МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ “ВИТЕБСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ П.М. МАШЕРОВА”

ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

КАФЕДРА ГРАЖДАНСКОГО ПРАВА И ГРАЖДАНСКОГО ПРОЦЕССА

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине «Современное построение  
реляционных баз данных»**

Автоматизация слияния ADO.NET SQL-таблиц с использованием XML-сериализации

Белькевич Глеб   
Александрович  
3 курс  
31 группа

Проверил:  
Ермоченко Сергей   
Александрович

Витебск, 2021

Оглавление

[Реферат 4](#_Toc60686260)

[Перечень ключевых слов 4](#_Toc60686261)

[Обзор используемой аппаратуры, инструментов 4](#_Toc60686262)

[MS SQL Server 4](#_Toc60686263)

[C# 4](#_Toc60686264)

[Winforms 5](#_Toc60686265)

[Теоретический блок. Термины. Сокращения 5](#_Toc60686266)

[C# 5](#_Toc60686267)

[MS SQL 5](#_Toc60686268)

[WinForms 8](#_Toc60686269)

[ADO.NET 8](#_Toc60686270)

[Введение 9](#_Toc60686271)

[Цели и задачи 9](#_Toc60686272)

[Объект и предмет исследования 9](#_Toc60686273)

[Акутальность и область применения 9](#_Toc60686274)

[Обзор литературных источников 9](#_Toc60686275)

[Ход работы 11](#_Toc60686276)

[Процесс разработки 11](#_Toc60686277)

[Каркас приложения 11](#_Toc60686278)

[Создание подключения 13](#_Toc60686279)

[Создание запросов 14](#_Toc60686280)

[Создание таблиц в пустой БД 15](#_Toc60686281)

[Отображение данных из выбранной таблицы 15](#_Toc60686282)

[Удаление данных из БД 16](#_Toc60686283)

[Изменение данных после их редактирования в DataGridView 16](#_Toc60686284)

[Завершение работы 18](#_Toc60686285)

[Демонтрация работы 18](#_Toc60686286)

[Запуск программы 18](#_Toc60686287)

[Главное окно 19](#_Toc60686288)

[Выбор таблицы в списке 19](#_Toc60686289)

[Отправка запроса 20](#_Toc60686290)

[Добавление новых записей 21](#_Toc60686291)

[Удаление имеющихся записей 21](#_Toc60686292)

[Изменение данных через DataGridView 22](#_Toc60686293)

[Заключение 22](#_Toc60686294)

[Список использованных источников 24](#_Toc60686295)

[Книги 24](#_Toc60686296)

[Интернет-ресурсы 24](#_Toc60686297)

[Интернет-ресурсы (образовательные YouTube-каналы от .NET Senior-разработчиков) 24](#_Toc60686298)

[Приложения 25](#_Toc60686299)

[Схема БД ITBeacon 25](#_Toc60686300)

[MainForm.cs 25](#_Toc60686301)

[SendRequestForm.cs 32](#_Toc60686302)

[ConnectionForm.cs 33](#_Toc60686303)

[Sql-запрос для создания таблиц в пустой БД 34](#_Toc60686304)

# Введение

## Цели и задачи

Освоить механизм слияния двух произвольных таблиц, хранящихся в XML-файле, различными способами.

## Объект и предмет исследования

Предметом исследования, в данном случае, является проблема автоматизации слияния таблиц, хранящихся в XML-формате.

Объектом исследования является база данных о недвижимости, требующей слияния двух её таблиц, которые записаны в XML-файле.

## Актуальность и область применения

В ходе разработки приложений с некоторой исходной базой информации по мимо привычного всем способа хранения данных в специализированных базах может возникнуть необходимость хранить их локально, в виде XML-документов для удобного и быстрого доступа к часто используемой неизменяемой информации, однако в таких ситуациях возникает проблема при попытке получить результат слияния двух подобных таблиц.

Проект представляет удобный интуитивный интерфейс для решения данной проблемы, построенный с использованием всех современных принципов построения программных продуктов.

Поскольку проект смоделирован таким образом, что может быть использован для работы с любой sql-бд, любым количеством таблиц и полями, сфера применения данного проекта по истине огромна: от банального списка дел или маленького интернет-ресурса, до крупных онлайн-проектов как в сфере продаж, так и в игровой индустрии.

Следовательно, при возникновении подобных целей, а также для экономии времени при ручном выборе столбцов для слияния в классическом варианте обращения к базе, этот проект поможет сэкономить время и предоставить готовый результат автоматически.

## Обзор литературных источников

Для достижения поставленных целей и задач было принято решение использовать следующие источники:

* Официальная документация Microsoft для освоения отдельных команд и технологий, изучения всех параметров, нюансов применения, исключительных ситуаций и т.п. т.к. это надёжный источник от разработчиков использованных технологий (языка программирования C# и использованного в примере MS SQL Server'а)
* Информационный портал metanit использовался для получения основных сведений о часто используемых командах в проекте.
* Книга Албахари Джозефа служила основным источником знаний о языке программирования C#.
* Информацию об sql как о языке, черпал из официальной документации PostgreSQL т.к. она написана доступнее остальных.
* YouTube-каналы XpucT, Byte++ и другие, служили дополнительным источником информации по реализации взаимодействия двух основных средств написания проекта.
* YouTube-канал SimpleCode служил основным помощником в решении проблем связанных с пониманием основных аспектов написания программной части проекта.
* YouTube-канал Роман Сакутин использовался для написания чистого кода, увеличения показателей быстродействия кода.

## Обзор используемой аппаратуры, инструментов

### MS SQL Server

MS SQL Server характеризуется такими особенностями как:

* Производительность. SQL Server работает очень быстро.
* Надежность и безопасность. SQL Server предоставляет шифрование данных.
* Простота. С данной СУБД относительно легко работать и вести администрирование.

### C#

Преимущества C#:

* C# – это объектно-ориентированный, простой и в то же время мощный язык программирования, который позволяет разработчикам создавать многофункциональные приложения.
* C# относиться к языкам компилируемого типа, поэтому он обладает всеми преимуществами таких языков.
* C# объединяет лучшие идеи современных языков программирования Java, C++, Visual Basic и т.д.
* Из-за большого разнообразия синтаксических конструкций и возможности работать с платформой .Net, C# позволяет быстрее, чем любой другой язык, разрабатывать программные решения.
* C# отличается надежностью и элегантностью

Примечание:

* Элегантностью C# отличается из-за большого разнообразия синтаксических конструкций.
* Большая надёжность была достигнута из-за работы CLR машины, ведь в отличии от других компилятор CLR запускает разработанное приложение на виртуальном процессоре. Поэтому в случаях возникновения каких-либо ошибок, это никак не повлияет на работу других программ в системе, но это также обозначает, что для запуска программы необходимо дополнительное время. Соответственно программы написанные на языке программирования C# более надёжные, но менее быстрые (нежели те же программы написанные на C++).

### Winforms

В отличии от своего прямого конкурента, WPF, выбор пал на WinForms по ряду причин:

Преимущества WinForms

* Технология более старая и, соответственно, лучше испытанная и протестированная
* На данный момент существует огромное множество готовых элементов управления, которые можно купить либо использовать бесплатно
* С точки зрения написания, дизайнер Visual Studio лучше приспособлен к WinForms, так как в WPF больше необходимо делать самому
* Прост в изучении
* Лично мной она изучена лучше, и, соответственно, качество, в таком случае, будет выше.

# Ход работы

## Теоретический блок. Термины. Сокращения

### C#

C# (произносится си шарп) — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота[7] как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в особенности, Java — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественная реализация интерфейсов).

### MS SQL

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире. Данная СУБД подходит для самых различных проектов: от небольших приложений до больших высоконагруженных проектов.

SQL Server был создан компанией Microsoft. Первая версия вышла в 1987 году. А текущей версией является версия 16, которая вышла в 2016 году и которая будет использоваться в текущем руководстве.

SQL Server долгое время был исключительно системой управления базами данных для Windows, однако начиная с версии 16 эта система доступна и на Linux.

Центральным аспектом в MS SQL Server, как и в любой СУБД, является база данных. **База данных** представляет хранилище данных, организованных определенным способом. Нередко физически база данных представляет файл на жестком диске, хотя такое соответствие необязательно. Для хранения и администрирования баз данных применяются системы управления базами данных (database management system) или СУБД (DBMS). И как раз MS SQL Server является одной из такой СУБД.

Для организации баз данных MS SQL Server использует реляционную модель. Эта модель баз данных была разработана еще в 1970 году Эдгаром Коддом. А на сегодняшний день она фактически является стандартом для организации баз данных.

Реляционная модель предполагает хранение данных в виде таблиц, каждая из которых состоит из строк и столбцов. Каждая строка хранит отдельный объект, а в столбцах размещаются атрибуты этого объекта.

Для идентификации каждой строки в рамках таблицы применяется первичный ключ (primary key). В качестве первичного ключа может выступать один или несколько столбцов. Используя первичный ключ, мы можем ссылаться на определенную строку в таблице. Соответственно две строки не могут иметь один и тот же первичный ключ.

Через ключи одна таблица может быть связана с другой, то есть между двумя таблицами могут быть организованы связи. А сама таблица может быть представлена в виде отношения ("relation").

Для взаимодействия с базой данных применяется язык SQL (Structured Query Language). Клиент (например, внешняя программа) отправляет запрос на языке SQL посредством специального API. СУБД должным образом интерпретирует и выполняет запрос, а затем посылает клиенту результат выполнения.

Изначально язык SQL был разработан в компании IBM для системы баз данных, которая называлась System/R. При этом сам язык назывался SEQUEL (Structured English Query Language). Хотя в итоге ни база данных, ни сам язык не были впоследствии официально опубликованы, по традиции сам термин SQL нередко произносят как "сиквел".

В 1979 году компания Relational Software Inc. разработала первую систему управления баз данных, которая называлась Oracle и которая использовала язык SQL. В связи с успехом данного продукта компания была переименована в Oracle.

Впоследствии стали появляться другие системы баз данных, которые использовали SQL. В итоге в 1989 году Американский Национальный Институт Стандартов (ANSI) кодифицировал язык и опубликовал его первый стандарт. После этого стандарт периодически обновлялся и дополнялся. Последнее его обновление состоялось в 2011 году. Но несмотря на наличие стандарта нередко производители СУБД используют свои собственные реализации языка SQL, которые немного отличаются друг от друга.

Выделяются две разновидности языка SQL: PL-SQL и T-SQL. PL-SQL используется в таких СУБД как Oracle и MySQL. T-SQL (Transact-SQL) применяется в SQL Server. Собственно поэтому в рамках текущего руководства будет рассматриваться именно T-SQL.

В зависимости от задачи, которую выполняет команда T-SQL, он может принадлежать к одному из следующих типов:

* **DDL** (Data Definition Language / Язык определения данных). К этому типу относятся различные команды, которые создают базу данных, таблицы, индексы, хранимые процедуры и т.д. В общем определяют данные.

В частности, к этому типу мы можем отнести следующие команды:

* + **CREATE**: создает объекты базы данных (саму базу даных, таблицы, индексы и т.д.)
  + **ALTER**: изменяет объекты базы данных
  + **DROP**: удаляет объекты базы данных
  + **TRUNCATE**: удаляет все данные из таблиц
* **DML** (Data Manipulation Language / Язык манипуляции данными). К этому типу относят команды на выбору данных, их обновление, добавление, удаление - в общем все те команды, с помощью которыми мы можем управлять данными.

К этому типу относятся следующие команды:

* + **SELECT**: извлекает данные из БД
  + **UPDATE**: обновляет данные
  + **INSERT**: добавляет новые данные
  + **DELETE**: удаляет данные
* **DCL** (Data Control Language / Язык управления доступа к данным). К этому типу относят команды, которые управляют правами по доступу к данным. В частности, это следующие команды:
  + **GRANT**: предоставляет права для доступа к данным
  + **REVOKE**: отзывает права на доступ к данным

### WinForms

Windows Forms — интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде. Причём управляемый код — классы, реализующие API для Windows Forms, не зависят от языка разработки. То есть программист одинаково может использовать Windows Forms как при написании ПО на C#, С++, так и на VB.Net, J# и др.

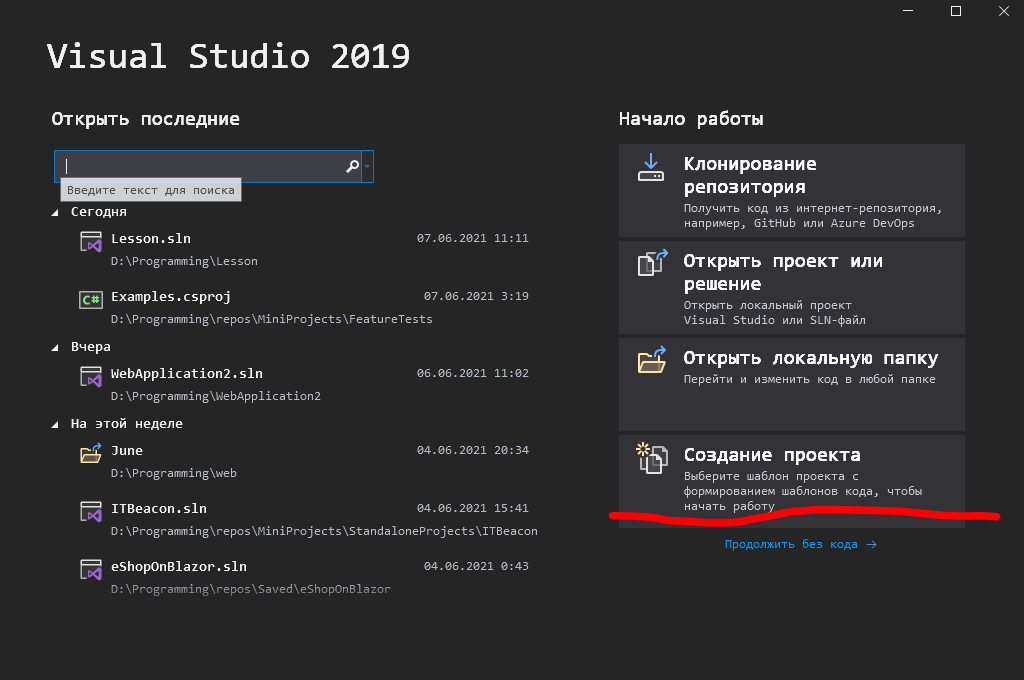
### ADO.NET

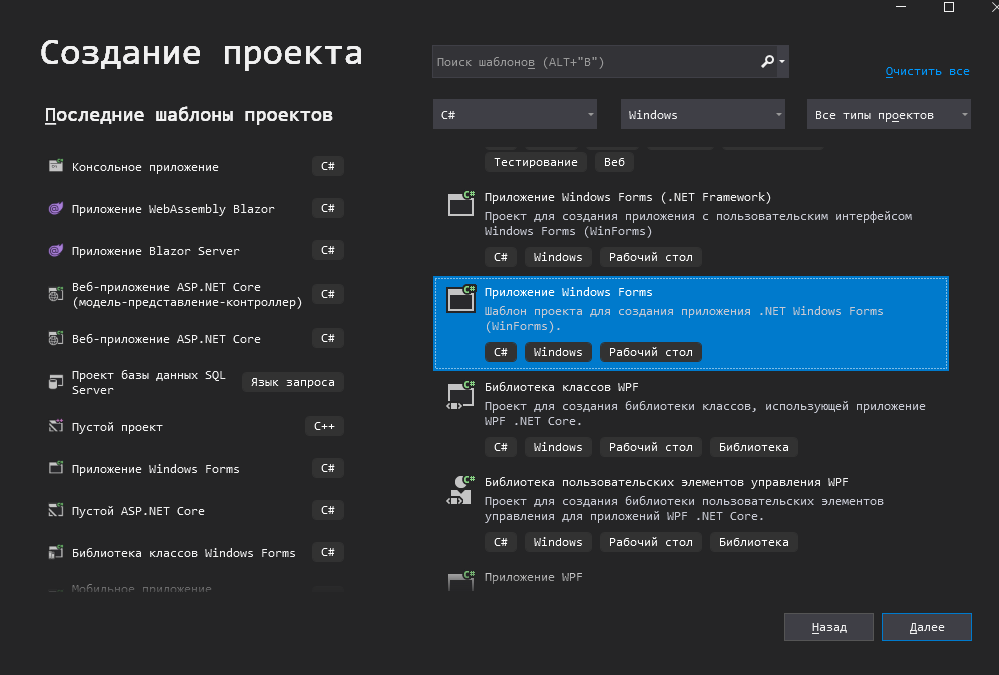
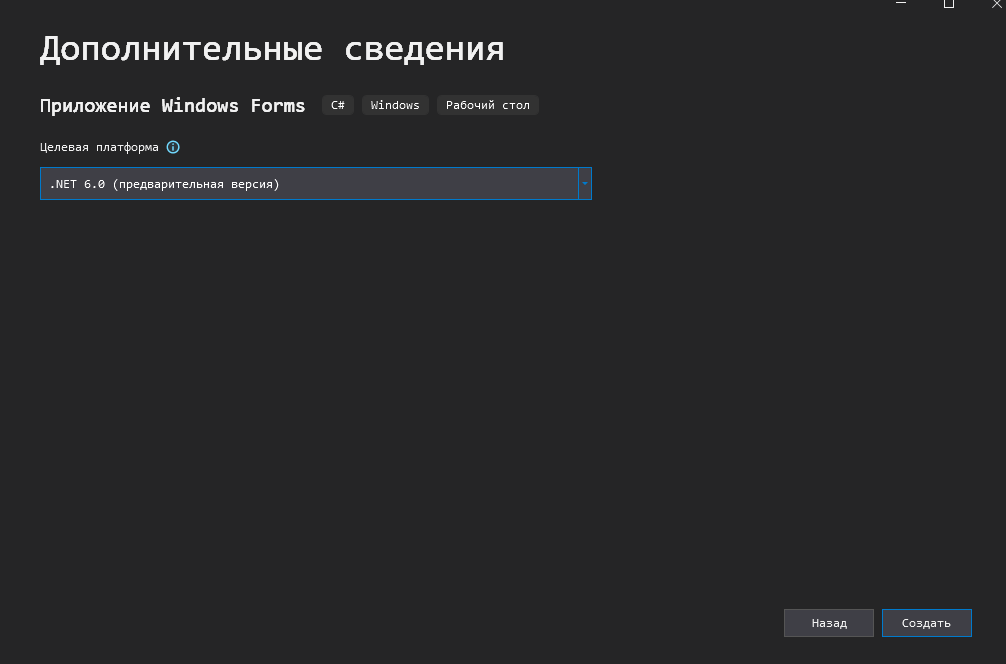
ADO.NET (ActiveX Data Object для .NET) — технология, предоставляющая доступ и управление данными, хранящимся в базе данных или других источниках (Microsoft SQL Server, Microsoft Access, Microsoft Excel, Microsoft Outlook, Microsoft Exchange, Oracle, OLE DB, ODBC, XML, текстовые файлы)[1], основанных на платформе .NET Framework и входящая в состав .NET Framework 2.0, представляет собой набор библиотек[2]. В отличие от технологии ADO, которая была в основном предназначена для тесно связанных клиент-серверных систем, ADO.NET больше нацелена на автономную работу с помощью объектов DataSet. Объекты DataSet представляют локальные копии взаимосвязанных таблиц данных, каждая из которых содержит набор строк и столбцов. Объекты DataSet позволяют вызывающей сборке (наподобие веб-страницы или программы, выполняющейся на настольном компьютере) работать с содержимым DataSet, изменять его, не требуя подключения к источнику данных, и отправлять обратно блоки измененных данных для обработки с помощью соответствующего адаптера данных. Но, пожалуй, самое фундаментальное различие между классической ADO и ADO.NET состоит в том, что ADO.NET является управляемой кодовой библиотекой, и, значит, подчиняется тем же правилам, что и любая управляемая библиотека. Типы, составляющие ADO.NET, используют протокол управления памятью CLR, принадлежат к той же системе типов (классы, интерфейсы, перечисления, структуры и делегаты), и доступ к ним возможен с помощью любого языка .NET. Классы ADO.NET находятся в сборке System.Data.dll.

## Процесс разработки

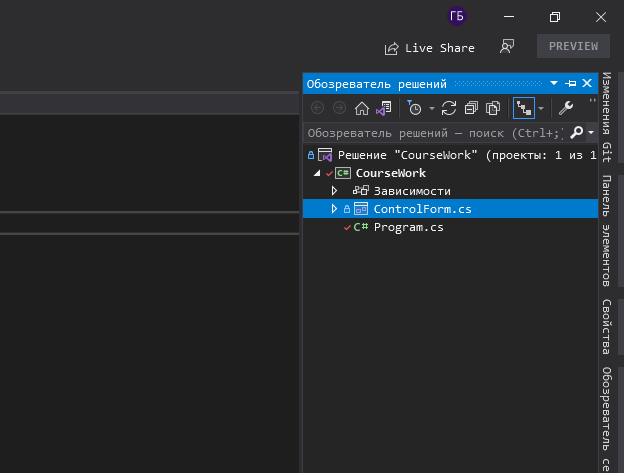
Сперва необходимо подготовить рабочую область, настроить инструменты, создать проект для разработки.

В Visual Studio 2019 Professional Preview создаём WinForms приложение

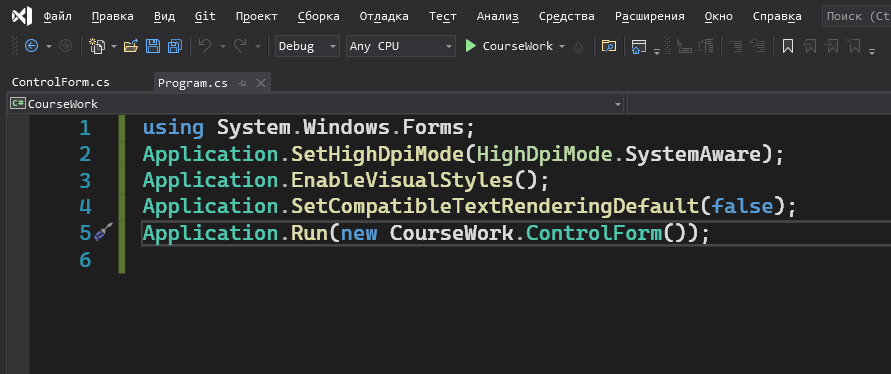


Структура проекта остаётся минимальной. Достатончо лишь переименовать форму:



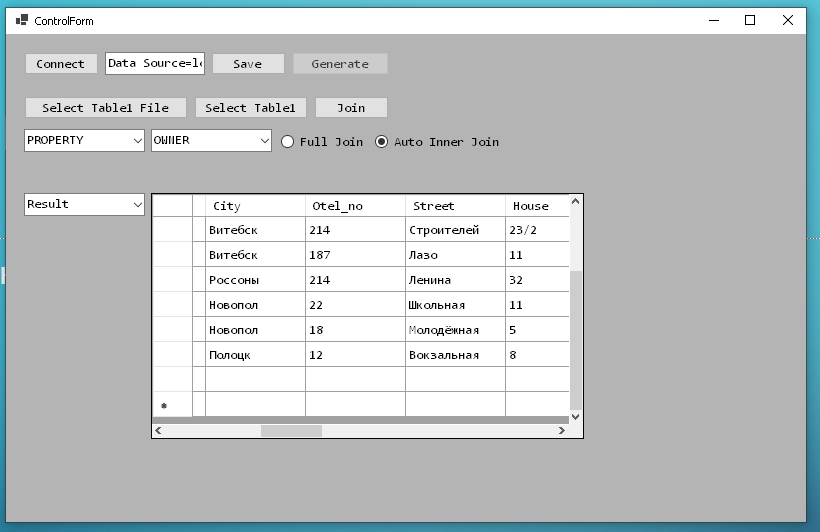
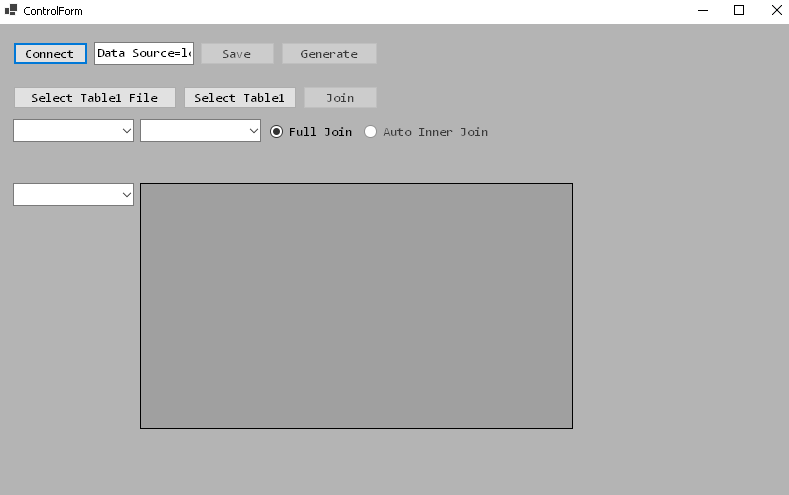
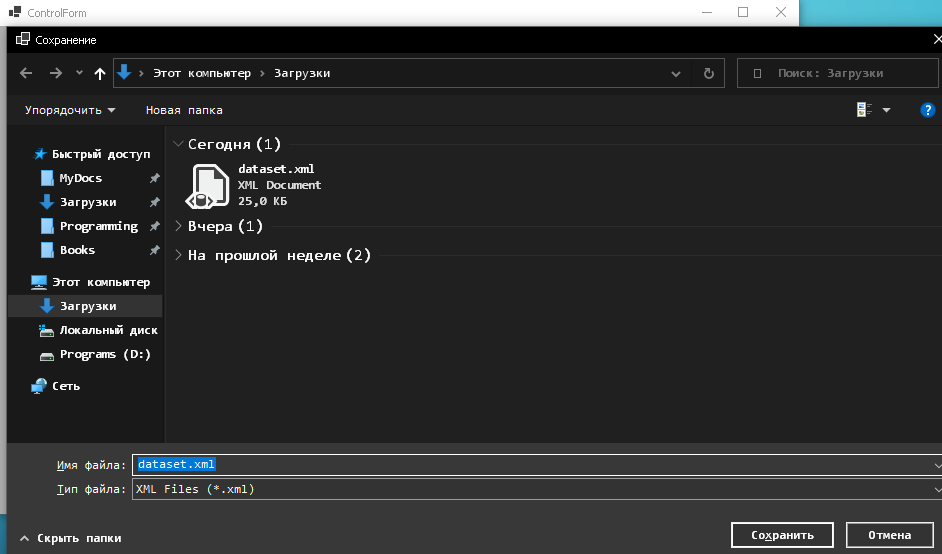
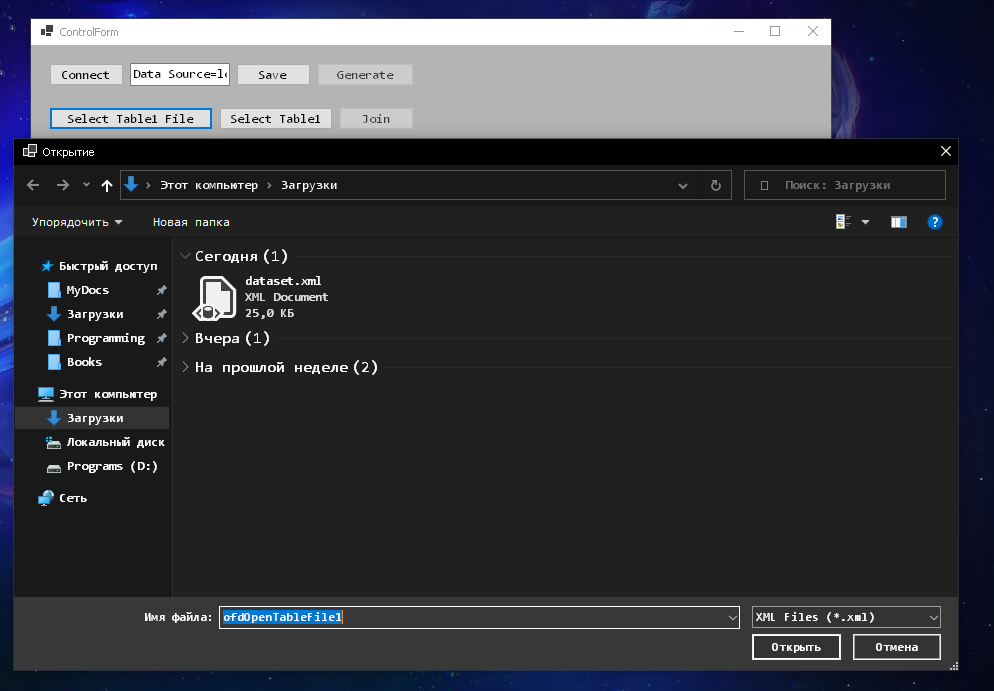
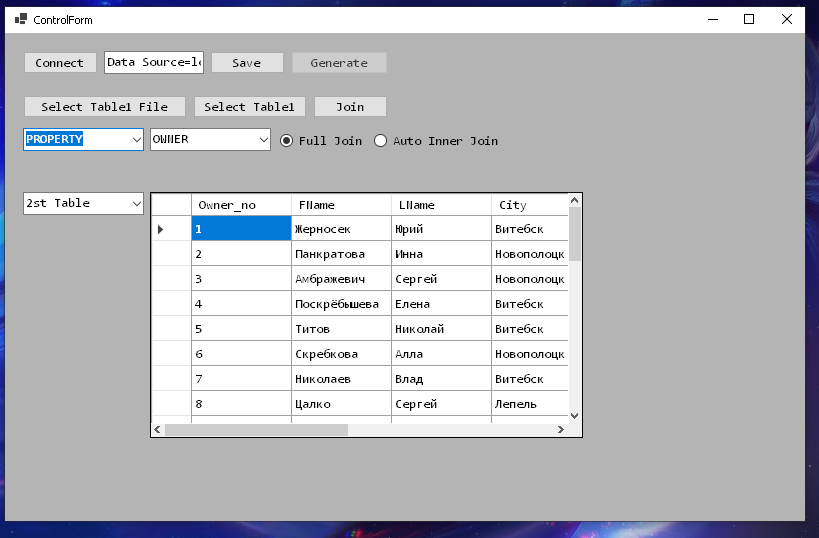
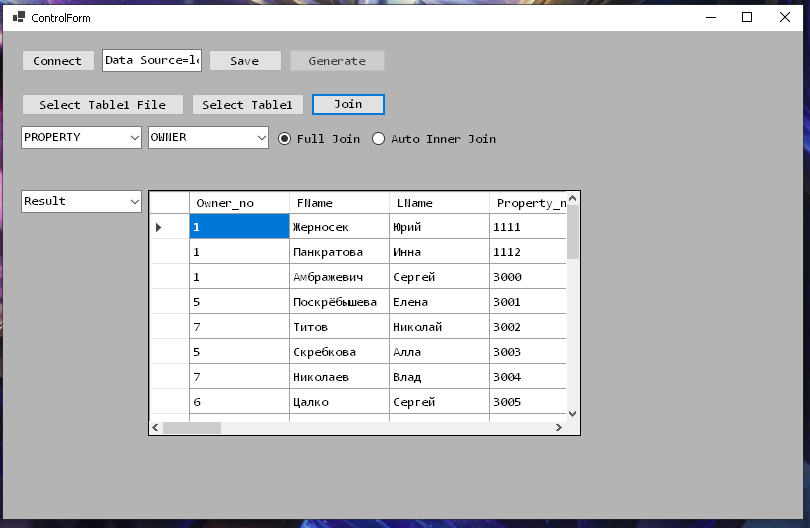
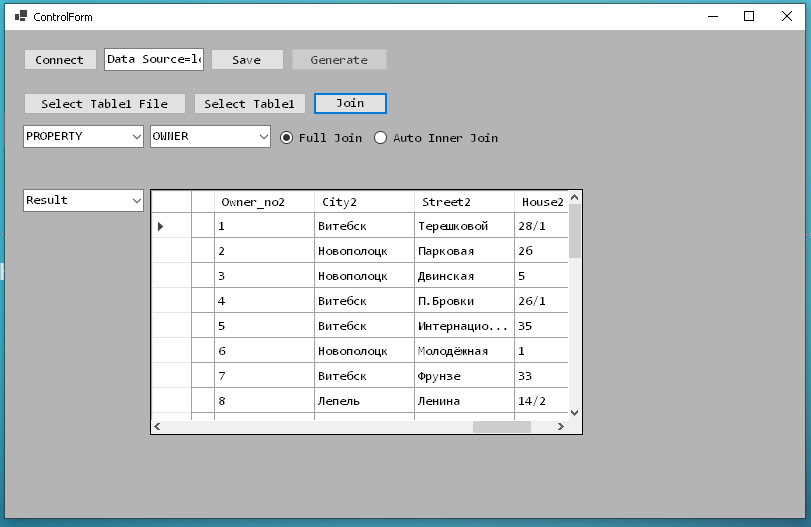
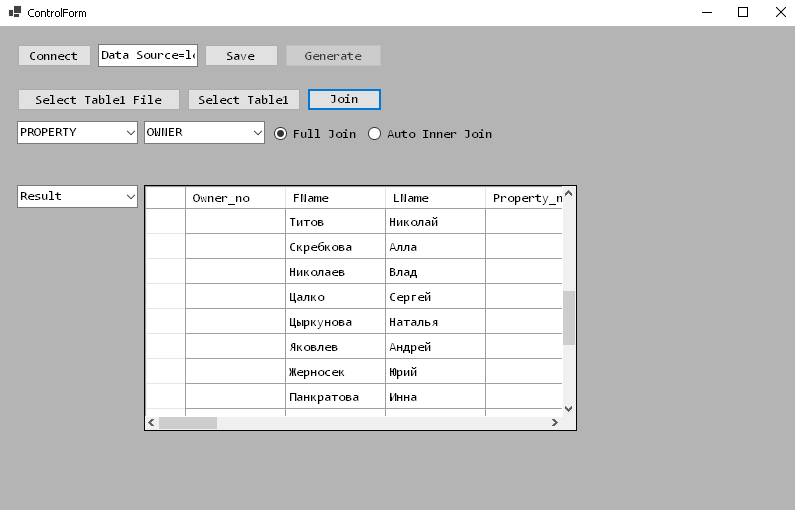
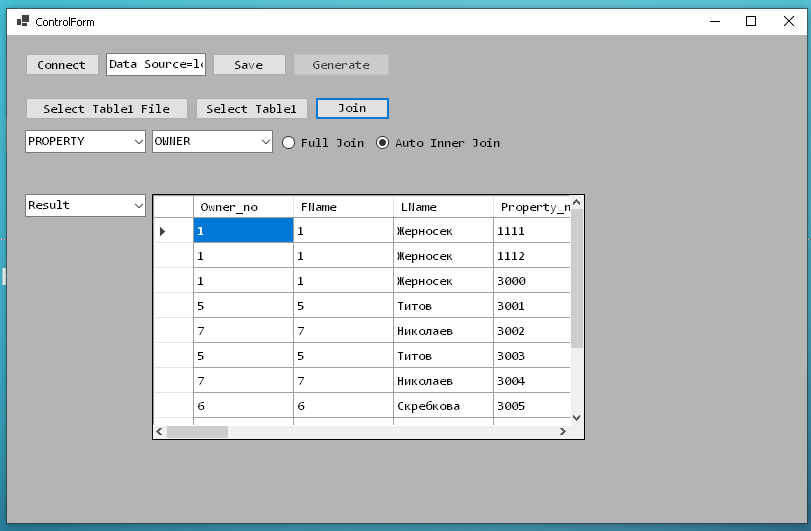
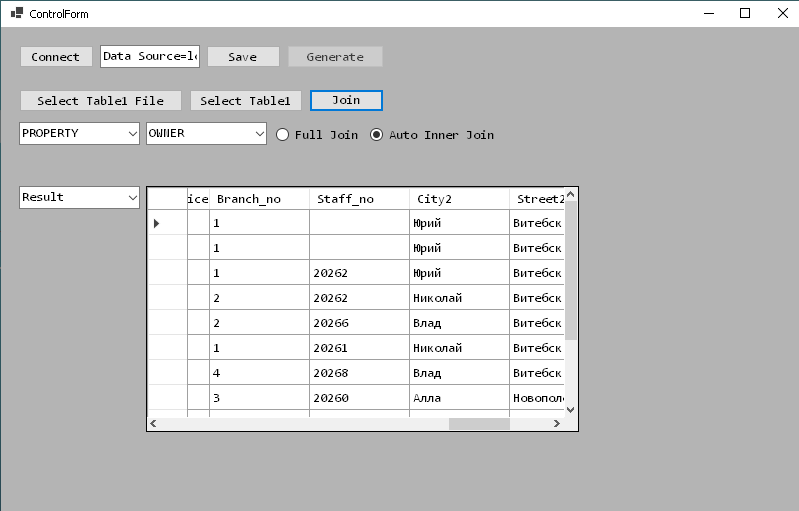
После чего переходим в Program.cs для очистки «лишнего кода» и получаем следующий вид:



Вся последующая работа будет проходить лишь в форме.

Сперва разместим все элементы управления на форму:

## Демонстрация работы



# Заключение

Таким образом, все поставленные цели и задачи были достигнуты:

Были изучены способы автоматического слияния таблиц баз данных, дополнены знания об XML-сериализации, изучены способы применения данных из баз в пользовательских приложениях, углублены знания работы с бд и языками программирования.

Приложение полностью соответствует ожидаемому функционалу, однако может быть дополнена в дальнейшем.

К примеру, в приложение планируется в дальнейшем внедрить функционал ручного выбора столбцов для симулирования всем привычного “Join on”.

# Список использованных источников

## Книги

1. Албахари, Джозеф, Албахари, Бен. «C# 8.0. Карманный справочник» Диалектика, 2020 год, 240 стр., ISBN 978-5-907203-14-3.
2. C# 7.0. Справочник. Полное описание языка, 7-е изд. Автор: Джозеф Албахари, Бен Албахари Издательство: Вильямс Год: 2018 Страниц: 1026

## Интернет-ресурсы

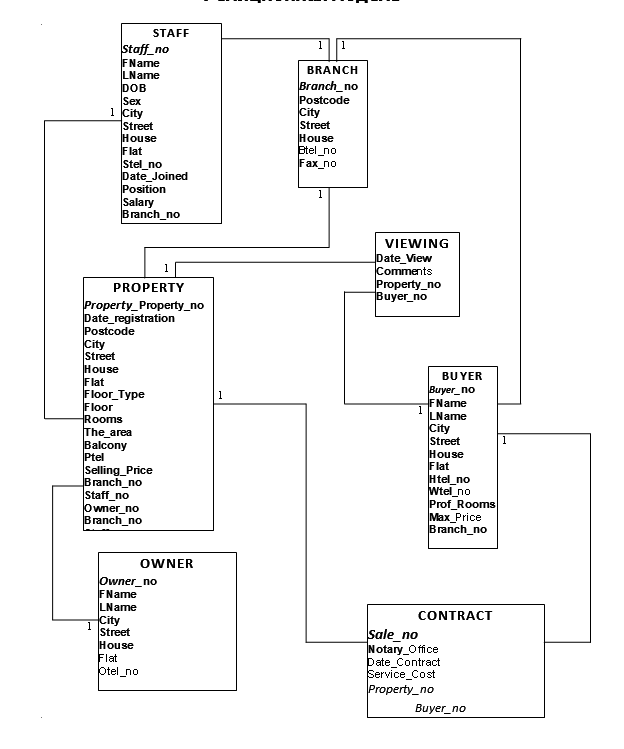
1. <https://docs.microsoft.com/en-us/dotnet/csharp/>
2. <https://metanit.com/sharp/>
3. <https://ru.wikipedia.org/wiki/C_Sharp>
4. <https://postgrespro.ru/docs/postgresql/>

## Интернет-ресурсы (образовательные YouTube-каналы от .NET Senior-разработчиков)

1. <https://www.youtube.com/channel/UC2CiWFIOjQix4E6WrARzDZg>
2. <https://www.youtube.com/channel/UCG7GW-X1cczyzLswoYTTnjQ>
3. <https://www.youtube.com/channel/UCtLKO1Cb2GVNrbU7Fi0pM0w>
4. <https://www.youtube.com/user/OfficialChannelGD>

# Приложения

## Схема БД Недвижимость



## Программная часть: