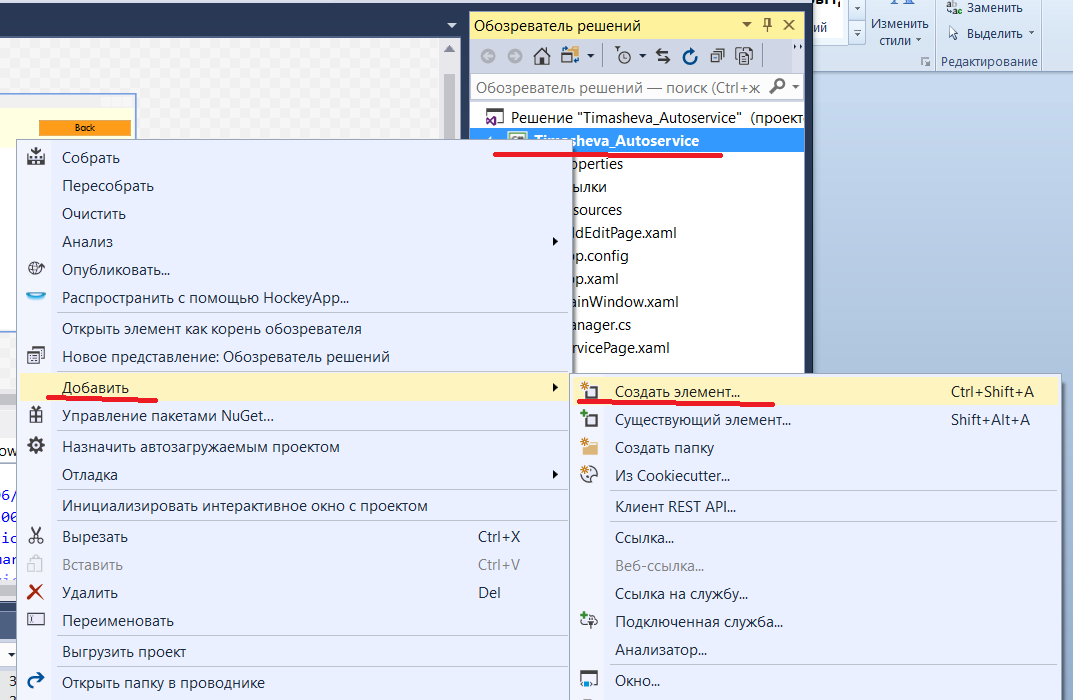
Присоединение БД к приложению. Разработка формы «Список услуг»

1. Работать будем в КОПИИ предыдущего проекта!

Выполним подключение базы данных к проекту:



**Выбрать Данные 🡪 Модель ADO.Net , ввести название модели FamiliyaModel**

2. Далее выбрать первый вариант.

3. Нажать кнопку «Создать соединение».

4. Выбрать MS SQL Server в качестве источника данных.

5. Ввести имя сервера (**скопировать из Management Studio лучше**),   
*Проверку подлинности Windows оставляем*, и в пункте Подключение к базе данных выбираем из выпадающего списка название вашей БД Фамилия\_Autoservice.

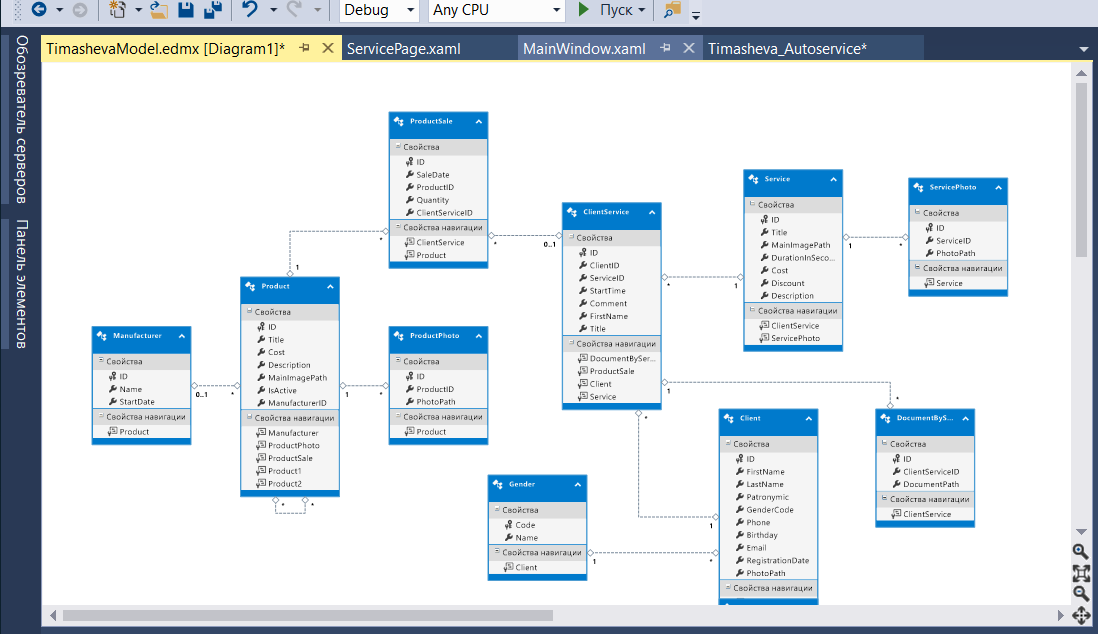
6. В открывшемся окне Мастера моделей оставляем имя соединения, которое предложено.

Далее оставляем наивысшую версию фреймворка:

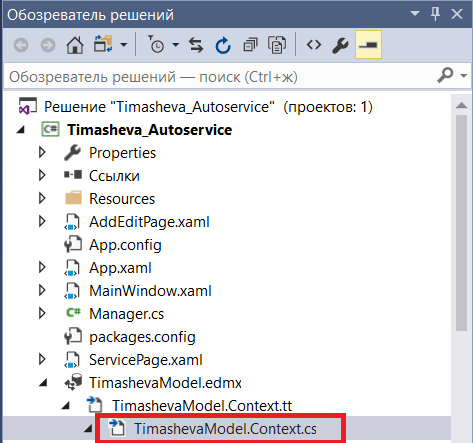
7. В открывшемся окне выбираем объекты БД для включения в модель: все таблицы ЗА ИСКЛЮЧЕНИЕМ **sysdiagrams.**

Жмем Готово, ждем результат присоединения модели (достаточно долго…)

По итогу примерно такая картина:

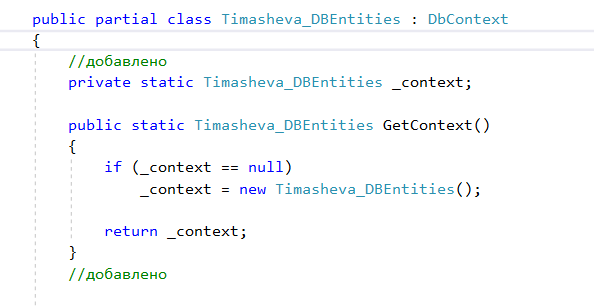


8. В Обозревателе решений откроем файл FamiliyaModel.Context.cs:



Добавляем строки кода для работы с моделью.

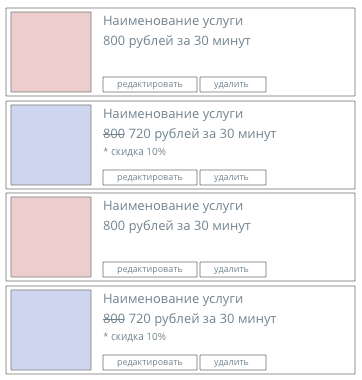
Ниже показано, куда нужно прописать эти строчки:



**Обязательно выполните сохранение проекта сейчас.**

9. База присоединена. Для отображения данных изменим страницу **ServicePage.xaml (форма Список услуг).**

В условии задачи имеется картинка с примерным расположением элементов на форме Список услуг:



После анализа этой картинки предполагаем следующее разделение:

две колонки – одна для картинки, вторая для четырех строк – 1)Название услуги (поле Title), 2)Цена (Cost) и длительность услуги (Duration), 3)Скидка (Discount), 4) кнопки редактировать и удалить.

Для вывода информации о сервисах будем использовать элемент управления ListView. Для вывода информации о сервисе создается шаблон элемента в списке — то представление, которое должно отображаться для каждого элемента.

Для этого:

Выполним разметку (курсивом выделено, что надо скопировать к себе):

<Grid>

***<ListView Name="ServiceListView" >***

***<ListView.ItemTemplate>***

***<DataTemplate>***

***<Grid>***

***<Grid.ColumnDefinitions>***

***<ColumnDefinition></ColumnDefinition>***

***<ColumnDefinition></ColumnDefinition>***

***</Grid.ColumnDefinitions>***

***<Grid.RowDefinitions>***

***<RowDefinition Height="50"/>***

***<RowDefinition Height="\*"/>***

***<RowDefinition Height="\*"/>***

***<RowDefinition Height="\*"/>***

*</Grid.RowDefinitions>*

*</Grid>*

*</DataTemplate>*

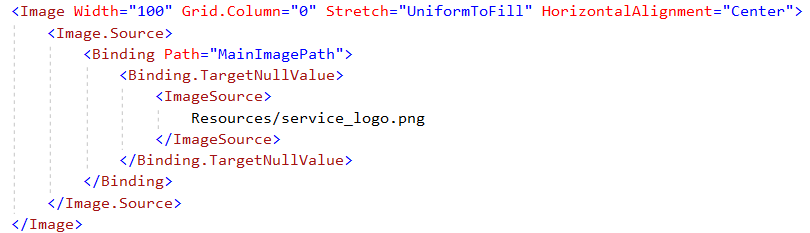
*</ListView.ItemTemplate>*

*</ListView>*

</Grid>

10. В ресурсах есть изображение, которое можно использовать для вывода при отсутствии картинки у услуги (сервиса) – это service\_logo.png, воспользуемся им далее в верстке.

Для этого добавляем изображение услуги (сервиса) в верстку, установив необходимые атрибуты, задаем объекты для привязки, а также изображение-заглушку, если изображения сервиса не будет:

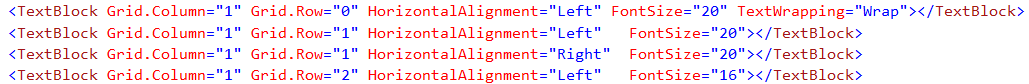


11. Добавим после рисунка текстовые поля:

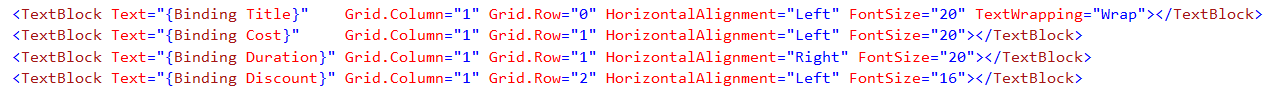
для **наименования**, указав: перенос текста, выравнивание, большой шрифт; для стоимости, указав: большой шрифт, выравниваем по левому краю;

для **длительности**, указав: большой шрифт, выравниваем по правому краю; для скидки – средний шрифт, по левому.

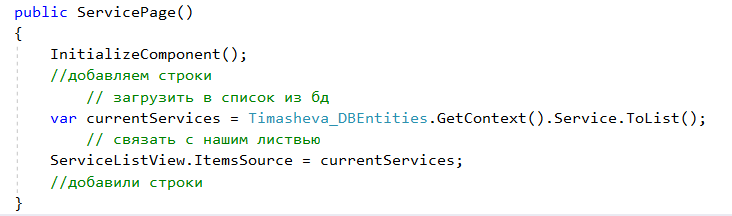
Располагаются все эти текстовые блоки в Колонке 1, но в разных строках:



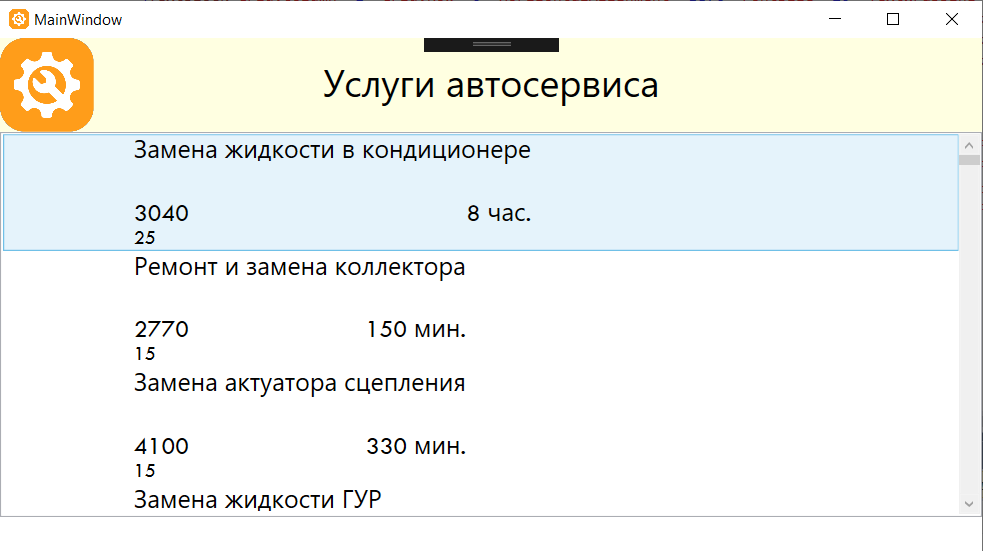
12. Теперь допишем в эти же текстовые блоки привязки (binding) для наименования (Title), стоимости (Cost), длительности (Duration) и скидки (Discount):



13. Загружаем коллекцию «список» в коде (F7):



Запускаем приложение:

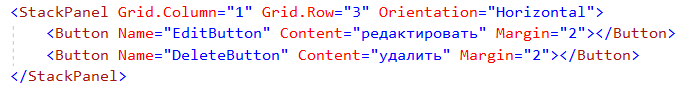


14. Поработаем над форматом отображения стоимости и скидки. Указываем им StringFormat формат:

<TextBlock Text="{Binding Cost, StringFormat={}{0:0} рублей за}" Grid.Column="1" Grid.Row="1" HorizontalAlignment="Left" FontSize="20"> </TextBlock>

<TextBlock Text="{Binding Discount, StringFormat=\*скидка {0} % }" Opacity="{Binding Discount}" Margin="5" Grid.Column="1" Grid.Row="2" HorizontalAlignment="Left" FontSize="16"></TextBlock>

15. Разместим на 4 строке кнопки Редактировать и Удалить:



16. Запустим, проверим.

Результат выполнения лаб.работы в архив, и в СДО.