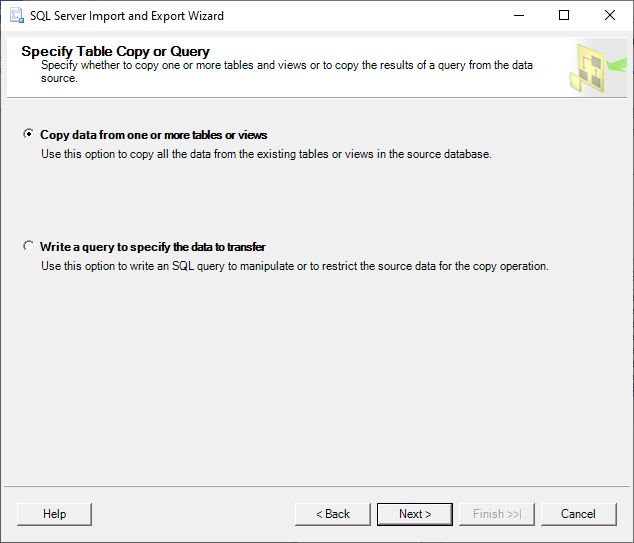
****

Рис.11 «Выполнение импорта данных»

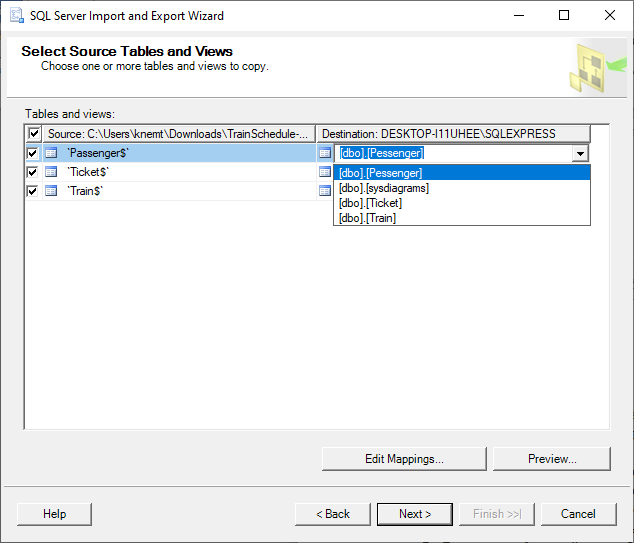
****

Рис.12 «Выполнение импорта данных»

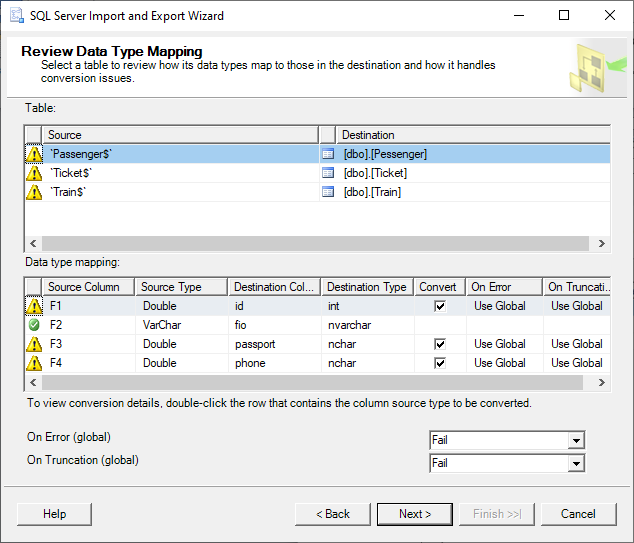
****

Рис.13 «Выполнение импорта данных»

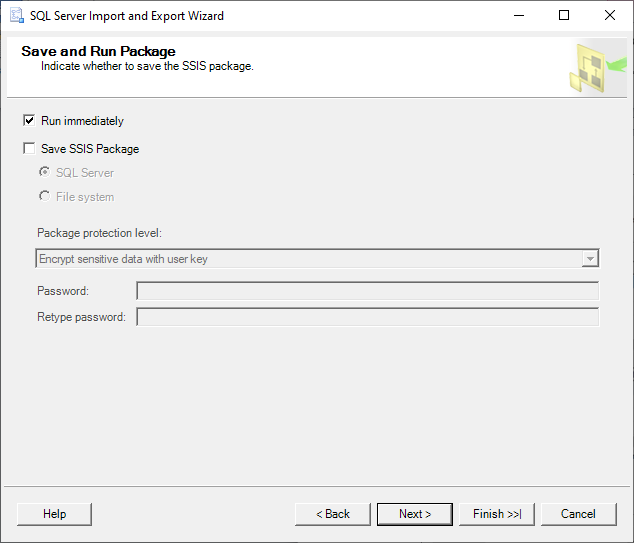
****

Рис.14 «Выполнение импорта данных»

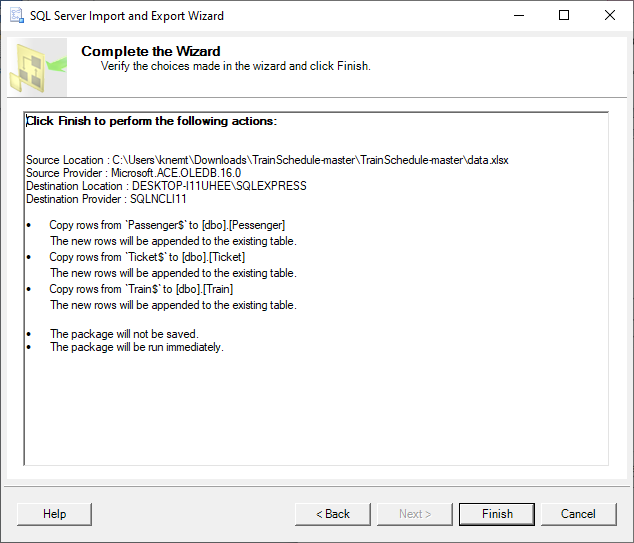
****

Рис.15 «Выполнение импорта данных»

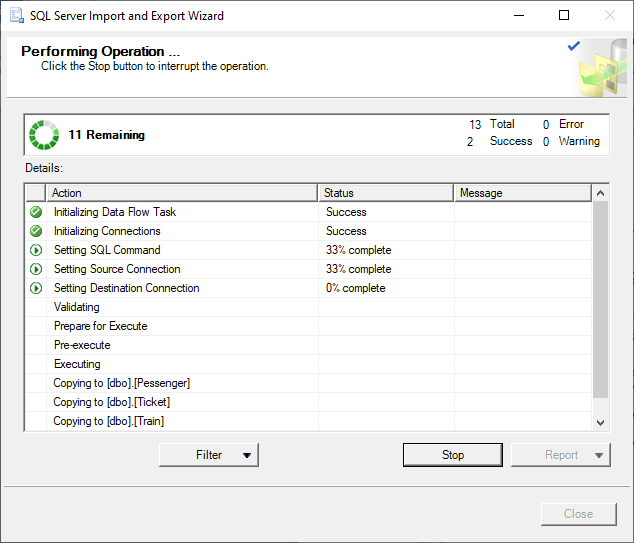
****

Рис.16 «Выполнение импорта данных»

* 1. Результат в SQL SERVER

Таблица «Train»:

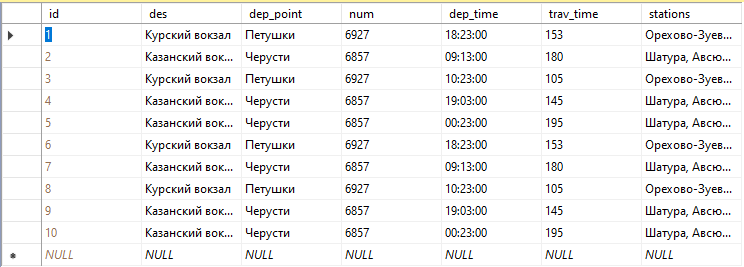


Рис.17 «Импортированные данные в таблице поездов»

Таблица «Pessenger»:

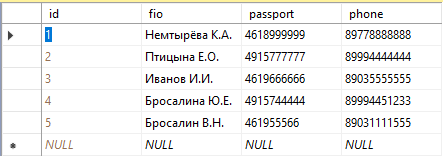


Рис.18 «Импортированные данные в таблице пассажиров»

Таблица «Ticket»:

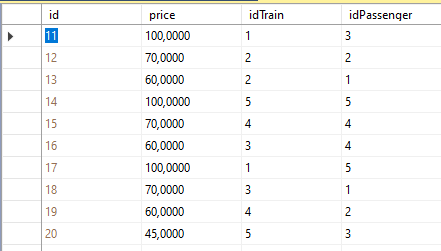
****

Рис.19 «Импортированные данные в таблице билеты»

1. **Создание настольного приложения**
   1. Заходим в Visual Studio. Выбираем «Создание проекта»

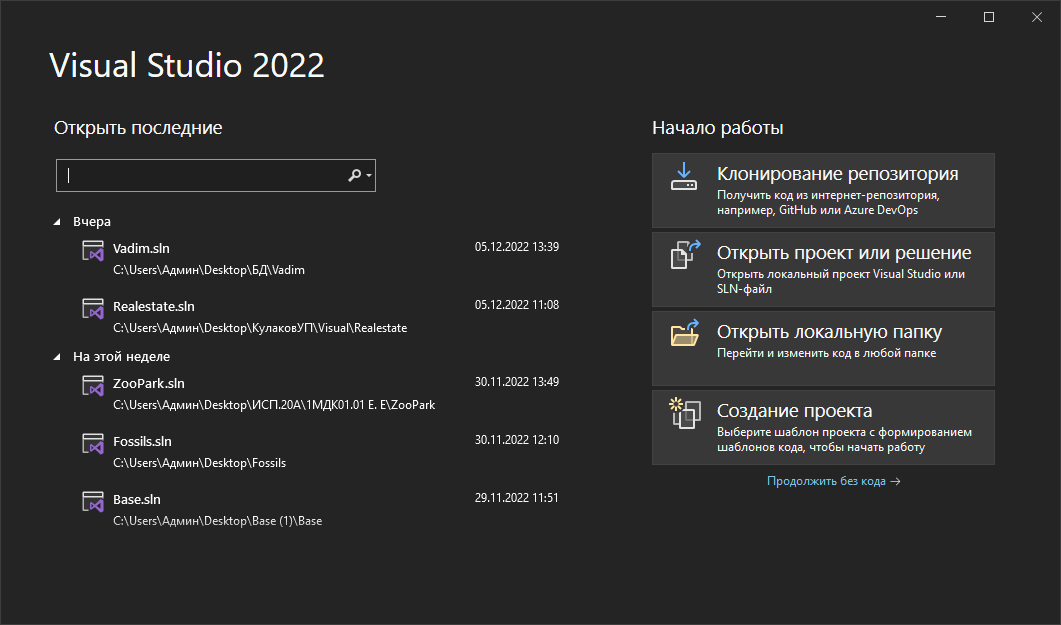


Рис.20 «Импортированные данные в таблице билеты»

* 1. Выбираем C# 🡪 Windows 🡪 Рабочий стол. В списке появившихся проектов выбираем «Приложение WPF (.NET Framework)»

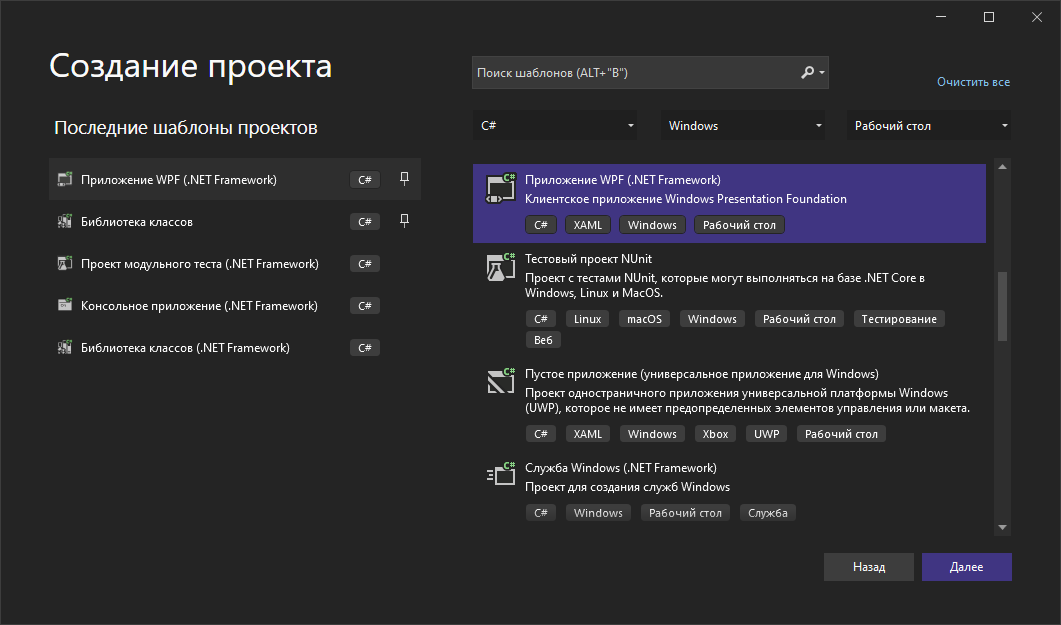


Рис.21 «Импортированные данные в таблице билеты»

* 1. Даём имя проекту 🡪 Выбираем его расположение 🡪 Помещаем решение и проект в один каталог (рекомендуемо) 🡪 Нажимаем «Создать»

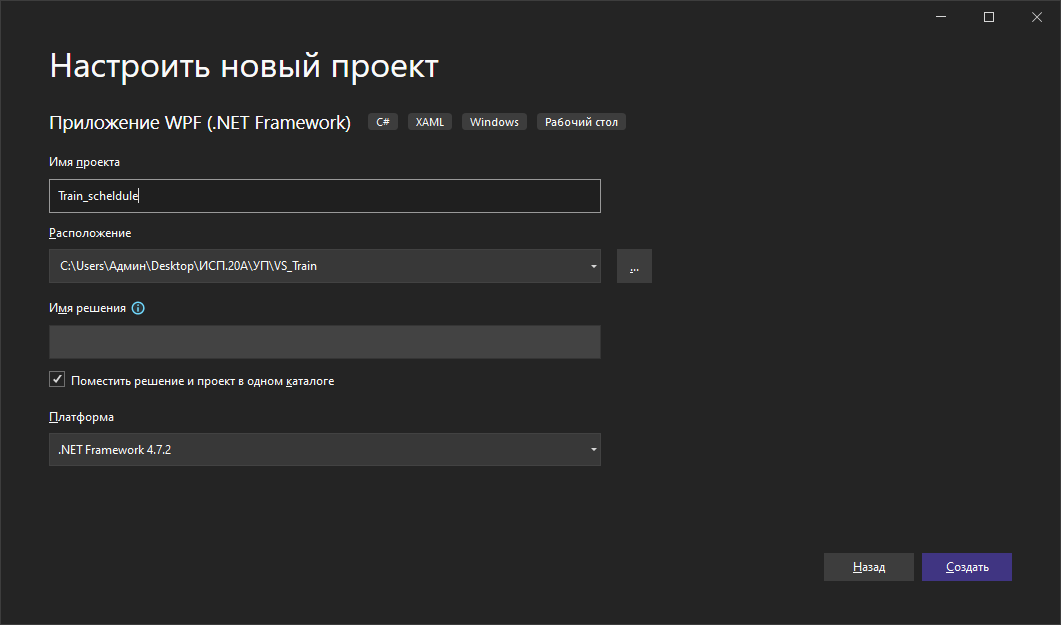


Рис.22 «Импортированные данные в таблице билеты»

* 1. В создавшемся проекте создадим его структуру (папки: для классов, для страниц, для ресурсов)

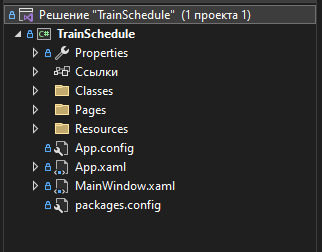


Рис.23 «Структура проекта»

Подключение базы данных SQL SERVER:

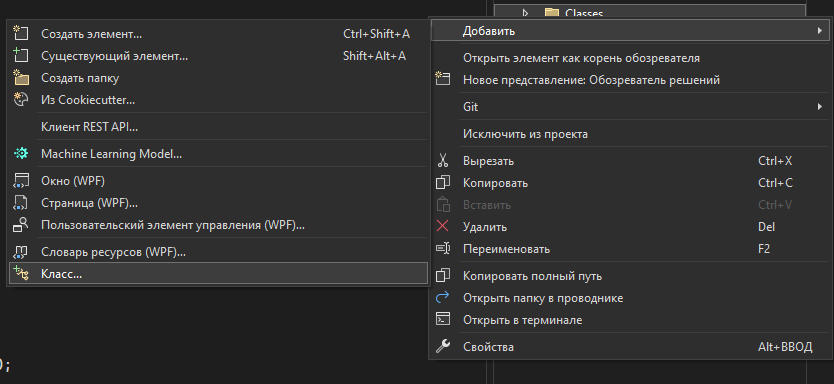


Рис.24 «Добавим класс»

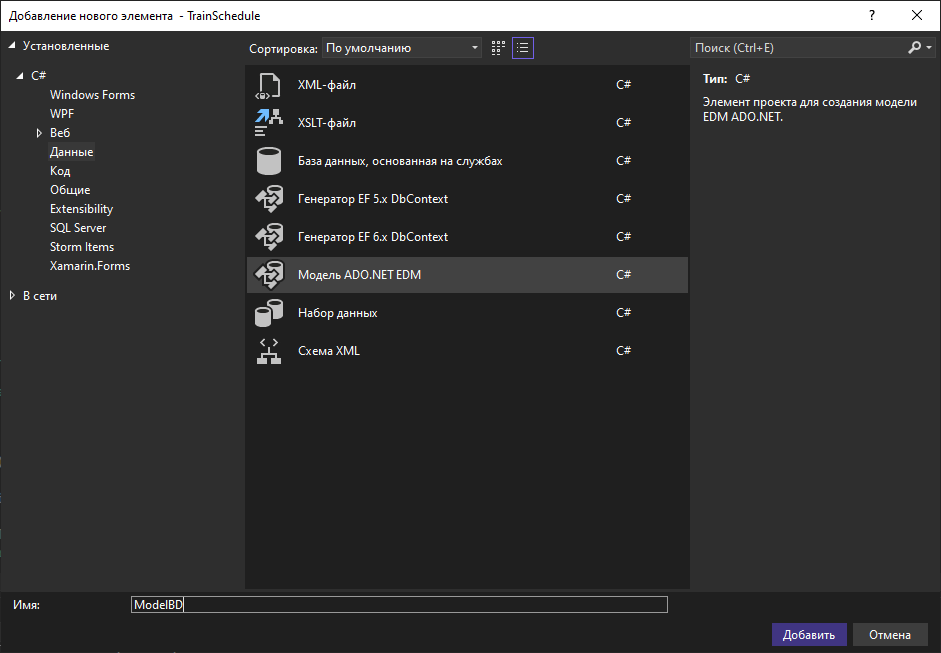


Рис.25 «Создадим модель ADO.NET TDM, Дадим ей имя»

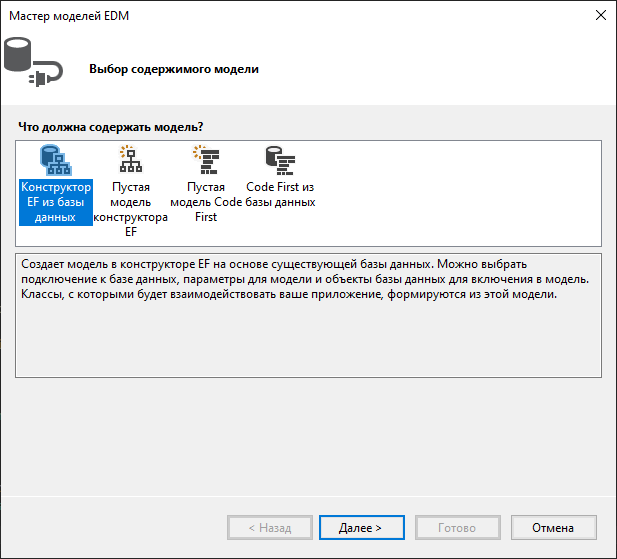


Рис.26 «Выбор содержимого модели»

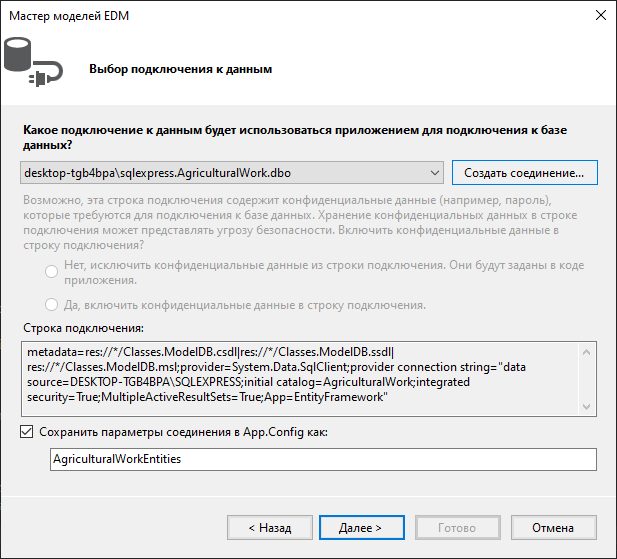


Рис.27 «Создадим подключение»

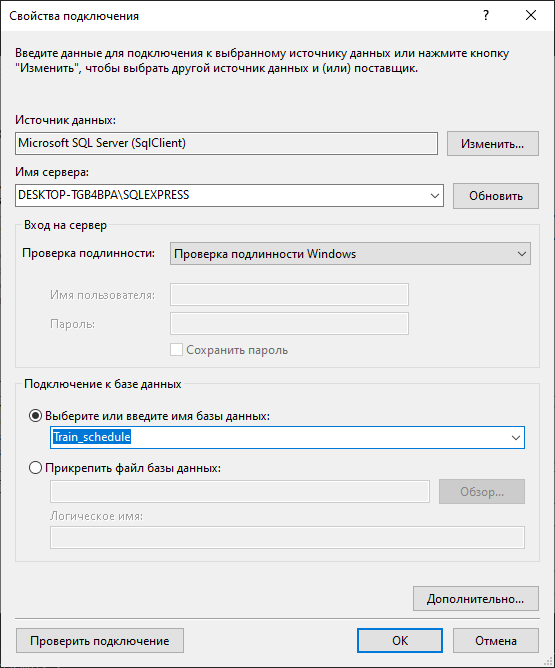


Рис.28 «Укажем имя сервера и базу, которую подключаем»

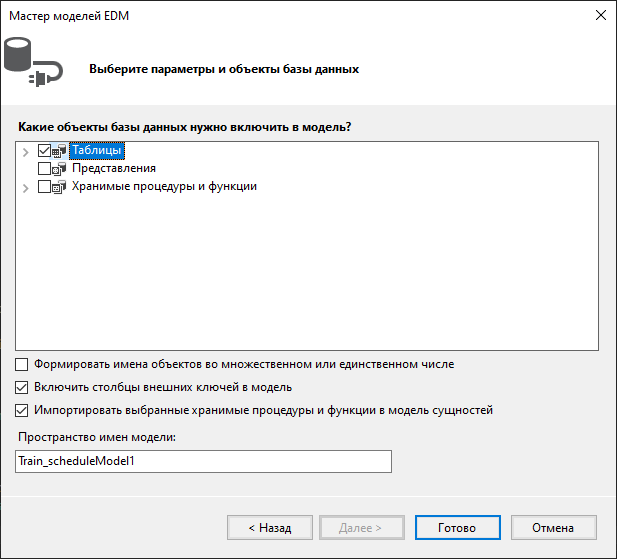


Рис.29 «Выбираем объекты, включаемые в модель. Нажимаем «Готово»»

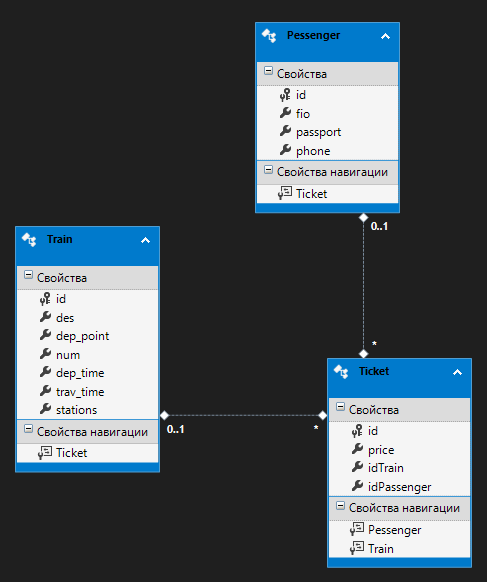


Рис.30 «Готовая модель в Visual Studio»

Для подключения пропишем код в TrainDB.Context.cs:

**private static Train\_scheduleEntities \_context;**

public Train\_scheduleEntities()

: base("name=Train\_scheduleEntities")

{

}

**public static Train\_scheduleEntities GetTrain()**

**{**

**if (\_context == null)**

**\_context = new Train\_scheduleEntities();**

**return \_context;**

**}**

1. **Проектирование интерфейса**

Спроектировать интерфейс приложения (XAML, Figma, draw.io и т.п.):

* 1. главная страница – название, логотип, переходы на остальные страницы;

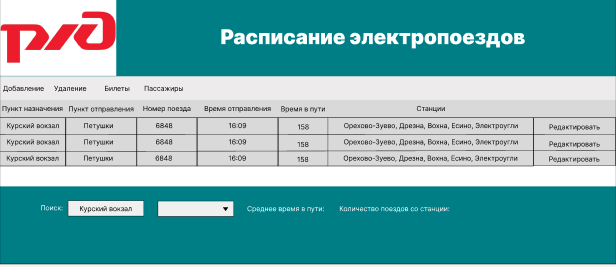


Рис.31 «Интерфейс главной страницы»

* 1. страница для работы со справочниками;

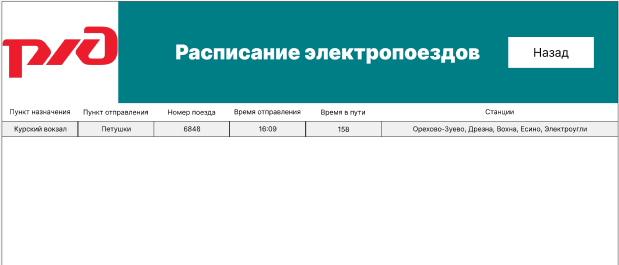


Рис.32 «Интерфейс страницы редактирования»

* 1. страница для работы с основной информацией;

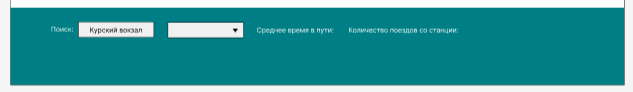


Рис.33 «панель для работы с основной информацией»

* 1. Дополнительные страницы (при необходимости)

Дополнительные страницы и переходы между страницами см. <https://www.figma.com/file/dCZyr04l7rmicXOYgbtxrk/Untitled?node-id=0%3A1&t=fBYPTK5UOA0GEU5G-0>

1. **Разработка интерфейса**

Интерфейс главной страницы:

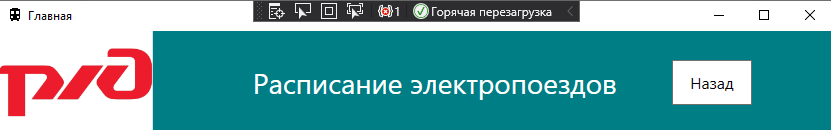


Рис.34 «Интерфейс главной страницы»

Программный код:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="2.5\*"/>

<RowDefinition Height="7.5\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0"

Orientation="Horizontal"

Background="#007E86"

HorizontalAlignment="Stretch">

<Image Source="Resources/Logo.png" Opacity="1" Visibility="Visible" />

<StackPanel Orientation="Horizontal" Width="600" >

<TextBlock Style="{StaticResource Header}"

VerticalAlignment="Center"

Text="Расписание электропоездов"

TextAlignment="Center"

Margin="100 10 50 10"/>

<Button Name="BtnBack" Content="Назад"

Click="BtnBack\_Click"/>

</StackPanel>

</StackPanel>

<Frame Grid.Row="1"

Name="frmMain"

NavigationUIVisibility="Hidden"

ContentRendered="frmMain\_ContentRendered">

</Frame>

</Grid>

Интерфейс для просмотра данных:

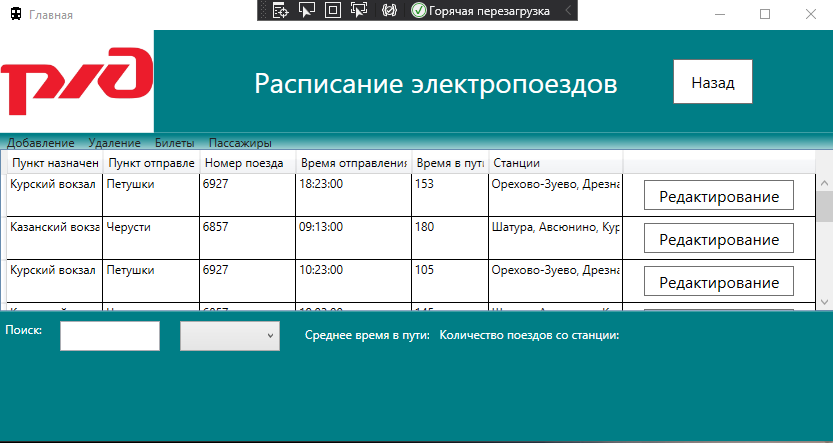
****

Рис.35 «Интерфейс для просмотра расписания поездов»

Программный код:

Приложение 1. «Программный код интерфейса расписание поездов»

Интерфейс для добавления/редактирования данных:

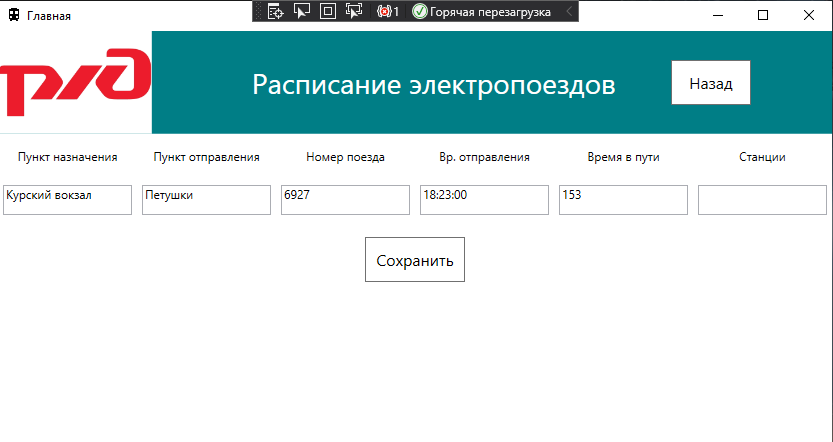
****

Рис.36 «Интерфейс для редактирования расписания поездов»

Программный код:

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

</Grid.RowDefinitions>

<TextBlock Text="Пункт назначения" Grid.Column="0" Grid.Row="0"

HorizontalAlignment="Center" VerticalAlignment="Center"/>

<TextBlock Text="Пункт отправления" Grid.Column="1" Grid.Row="0"

HorizontalAlignment="Center" VerticalAlignment="Center"/>

<TextBlock Text="Номер поезда" Grid.Column="2" Grid.Row="0"

HorizontalAlignment="Center" VerticalAlignment="Center"/>

<TextBlock Text="Вр. отправления" Grid.Column="3" Grid.Row="0"

HorizontalAlignment="Center" VerticalAlignment="Center"/>

<TextBlock Text="Время в пути" Grid.Column="4" Grid.Row="0"

HorizontalAlignment="Center" VerticalAlignment="Center"/>

<TextBlock Text="Станции" Grid.Column="5" Grid.Row="0"

HorizontalAlignment="Center" VerticalAlignment="Center"/>

<TextBox Text="{Binding des}" Grid.Column="0" Grid.Row="1"/>

<TextBox Text="{Binding dep\_point}" Grid.Column="1" Grid.Row="1"/>

<TextBox Text="{Binding num}" Grid.Column="2" Grid.Row="1"/>

<TextBox Text="{Binding dep\_time}" Grid.Column="3" Grid.Row="1"/>

<TextBox Text="{Binding trav\_time}" Grid.Column="4" Grid.Row="1"/>

<TextBox Text="{Binding stations}" Grid.Column="5" Grid.Row="1"/>

<StackPanel Grid.Row="2" Grid.ColumnSpan="6" Margin="10">

<Button Name="BtnSave" Content="Сохранить" Click="BtnSave\_Click"/>

</StackPanel>

</Grid>

Страницы для просмотра билетов и пассажиров, а также страницы для их редактирования и добавления реализованы аналогично.

1. **Реализация функций для работы со справочниками**

Кнопка добавления данных:



Рис.37 «При нажатие кнопки происходит переход на страницу добавления/редактирования с нулевыми значениями»

Программный код:

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageEdit(null));

}

Кнопка редактирования данных:



Рис.38 «При нажатие кнопки происходит переход на страницу добавления/редактирования с заполненными полями»

Программный код:

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageEdit((sender as Button).DataContext as Train));

}

Кнопка удаления данных:



Рис.39 «Все выделенные поля удалятся»

Программный код:

private void BtnDelete\_Click(object sender, RoutedEventArgs e)

{

var usersForRemoving = DgridTrain.SelectedItems.Cast<Train>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} пользователей?", "Внимание!",

MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

Train\_scheduleEntities.GetTrain().Train.RemoveRange(usersForRemoving);

Train\_scheduleEntities.GetTrain().SaveChanges();

MessageBox.Show("Данные удалены");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

DgridTrain.ItemsSource = Train\_scheduleEntities.GetTrain().Train.ToList();

}

1. **Реализация функций для работы с основной информацией**

Поиск по пункту назначения/номеру поезда/времени отправления



Рис.40 «Поиск»

private void TxtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

string search = TxtSearch.Text;

if (TxtSearch.Text != null)

{

DgridTrain.ItemsSource = Train\_scheduleEntities.GetTrain().Train.

Where(x => x.des.Contains(search)

|| x.dep\_time.ToString().Contains(search)

|| x.num.ToString().Contains(search)).ToList();

}

}

Поиск вывод итоговых значений по comboBox



Рис.41 «Вывод итоговых значений»

private void CmbAvr\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string txt = CmbAvr.SelectedValue.ToString();

txtAvr.Text = Train\_scheduleEntities.GetTrain().Train.Where(x => x.dep\_point == txt).Average(x => x.trav\_time).ToString();

txtCount.Text = Train\_scheduleEntities.GetTrain().Train.Where(x => x.dep\_point == txt).Count().ToString();

}

1. **Реализация функции вывода на печать**

Функция вывода на печать

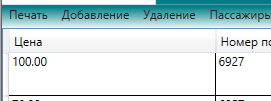
****

Рис.42 «Вывод на печать»

private void BtnExcel\_Click(object sender, RoutedEventArgs e)

{

Excel.Application excelApp = new Excel.Application();

Excel.Workbook wb = excelApp.Workbooks.Open($"{Directory.GetCurrentDirectory()}\\Отчёт по продажам.xlsx");

Excel.Worksheet ws = (Excel.Worksheet)wb.Worksheets[1];

ws.Cells[1, 6] = DateTime.Now.ToString();

int indexRows = 1;

//ячейка

ws.Cells[1][indexRows] = "Номер";

ws.Cells[2][indexRows] = "Номер поезда";

ws.Cells[3][indexRows] = "ФИО";

ws.Cells[4][indexRows] = "Цена";

//список пользователей из таблицы после фильтрации и поиска

var printItems = DgridTicket.Items;

//цикл по данным из списка для печати

foreach (Ticket item in printItems)

{

ws.Cells[1][indexRows + 1] = indexRows;

ws.Cells[2][indexRows + 1] = item.Train.num.ToString();

ws.Cells[3][indexRows + 1] = item.Pessenger.fio;

ws.Cells[4][indexRows + 1] = item.price;

indexRows++;

}

Excel.Range range = ws.Range[ws.Cells[2][indexRows + 1], ws.Cells[5][indexRows + 1]];

range.ColumnWidth = 15; //ширина столбцов

range.Font.Italic = true;

range.HorizontalAlignment = Excel.XlHAlign.xlHAlignLeft;//выравнивание по левому краю

ws.Cells[3, 6] = "Итого:";

ws.Cells[3, 7].Formula = $"=SUM(D:D)";

ws.Cells[indexRows + 2, 3] = "Подпись";

ws.Cells[indexRows + 2, 5] = "Немтырёва К.А.";

excelApp.Visible = true;

}

Отчёт по продажам:

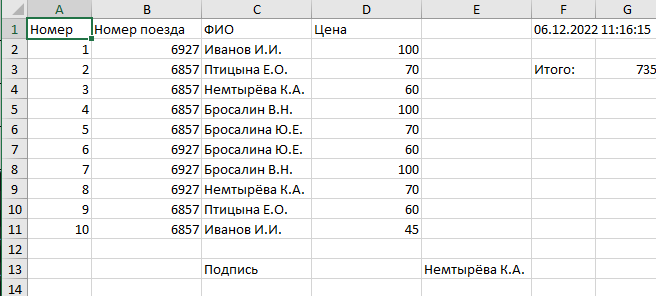


Рис.43 «Вывод в Excel»

1. **Отладка**

Ошибки, появившиеся в ходе работы

* 1. Отладка отображения стоимости

До:

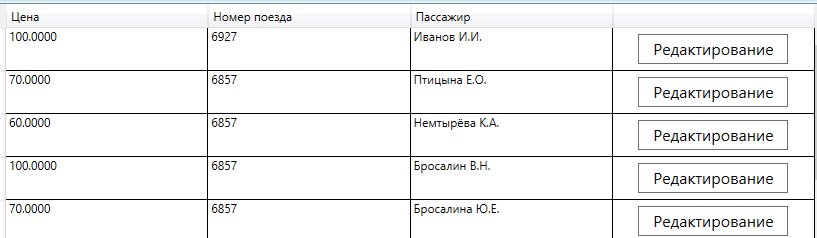


Рис.44 «Вывод данных до отладки»

<DataGridTextColumn Width="\*" Header="Цена"

Binding="{Binding price}"/>

После:

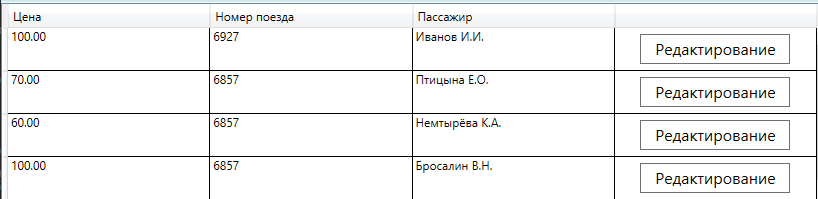


Рис.45 «Вывод после отладки»

<DataGridTextColumn Width="\*" Header="Цена"

Binding="{Binding **StringFormat={}{0:F}, Path=price**}"/>

Ознакомиться с программным кодом можно по ссылке: <https://github.com/Kseniya2004/TrainSchedule.git>

# Список литературы

# Приложения

Приложение 1. «Программный код интерфейса расписание поездов»

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="4\*"/>

</Grid.RowDefinitions>

<Menu Grid.Row="0"

Height="25">

<Menu.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#FF007E86" Offset="0.083"/>

<GradientStop Color="White" Offset="1"/>

</LinearGradientBrush>

</Menu.Background>

<MenuItem Header="Функции с данными">

<MenuItem Header="По пункту назначения"

Name="BtnWord"/>

<Separator/>

<MenuItem Header="Все станции"

/>

<Separator/>

<MenuItem Header="Кол-во поездов"

/>

<Separator/>

<MenuItem Header="Среднее время в пути"

Name="MIavr" Click="MIavr\_Click"/>

</MenuItem>

<MenuItem Header="Добавление"

Name="BtnAdd" Click="BtnAdd\_Click"/>

<MenuItem Header="Удаление"

Name="BtnDelete" Click="BtnDelete\_Click"/>

<MenuItem Header="Билеты"

Name="BtnTicket" Click="BtnTicket\_Click"/>

<MenuItem Header="Пассажиры"

Name="BtnPas" Click="BtnPas\_Click"/>

</Menu>

<DataGrid Name="DgridTrain"

AutoGenerateColumns="False"

IsReadOnly="True"

Grid.Row="1">

<DataGrid.Columns>

<DataGridTextColumn Width="0.5\*" Header="Пункт назначения"

Binding="{Binding des}"/>

<DataGridTextColumn Width="0.5\*" Header="Пункт отправления"

Binding="{Binding dep\_point}"/>

<DataGridTextColumn Width="0.5\*" Header="Номер поезда"

Binding="{Binding num}"/>

<DataGridTextColumn Width="0.6\*" Header="Время отправления"

Binding="{Binding dep\_time}"/>

<DataGridTextColumn Width="0.4\*" Header="Время в пути"

Binding="{Binding trav\_time}"/>

<DataGridTextColumn Width="0.7\*" Header="Станции"

Binding="{Binding stations}"/>

<DataGridTemplateColumn Width="\*">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Click="BtnEdit\_Click"

Content="Редактирование"

Width="150" Height="30"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<StackPanel Grid.Row="2" Background="#FF007E86" Orientation="Horizontal">

<StackPanel Orientation="Horizontal" VerticalAlignment="Top" Margin="5">

<TextBlock Text="Поиск:" Foreground="White" Height="30" Width="50"/>

<TextBox Name="TxtSearch" Height="30" Width="100" TextChanged="TxtSearch\_TextChanged"/>

</StackPanel>

<ComboBox Name="CmbAvr" Margin="10" Height="30" Width="100" SelectionChanged="CmbAvr\_SelectionChanged" VerticalAlignment="Top"/>

<StackPanel Orientation="Horizontal" Margin="10" VerticalAlignment="Top">

<StackPanel Orientation="Horizontal">

<TextBlock Text="Среднее время в пути:" Foreground="White" Margin="5" Height="30"/>

<TextBlock Name="txtAvr" Foreground="White" Height="30"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Количество поездов со станции:" Foreground="White" Margin="5" Height="30"/>

<TextBlock Name="txtCount" Foreground="White" Height="30"/>

</StackPanel>

</StackPanel>

</StackPanel>

</Grid>