Инструкция по созданию древовидного представления БД в WPF

Оглавление

[1. Создание БД 3](#_Toc60930776)

[2. Подключение к бд 5](#_Toc60930777)

[3. Интерфейс 6](#_Toc60930778)

[4. Сущности и Контекст 10](#_Toc60930779)

[5. Репозитории 13](#_Toc60930780)

[TGROUP Repository 14](#_Toc60930781)

[Конструктор: 14](#_Toc60930782)

[Создание группы: 14](#_Toc60930783)

[Удаление группы: 15](#_Toc60930784)

[Обновление группы: 16](#_Toc60930785)

[Чтение группы: 16](#_Toc60930786)

[TRELAION Repository 18](#_Toc60930787)

[Конструктор и создание отношения: 18](#_Toc60930788)

[Удаление отношения: 18](#_Toc60930789)

[Обновлние отношения: 19](#_Toc60930790)

[Чтение отношения: 19](#_Toc60930791)

[TPROPERTY Repository 20](#_Toc60930792)

[Конструктор и создание свойства: 20](#_Toc60930793)

[Удаление свойства: 20](#_Toc60930794)

[Обновлние свойства: 21](#_Toc60930795)

[Чтение свойства: 21](#_Toc60930796)

[MainRepository 22](#_Toc60930797)

[6. Отображение дерева 23](#_Toc60930798)

[7. Работа с фокусом 25](#_Toc60930799)

[7. Добавление сущности 28](#_Toc60930800)

[8. Редактирование сущности 30](#_Toc60930801)

[9. Удаление сущности 32](#_Toc60930802)

# 1. Создание БД

1. На предварительном шаге необходимо запустить клиент MS SQL Server.
2. Создается база данных с названием **Core\_Entity**.
3. Создается таблица с названием **TGROUP**, которая будет хранить перечень групп. Таблица **TGROUP** состоит из двух полей: **id** (уникальный идентификатор группы), **name** (наименование группы). Поле **id** является уникальным (первичным) и имеет тип int, полю **name** следует назначить тип varchar(255).
4. Далее необходимо заполнить таблицу **TGROUP** следующими данными:  
   insert TGROUP

values ('Первая группа');

insert TGROUP

values ('Вторая группа');

insert TGROUP

values ('Третья группа');

insert TGROUP

values ('Четвертая группа');

insert TGROUP

values ('Пятая группа');

1. Создается таблица **TRELATION** со следующими полями: **id\_parent** (идентификатор родительской группы), **id\_child** (идентификатор дочерней группы). Поля таблицы **TRELATION** имеют следующие параметры: **id\_parent** – внешний ключ к полю **id** таблицы **TGROUP**, **id\_child** – внешний ключ к полю **id** таблицы **TGROUP**. Поля (**id\_parent**, **id\_child**) образуют уникальный составной ключ таблицы **TRELATION**.
2. Заполнить таблицу **TRELATION** следующими данными:

insert TRELATION

values (1, 2);

insert TRELATION

values (1, 3);

insert TRELATION

values (2, 4);

1. Необходимо создать таблицу **TPROPERTY** со следующими полями: **id** – уникальный идентификатор (первичный ключ) свойства сущности, имеет целочисленный тип; **name** – наименование свойства сущности, имеет строковый тип данных; **value** – значение, хранящее свойство сущности, имеет строковый тип данных; **group\_id**–внешний ключ к полю id таблицы TGROUP, это поле однозначно определяет привязку свойства к группе, это поле имеет целочисленный тип.
2. Заполнить таблицу **TRELATION** следующими данными:   
   insert TPROPERTY   
   values ('Свойство 1', 'Описание свойства 1', 1);  
   insert TPROPERTY   
   values ('Свойство 2', 'Описание свойства 2', 3);

# 2. Подключение к бд

1. Создайте проект WPF (.NET FRAMEWORK)
2. Устанавливается Entity Framework в приложение TestWinApp с помощью инструмента NuGet
3. Настройка подключения к базе данных MS SQL Server.   
   В окне «Обозреватель решений» приложения TestWinApp находится файл «App.config». После тела тега «configSections» (см. конец параметра </configSections>) добавляется следующий код настройки подключения в базе данных.

<connectionStrings>

<add name="TestConnection" connectionString="Data Source=DQREST\SQLEXPRESS;Initial Catalog=test\_base;Integrated Security=True;" providerName="System.Data.SqlClient"/>

</connectionStrings>

В вышеуказанном коде в теге name указывается наименование подключения (название придумывает сам программист, например TestConnection), connectionString – адрес сервера.

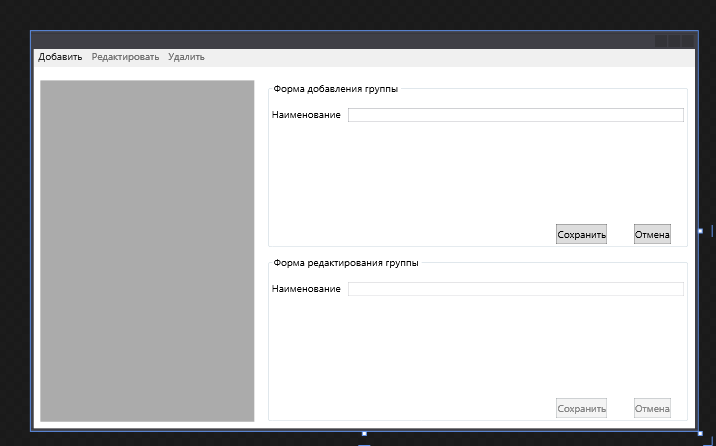
# 3. Интерфейс

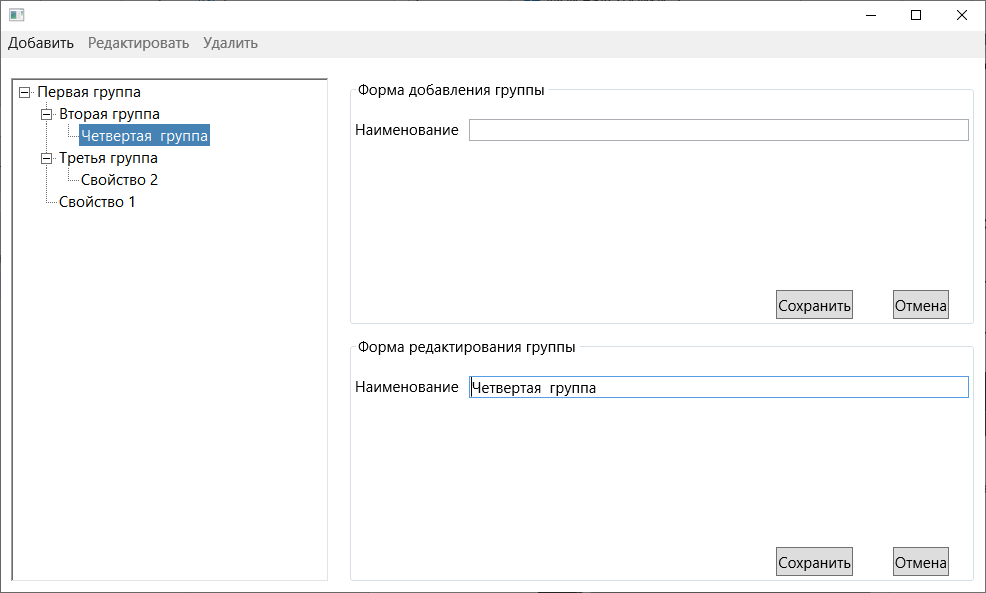
Интерфейс будет такого вида. Сверху справа будут панели добавления групп/свойств, снизу — редактирования. Они будут сменять друг друга с помощью свойства Visibility.

При нажатии на кнопку добавить будет вызываться меню с выбором формы добавления (группы или свойства).

Кнопки Редактировать и Удалить по умолчанию выключены, активируются при выборе узла в дереве. Формы редактирования тоже по умолчанию выключены, активируются при нажатии на кнопку Редактировать и в форму автоматически вбиваются все данные сущности.

Кнопка отмена на всех формах удаляет текст из текст боксов



Т.К. в приложении реализуется древовидная структура, сущности будут добавляться к другим Выбранным сущностям, поэтому они по возможности всегда будут подчеркиваться 

<Grid >

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="20\*"/>

</Grid.RowDefinitions>

<Grid Grid.Row="0">

<Menu>

<MenuItem x:Name="AddMenuButton" Header="Добавить">

<MenuItem x:Name="AddTgroupMenuButton" Header="Группу" Click="AddTgroupMenuButton\_Click" />

<MenuItem x:Name="AddTpropertyMenuButton" Header="Свойство" Click="AddTpropertyMenuButton\_Click"/>

</MenuItem>

<MenuItem x:Name="EditButton" Header="Редактировать" IsEnabled="False" Click="EditButton\_Click"/>

<MenuItem x:Name="DeleteButton" Header="Удалить" Click="DeleteButton\_Click" IsEnabled="False"/>

</Menu>

</Grid>

<Grid Grid.Row="1" Margin="0 20 0 0">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0" Margin="10 0 0 10">

<WindowsFormsHost x:Name="HostTreeView" Grid.ColumnSpan="2" />

</Grid>

<Grid Grid.Column="1" Margin="20 0 10 0">

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<Border Grid.Row="0" Margin="0 0 0 10">

<Grid>

<GroupBox x:Name="AddTgroupGroupBox" Header="Форма добавления группы" Visibility="Visible" GotFocus="AddTgroupGroupBox\_GotFocus" LostFocus="AddTgroupGroupBox\_LostFocus">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="20\*"/>

<RowDefinition Height="3\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.RowSpan="2" Grid.Row="0">

<DockPanel Margin="0 20 0 0 ">

<TextBlock Text="Наименование"/>

<TextBox x:Name="AddTgroupNameTextBox" Margin="10 0 0 0"/>

</DockPanel>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right">

<Button x:Name="AddTgroupSaveButton" Content="Сохранить" Margin="0 0 40 0" Click="AddTgroupSaveButton\_Click"/>

<Button x:Name="AddTgroupCancelButton" Content="Отмена" Margin="0 0 20 0" Click="AddTgroupCancelButton\_Click"/>

</StackPanel>

</Grid>

</GroupBox>

<GroupBox x:Name="AddTpropertyGroupBox" Header="Форма добавления свойства" Visibility="Hidden" GotFocus="AddTpropertyGroupBox\_GotFocus" LostFocus="AddTpropertyGroupBox\_LostFocus">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="20\*"/>

<RowDefinition Height="3\*"/>

</Grid.RowDefinitions>

<StackPanel>

<Grid Margin="0 20 0 0 ">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<StackPanel>

<TextBlock Text="Наименование"/>

<TextBlock Text="Значение" Margin="0 20 0 0"/>

</StackPanel>

</Grid>

<Grid Grid.Column="1">

<StackPanel>

<TextBox x:Name="AddTpropertyNameTextBox"/>

<TextBox x:Name="AddTpropertyValueTextBox" Margin="0 20 0 0"/>

</StackPanel>

</Grid>

</Grid>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right">

<Button x:Name="AddTpropertySaveButton" Content="Сохранить" Margin="0 0 40 0" Click="AddTpropertySaveButton\_Click" />

<Button x:Name="AddTpropertyCancelButton" Content="Отмена" Margin="0 0 20 0" Height="29" VerticalAlignment="Top" Click="AddTpropertyCancelButton\_Click" />

</StackPanel>

</Grid>

</GroupBox>

</Grid>

</Border>

<Border Grid.Row="1" >

<Grid Margin="0 0 0 10">

<GroupBox x:Name="EditTgroupGroupBox" Header="Форма редактирования группы" Visibility="Visible" IsEnabled="False" LostFocus="EditTgroupGroupBox\_LostFocus" GotFocus="EditTgroupGroupBox\_GotFocus">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="20\*"/>

<RowDefinition Height="3\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.RowSpan="2">

<DockPanel Margin="0 20 0 0 ">

<TextBlock Text="Наименование"/>

<TextBox x:Name="EditTgroupNameTextBox" Margin="10 0 0 0"/>

</DockPanel>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right">

<Button x:Name="EditTgroupSaveButton" Content="Сохранить" Margin="0 0 40 0" Click="EditTgroupSaveButton\_Click"/>

<Button x:Name="EditTgroupCancelButton" Content="Отмена" Margin="0 0 20 0" Click="EditTgroupCancelButton\_Click"/>

</StackPanel>

</Grid>

</GroupBox>

<GroupBox x:Name="EditTpropertyGroupBox" Header="Форма редактирования свойства" Visibility="Hidden" IsEnabled="False" LostFocus="EditTpropertyGroupBox\_LostFocus" GotFocus="EditTpropertyGroupBox\_GotFocus">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="20\*"/>

<RowDefinition Height="3\*"/>

</Grid.RowDefinitions>

<StackPanel>

<Grid Margin="0 20 0 0 ">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<StackPanel>

<TextBlock Text="Наименование"/>

<TextBlock Text="Значение" Margin="0 20 0 0"/>

</StackPanel>

</Grid>

<Grid Grid.Column="1">

<StackPanel>

<TextBox x:Name="EditTpropertyNameTextBox"/>

<TextBox x:Name="EditTpropertyValueTextBox" Margin="0 20 0 0"/>

</StackPanel>

</Grid>

</Grid>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right">

<Button x:Name="EditTpropertySaveButton" Content="Сохранить" Margin="0 0 40 0" Click="EditTpropertySaveButton\_Click" />

<Button x:Name="EditTpropertyCancelButton" Content="Отмена" Margin="0 0 20 0" Click="EditTpropertyCancelButton\_Click" />

</StackPanel>

</Grid>

</GroupBox>

</Grid>

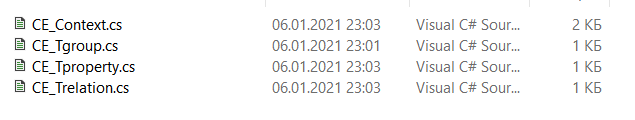
</Border>

</Grid>

</Grid>

</Grid>

# 4. Сущности и Контекст

Создайте папку EntitiesAndContext и поместите в неё следующие классы: 

В каждом классе нужно будет исправить namespace на тот, что указан в MainWindow.xaml.cs

В классах подключаются следующие пространства имен:

using System.ComponentModel.DataAnnotations.Schema;

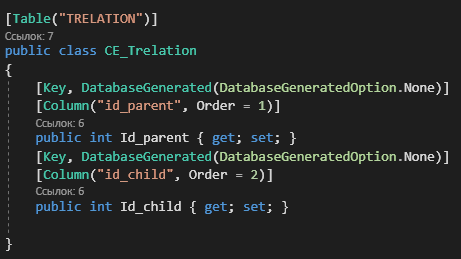
using System.ComponentModel.DataAnnotations;

using System.Data.Entity;

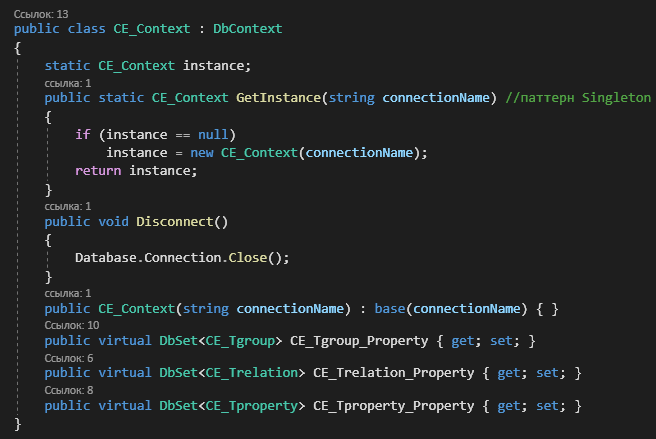
В классе CE\_Tgroup, CE\_Trelation Tproperty описываем его столбцы. Методы GetText(), GetName(), ConvertToTreeNode() нужны соответственно для того, чтобы конвертировать сущность в узел дерева.





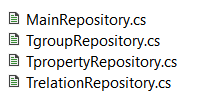


В классе CE\_Context определяем перечень таблиц, к которым разрешаем доступ на уровне приложения. Реализуем паттерн Singleton и создаем метод отключения от БД



# 5. Репозитории

Создайте папку Repositories и поместите в нее следующие классы



В каждом классе нужно будет исправить namespace на тот, что указан в MainWindow.xaml.cs

В классах подключаются пространство имен System.Windows

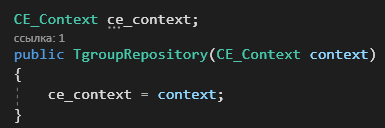
( using System.Windows;)

В данных репозиториях будем реализовывать CRUD’ы

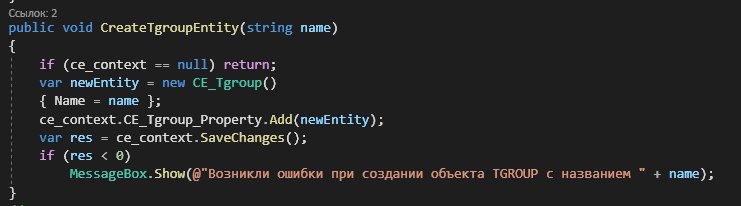
Начнем с TGROUP

## TGROUP Repository

### Конструктор:

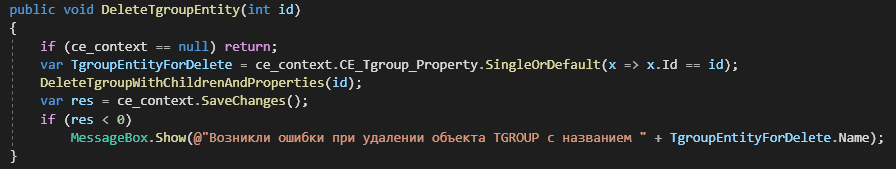
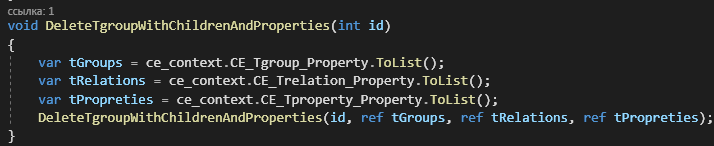
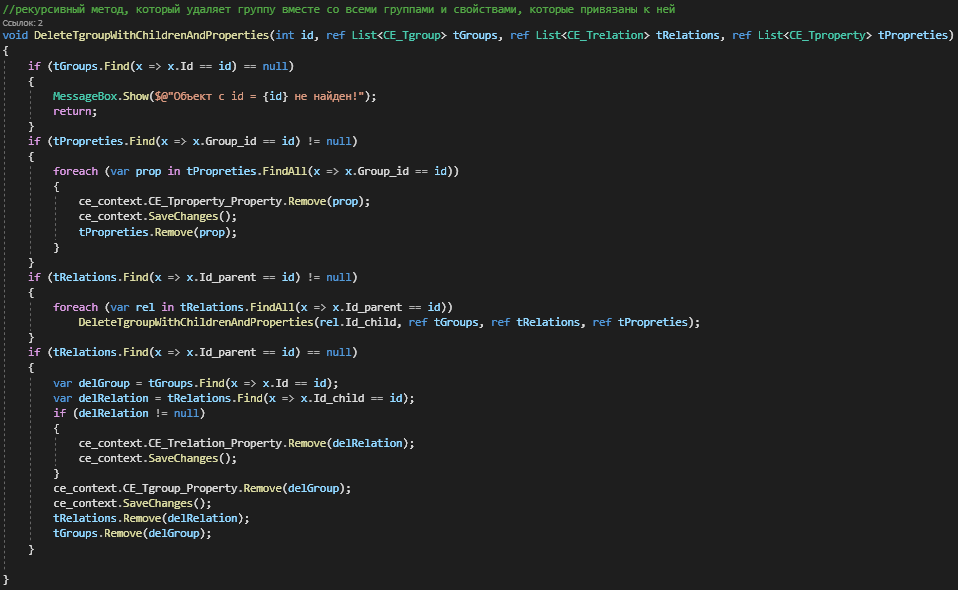


### Создание группы:

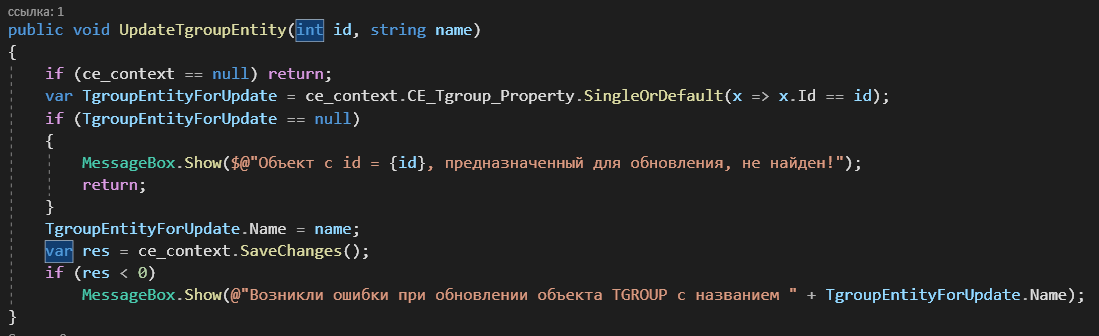


### Удаление группы:

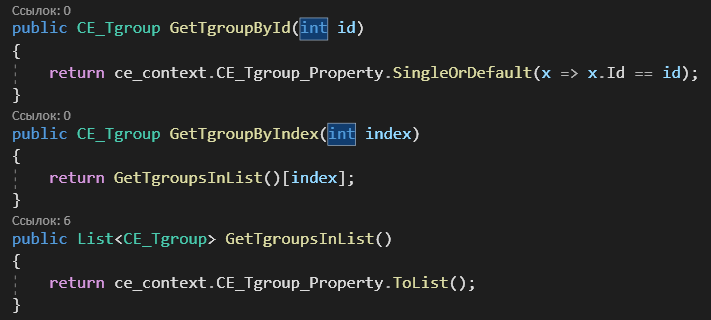
Происходит с помощью рекурсивной функции DeleteTgroupWithChildrenAndProperties(), которая на каждой итерации проверяет не привязано ли к текущей группе свойство и удаляет его. Далее если у группы есть ребенок, рекурсивная функция запускается уже с данными ребенка. Так до тех пор, пока функция не дойдет до группы без ребенка, эта группа удаляется вместе с TRELATION. Таким образом реализовывается каскадное удаление со стороны приложения

### Обновление группы:

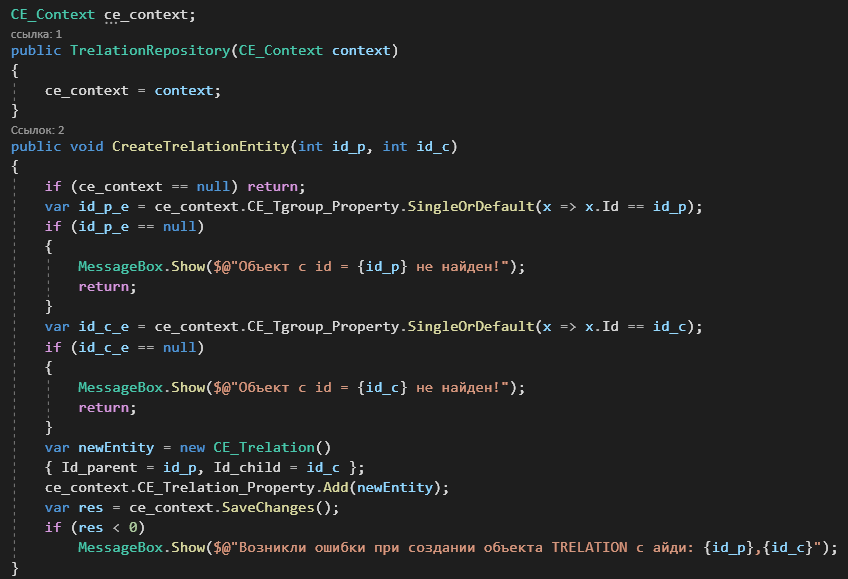


Чтение группы:

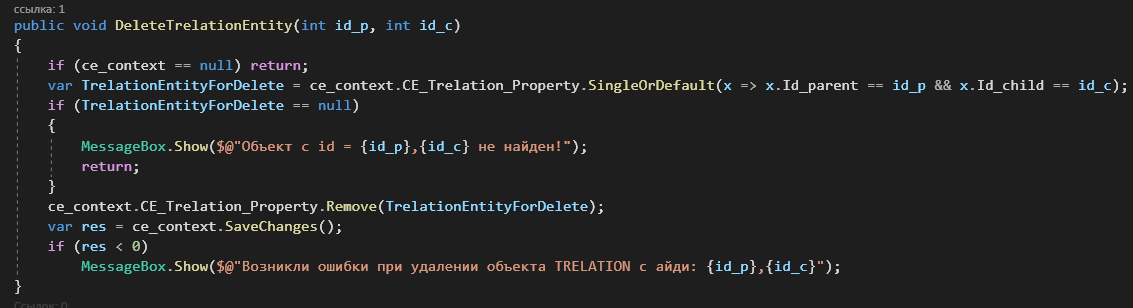


## TRELAION Repository

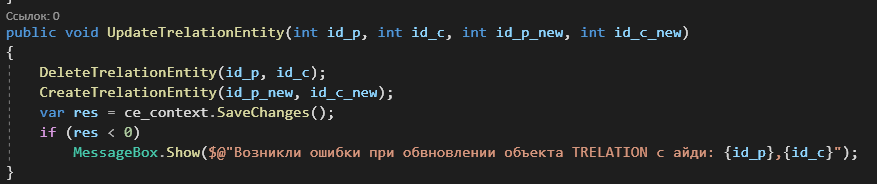
### Конструктор и создание отношения:



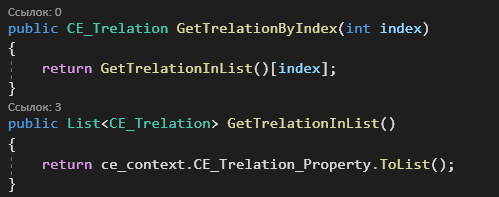
### Удаление отношения:



### Обновлние отношения:

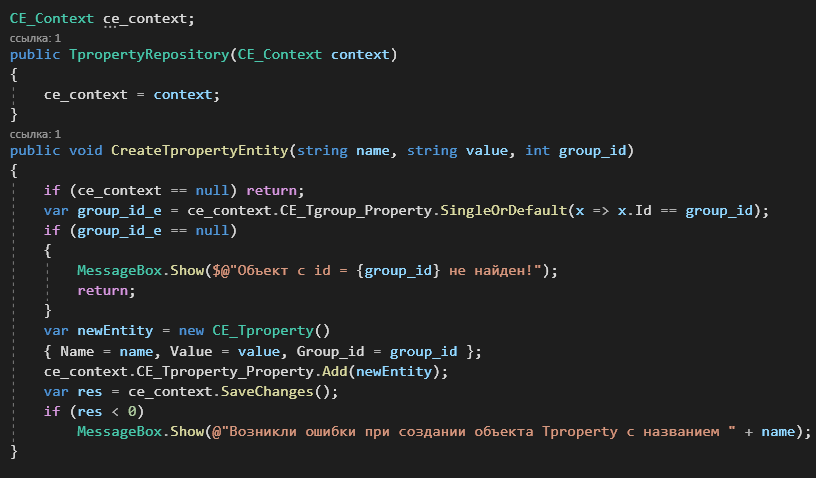


### Чтение отношения:

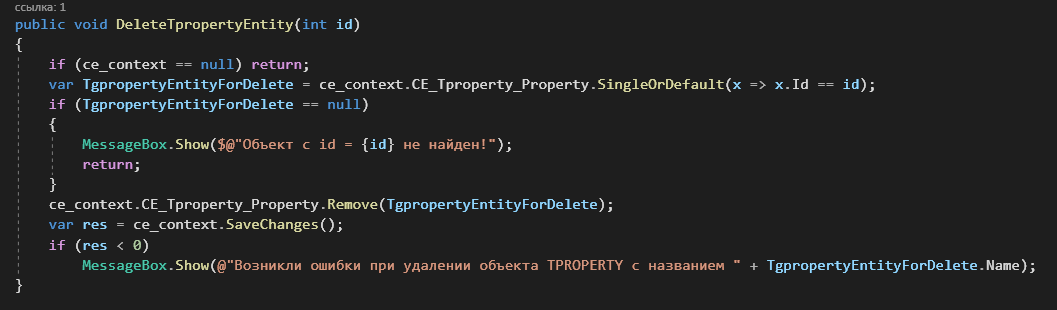


## TPROPERTY Repository

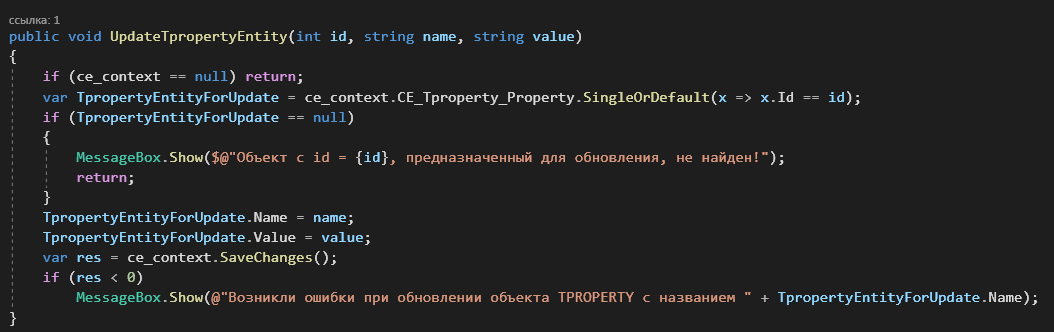
### Конструктор и создание свойства:



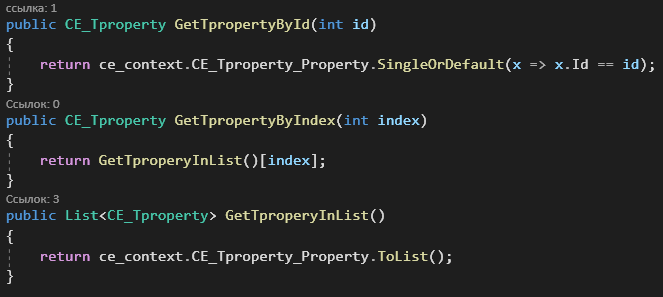
### Удаление свойства:



### Обновлние свойства:

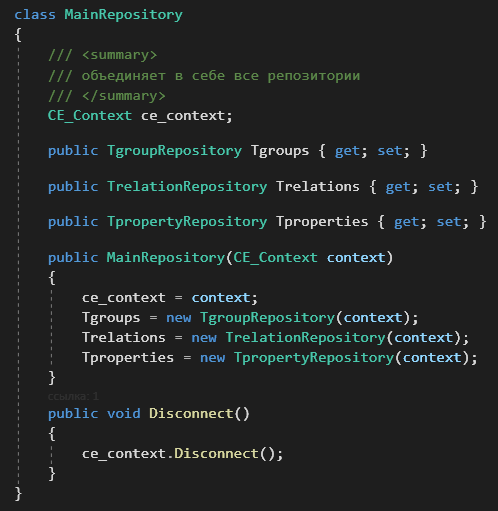


### Чтение свойства:



## MainRepository

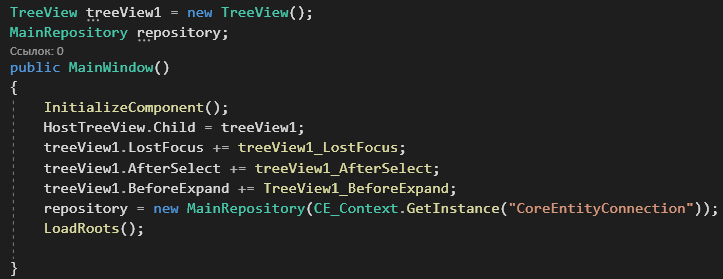
MainRepository объединяет в себе все созданные раннее репозитории



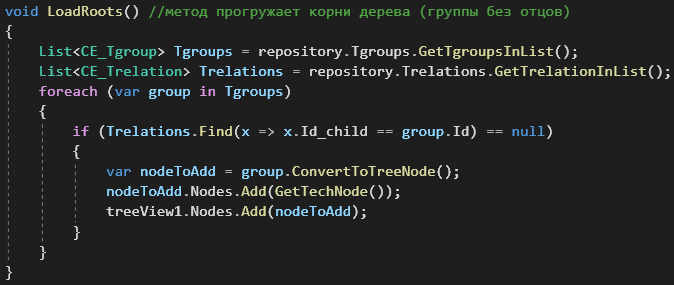
# 6. Отображение дерева

Нам понадобится пространство имен System.Windows.Forms; потому что ЭУ TreeView мы будем использовать оттуда. И соответственно нам понадобится ЭУ WindowsFormsHost, чтобы использовать TreeView из WinfowForms.

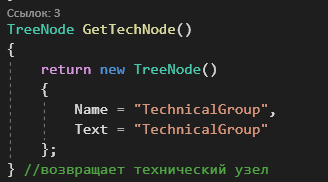
Подключаем наше дерево и инициализируем репозиторий



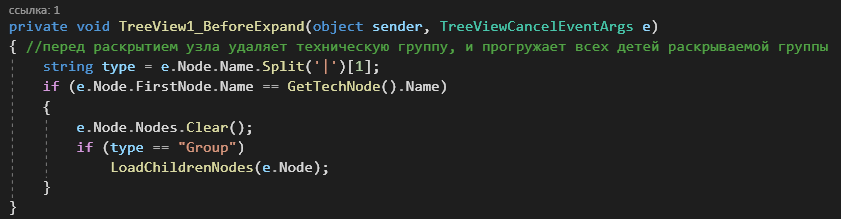
LoadRoots() прогружает корни дерева, т.е. группы без отцов. Всем этим группам добавляется технический узел, который будет удаляться при раскрытии основного узла.



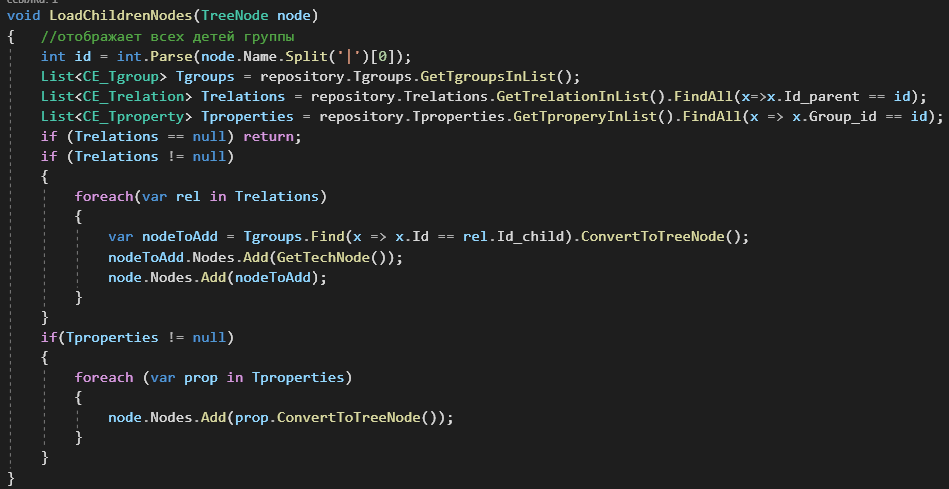
Метод для создание технического узла



Далее определите обработчик события TreeView1\_BeforeExpand(), который удаляет всех подузлы выбранного узла, если он первый узел был техническим и прогружает всех детей узла с помощью метода LoadChildrenNodes()



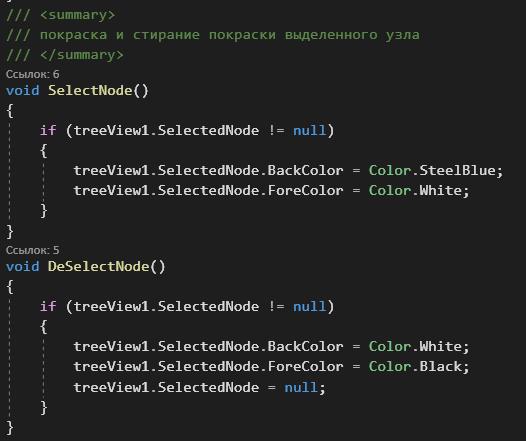
LoadChildrenNodes() отображает всех детей (группы и свойства) раскрываемого узла, предварительно найдя их в бд



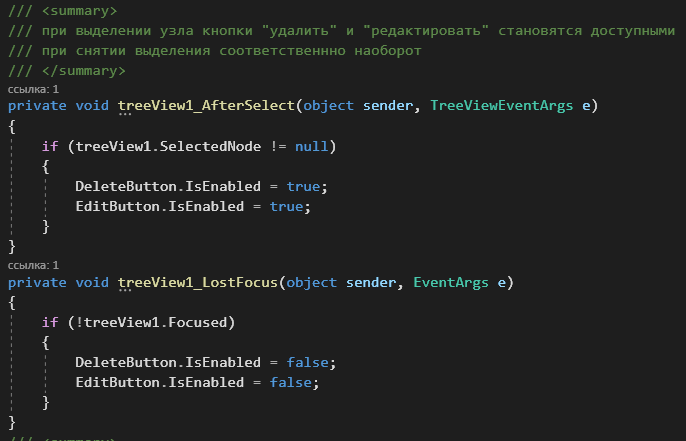
# 7. Работа с фокусом

Как говорилось в п.3. Интерфейс, мы будем для удобства пользователя подчеркивать выбранные раннее узлы, определим для этого методы. Для этого нам понадобится работа с фокусом на ЭУ.

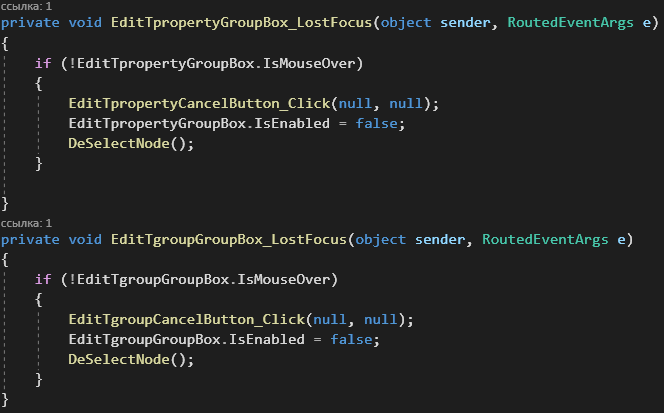
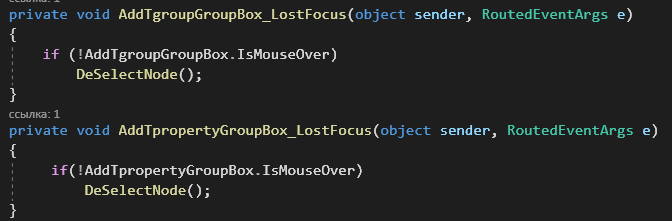
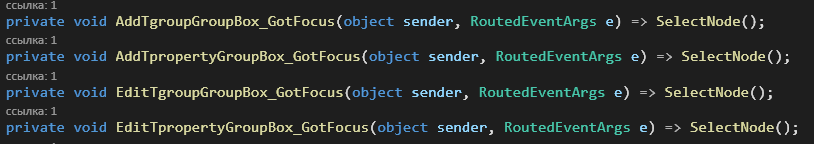
Определим следующие служебные методы:



Определим обработку фокуса на TreeView

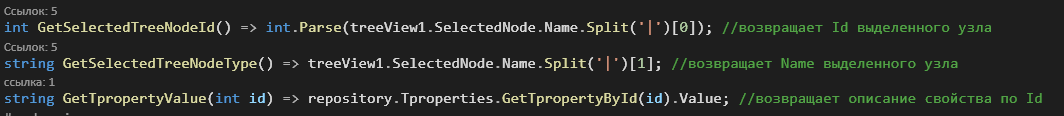


Определим обработчики событий фокуса на формах добавления/редактирования:

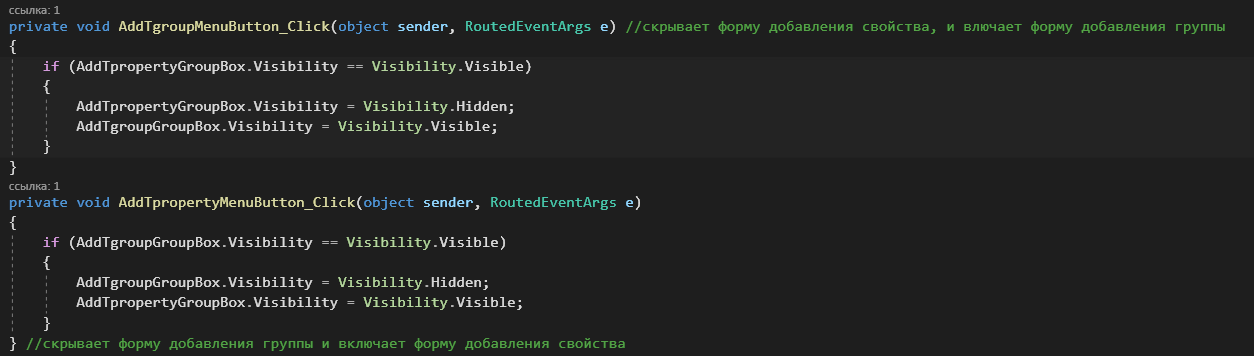


# 7. Добавление сущности

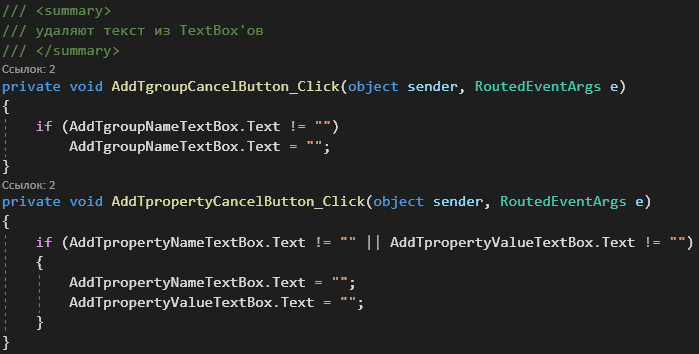
Определим следующие служебные методы, которые понадобятся и для функций редактирования и удаления сущностей.



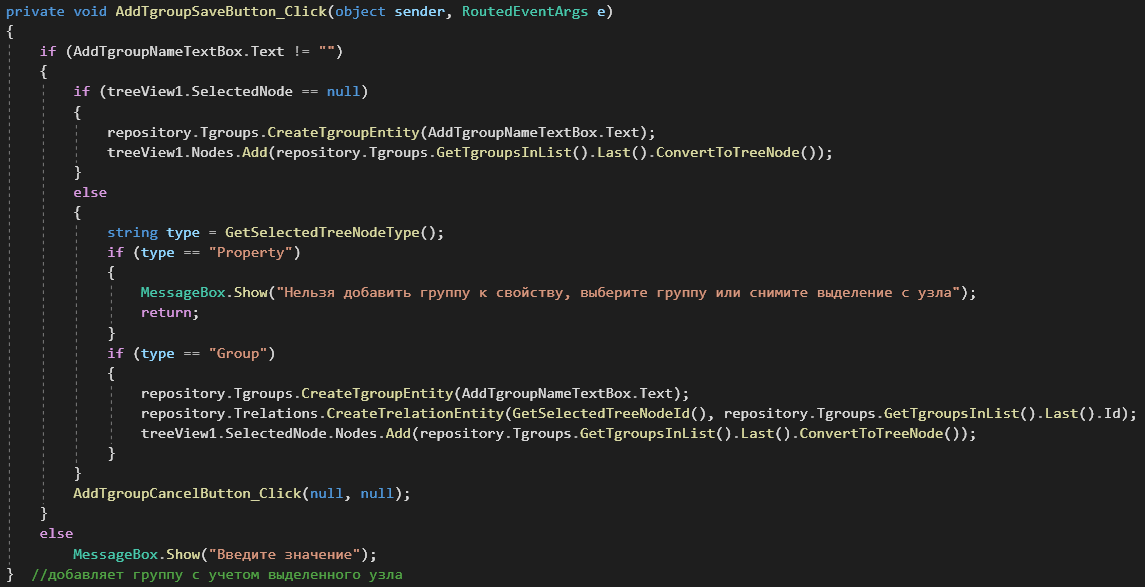
Определим следующие обработчики событий, чтобы при нажатии соотвествующей кнопки в меню, появлялась форма добавление нужного нам типа сущности



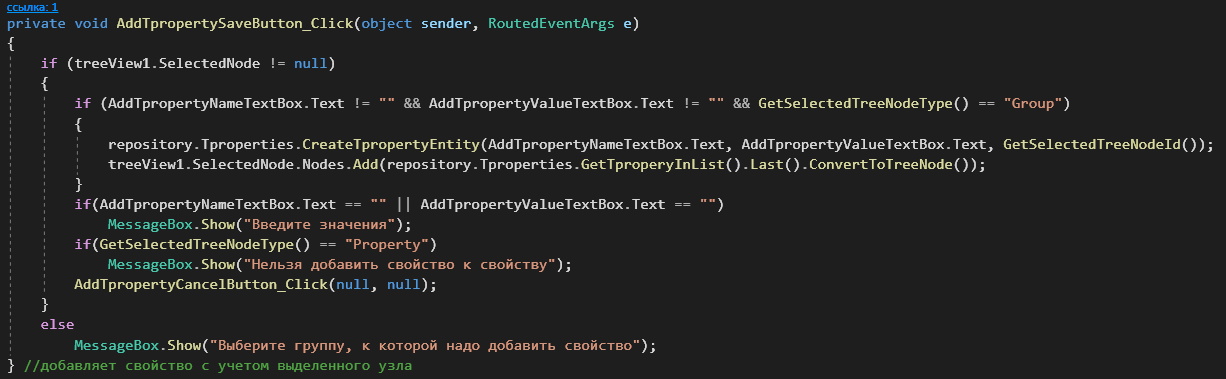
Добавим обработчики событий кнопки отмена на всех формах добавления:



Добавим обработчик события нажатия по кнопке Сохранить на форме добавления группы.  
Обработчик события сначала проверяет ввееден ли текст. Если узел не выбран, то просто добавляется группа и отображается в дереве. Если узел выбрал, и этот узел является группой, то группа создаетеся вместе с отношением (TRELATION), далее сбрасывается весь введеный текст

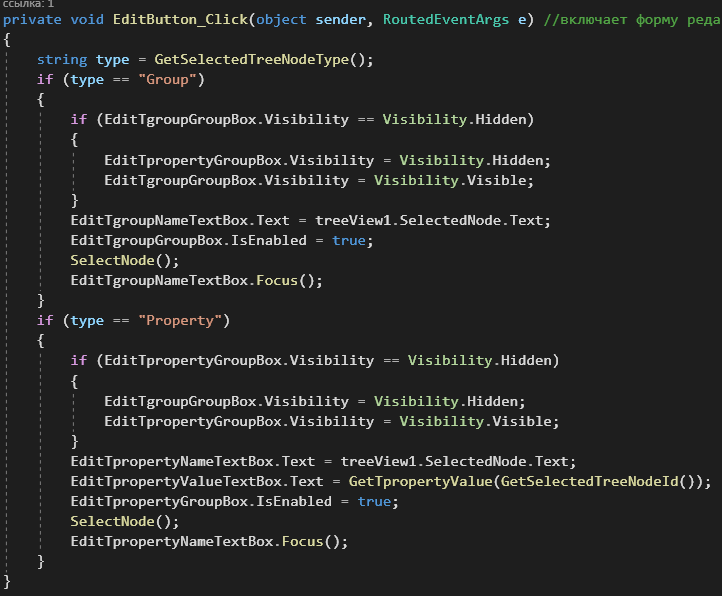


С добавлением свойство все по аналогии, только свойство нужно обязательно добавить к определенной группе.

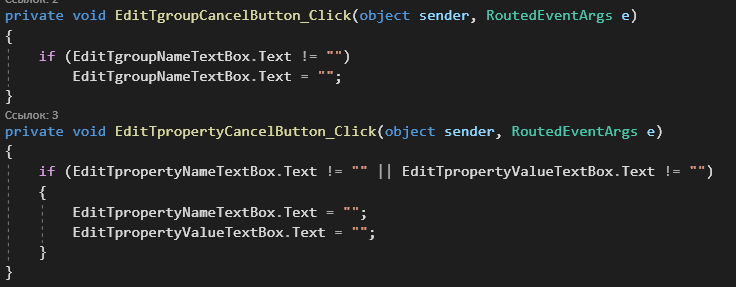


# 8. Редактирование сущности

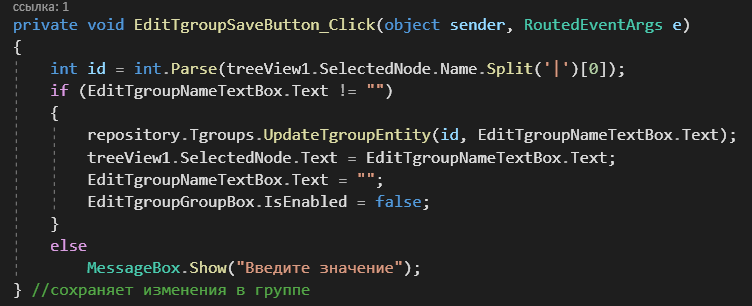
При нажатии на кнопку редактировать выходит форма редактирования выбранной сущности, все данные сущности зараннее вбиваются в текстБоксы, а выбранный раннее узел помечается для удобства пользователя.

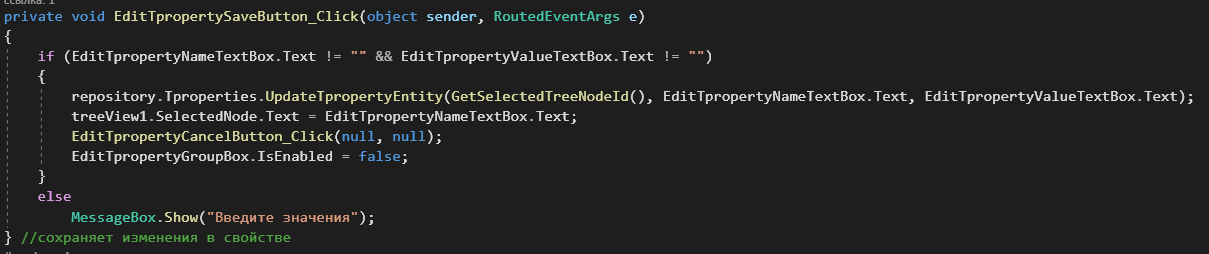


Определим обработчики событий нажатия по кнопке отмена на формах редактирования сущностей. Они также удаляют содержимое текстБоксов.



При нажатии на кнопку Сохранить в формах редактирования сущностей соотвественно обновляются ее поля и обновляется отображение узла в дереве.





# 9. Удаление сущности

При нажатии на кнопку Удалить, если узел выбран, он удаляется и соотвествующая сущность тоже. Кнопки Удалить и Редактировать выключаются

