С# в отличие от Java не умеет преобразовывать строку сразу в Объект с которым можно легко работать при помощи методов put/get.

В C# необходимо сначала создать шаблон данных, на который и будет разбиваться строка при создании Json-объекта.

Пример:

[

{

"Address": "г. Санкт-Петербург, Богатырский проспект, 23",

"Station\_ID": 1,

"data": [{

"Name": "92",

"Price": 44.09,

"AmountOfFuel": 43360

}, {

"Name": "95",

"Price": 47.67,

"AmountOfFuel": 50149

}, {

"Name": "98",

"Price": 56.2,

"AmountOfFuel": 17943

}, {

"Name": "Disel Fuel",

"Price": 48.5,

"AmountOfFuel": 30075

}]

},

{

"Address": "г. Санкт-Петербург, Витебский проспект, 9к2 литера А",

"Station\_ID": 2,

"data": [{

"Name": "92",

"Price": 44.49,

"AmountOfFuel": 56531

}, {

"Name": "95",

"Price": 47.85,

"AmountOfFuel": 81209

}, {

"Name": "98",

"Price": 56.09,

"AmountOfFuel": 49249

}, {

"Name": "Disel Fuel",

"Price": 49.69,

"AmountOfFuel": 82789

}]

}

]

Данный файл содержит информацию о

* Заправке
  + Адрес
  + ID
* Топливе
  + Цена
  + Марка
  + Доступное количество

Если придерживаться 3НФ, это информация из 2х таблиц:

* Станция
* Топливо на станции

Есть 2 варианта импорта данных.

* Импортировать данные в готовую БД
* Импортировать данные в пустую БД

**Пустая БД**

Visual Studio умеет быстро преобразовывать Json данные в шаблоны для Json-объектов.

Для этого необходимо

1. Скопировать общий элемент данных. В данном случае

{

"Address": "г. Санкт-Петербург, Богатырский проспект, 23",

"Station\_ID": 1,

"data": [{

"Name": "92",

"Price": 44.09,

"AmountOfFuel": 43360

}, {

"Name": "95",

"Price": 47.67,

"AmountOfFuel": 50149

}, {

"Name": "98",

"Price": 56.2,

"AmountOfFuel": 17943

}, {

"Name": "Disel Fuel",

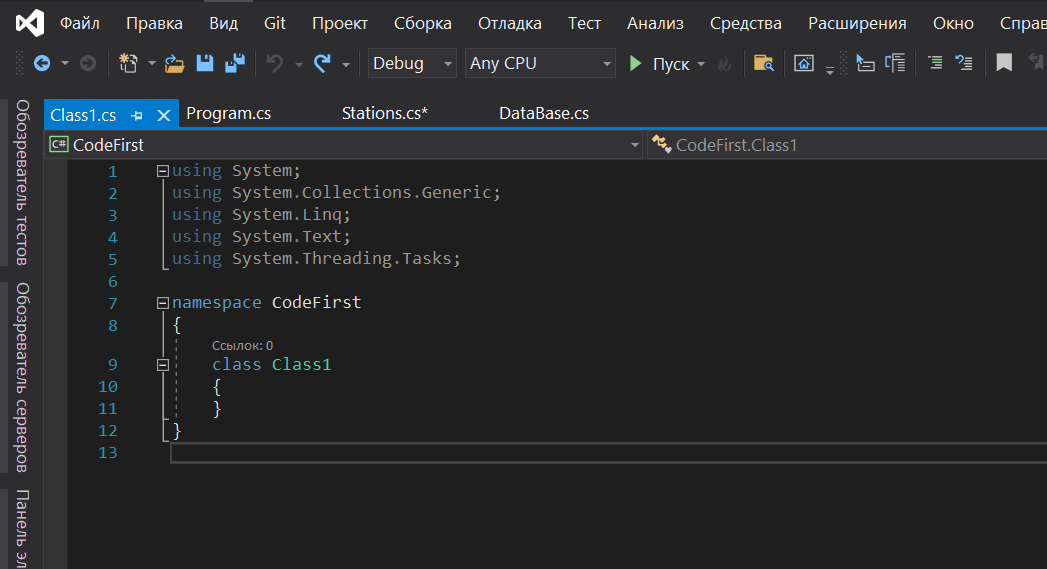
"Price": 48.5,

"AmountOfFuel": 30075

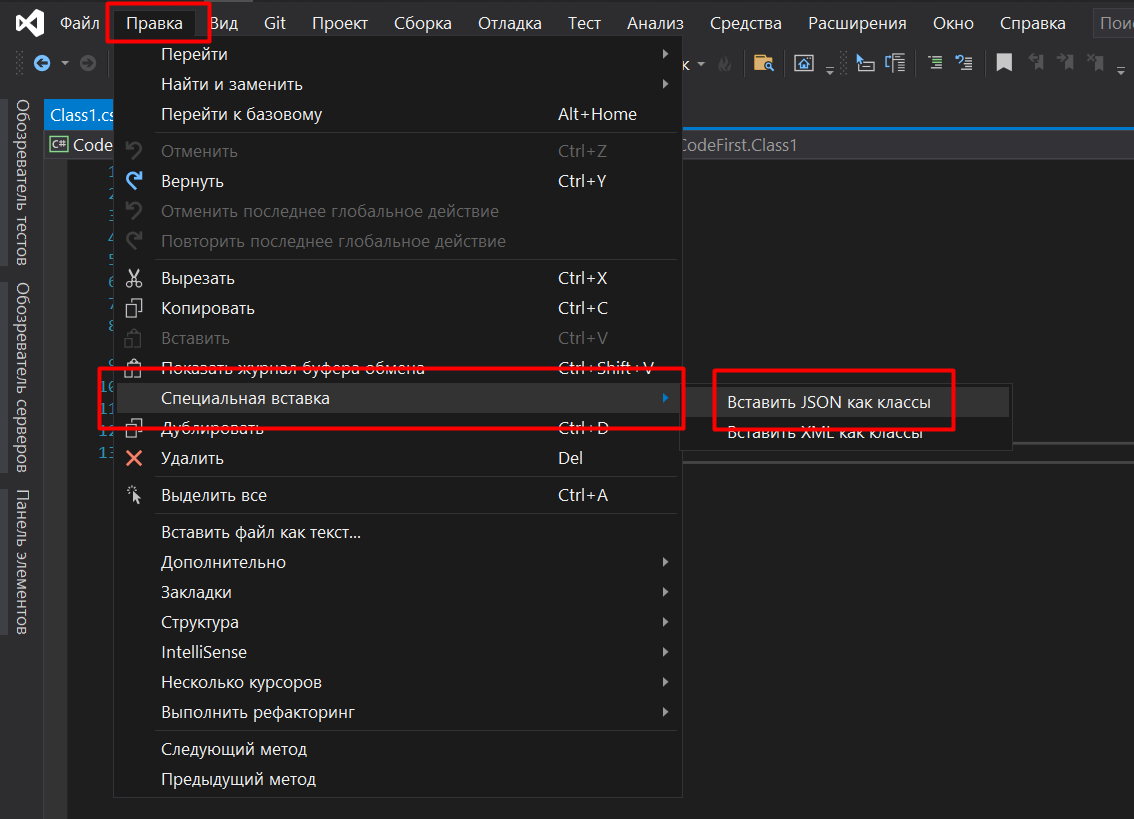
}]

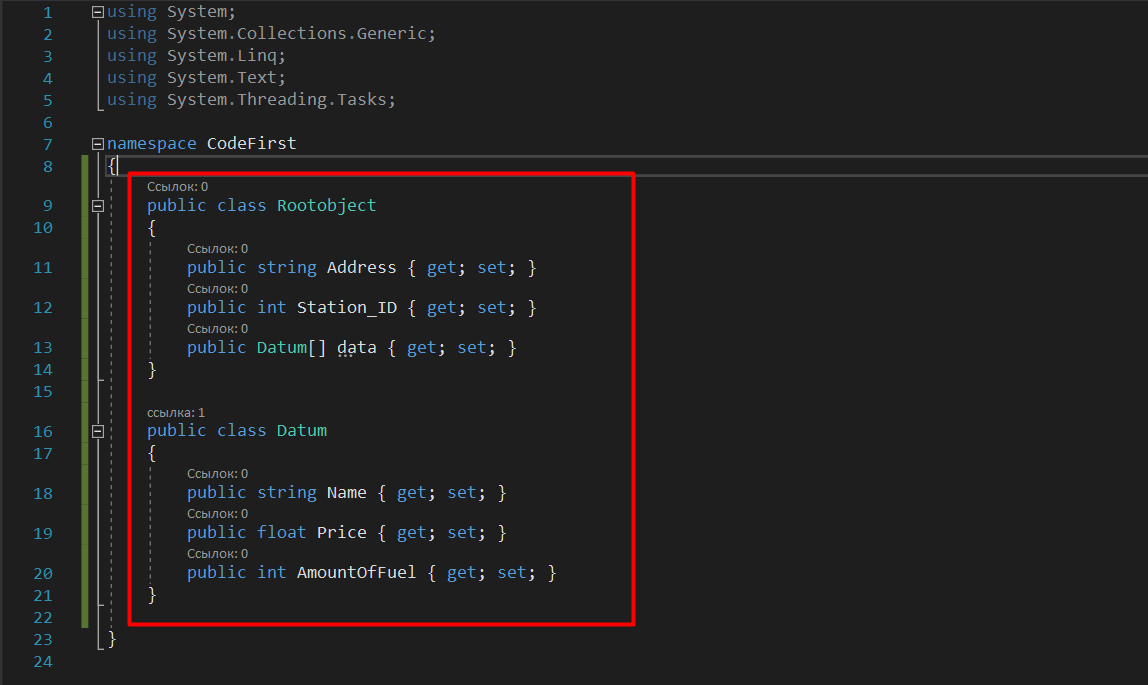
}

1. Создать новый класс

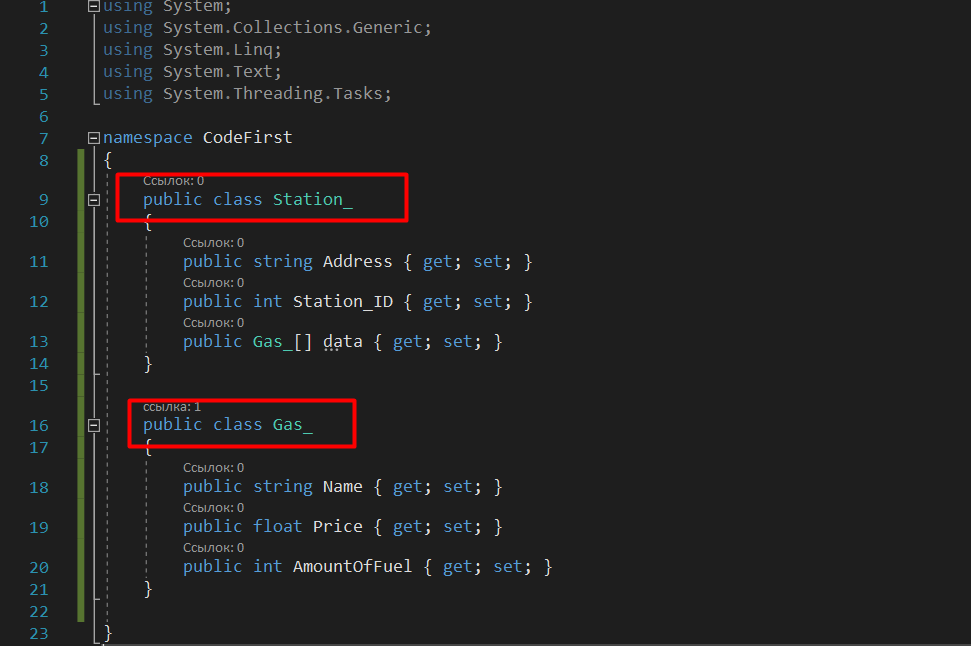


1. Выполнить специальную вставку через Visual Studio

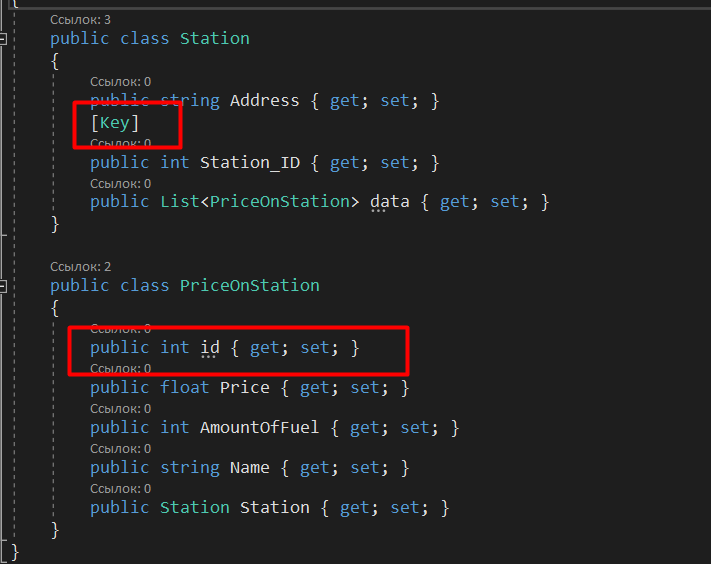




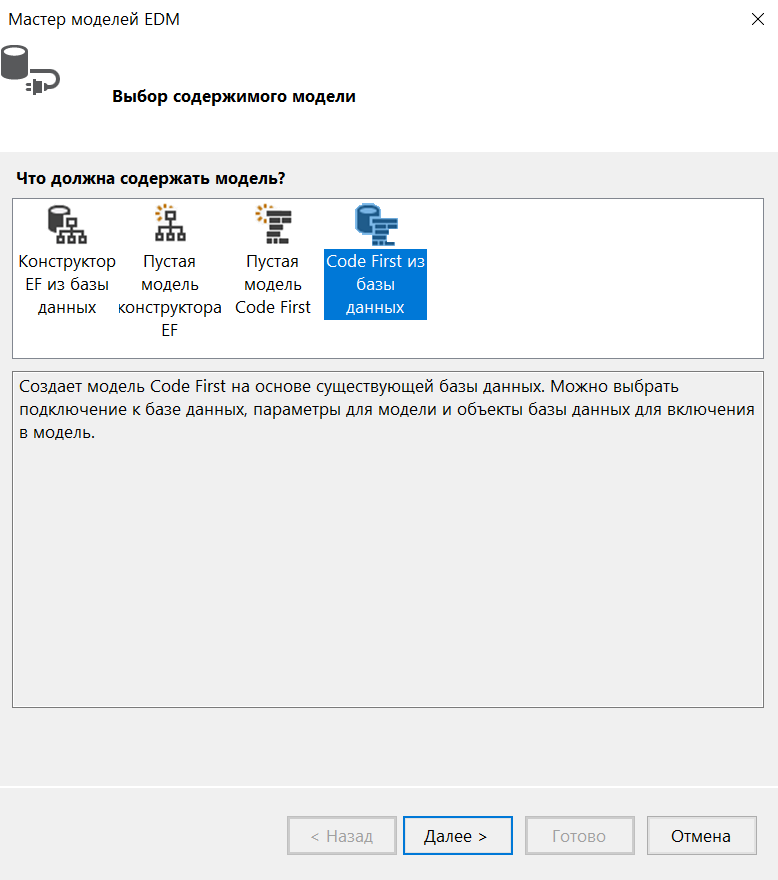
1. Необходимо переименовать классы так как должны называться таблицы в БД.

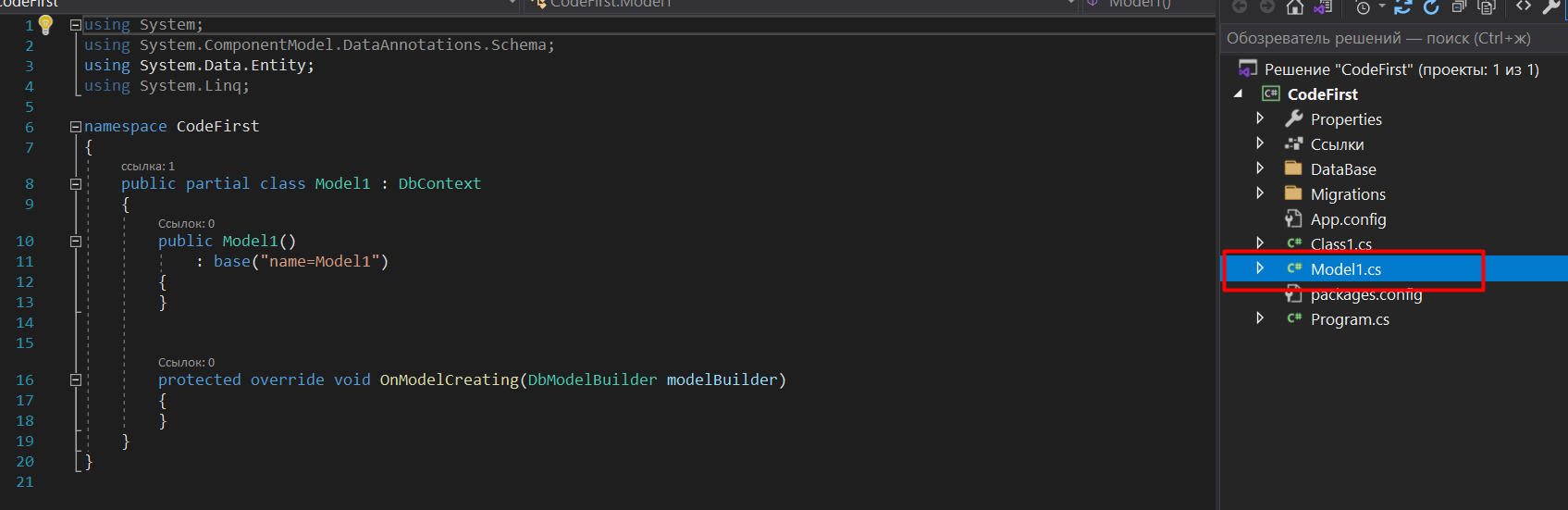


1. Необходимо добавить ID, иначе модель не сработает.



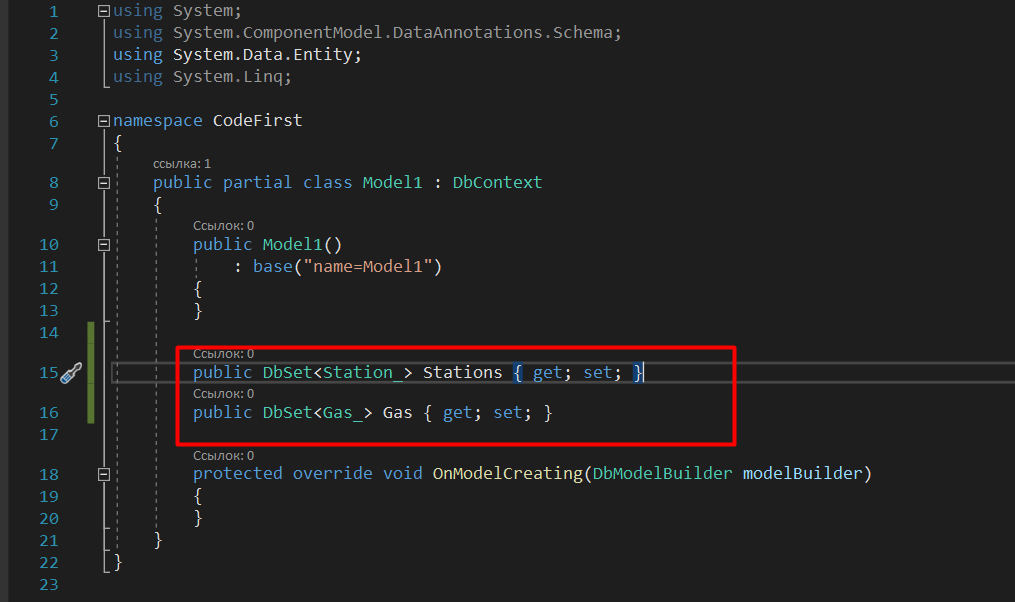
1. Добавить Model Ado.Net через CodeFirst



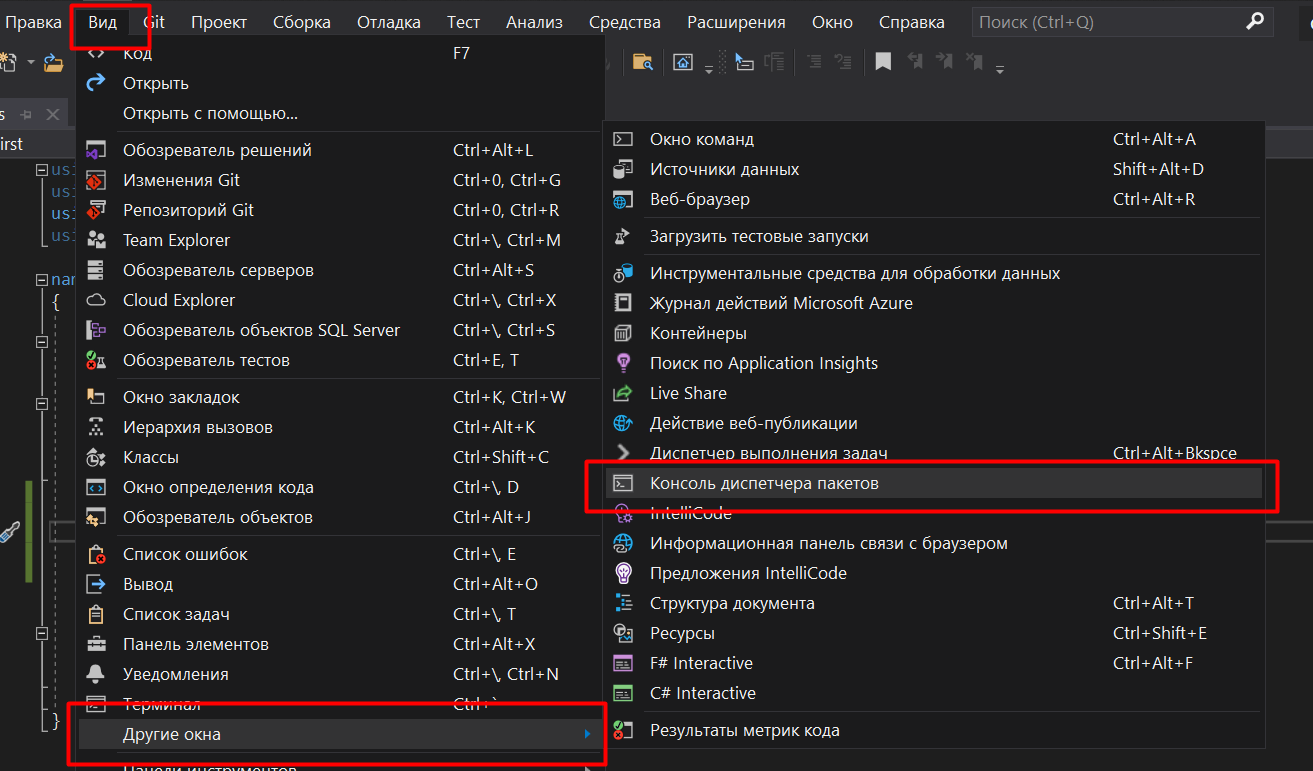


В результате создастся только 1 файл, а не целая модель с Визуальным интерфейсом.

1. Необходимо в этом файле зарегистрировать созданные ранее «таблицы» (Station\_, Gas\_)



1. Необходимо выполнить миграции в БД.



Ввести команды

* Enable-migrations // только первый раз для БД
* Add-migration “name migration” // после каждых изменений и последующего запуска кода
* Update-database // применить миграции

По данному Json-файлу созданы таблицы можно приступать к импорту данных. Дальнейшие шаги общие для пустой и готовой БД.

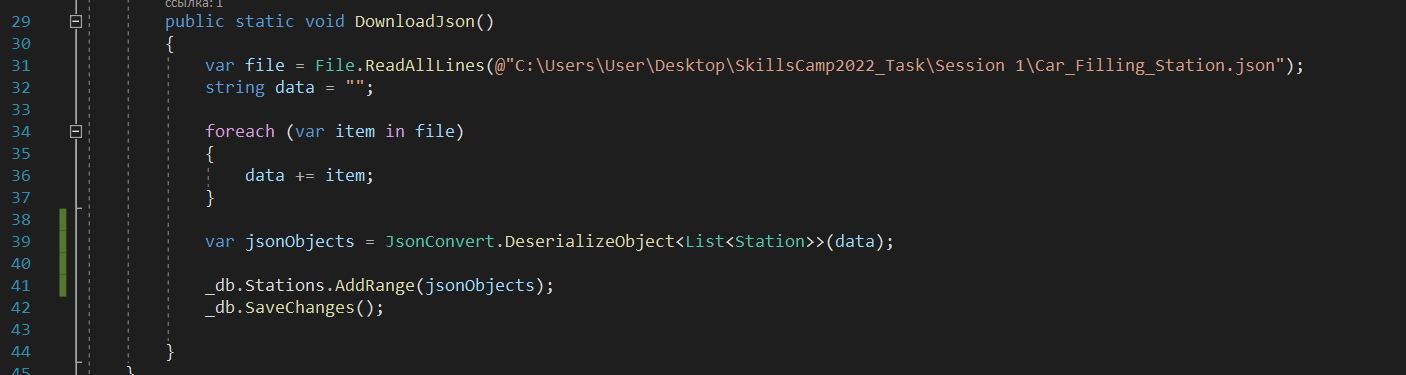
**Импорт данных в БД**

Для работы с Json, есть специальная библиотека, но она не добавляется по умолчанию в проекты, но является частью ASP.NET.

Другого способа найти данную библиотеку я пока не нашел.

1. Создать решение ASP.NET. (Веб-сервер)
2. Найти ссылку на Newtonsoft.Json
3. Зайти в свойства данной ссылки и скопировать путь до dll
4. Вернуться в решение для импорта данных, Добавить ссылку – обзор – обзор – вставить путь к папке с dll (обрезать ссылку до папки а не файла) – выбрать dll.

Остается импортировать сами данные



1. Прочитать строки из файла
2. Соединить все строки в одну
3. Создать с помощью метода из Newtonsoft.Json объекты понятные для C#. Где List<> - указывает что в данных находится массив значений, а Station – указывает на то что главной таблицей является Station. Вложенную таблицу он сам заполнит данными. (Но если происходит импорт из готовой модели, необходимо будет заменить ссылки на внешние таблицы в Json-файле, через replace(). Он очень требовательный к переменным, не совпадает название – ставит null)
4. Добавить записи в главную таблицу.
5. Сохранить все изменения
6. Данные импортированы во все связанные таблицы в БД.