**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Владимирский государственный университет**

**имени Александра Григорьевича и Николая Григорьевича Столетовых»**

**(ВлГУ)**

**Колледж инновационных технологий и предпринимательства**

кафедра физики и прикладной математики

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

**ПО МОДУЛЮ ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем»**

студентки Казанкиной Татьяны Сергеевны

специальности 09.02.07 Информационные системы и программирование, квалификация Программист

группы ИПсп-122

Время прохождения практики

с «05» мая 2025 г.

по «25» мая 2025 г.

*Руководитель от ВлГУ*:

Преподаватель КИТП *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*М.Ю. Кабанова

Владимир, 2025

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ |
|  | Зав. кафедрой /К.С. Хорьков/ |
|  | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 г. |

З А Д А Н И Е

на *учебную практику по модулю ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем»* студентки 3 курса группы **ИПсп-122**

***Казанкиной Татьяны Сергеевны***

Специальность: 09.02.07 Информационные системы и программирование (программист)

Предприятие: Кафедра ФиПМ ВлГУ г. Владимир

***Последовательность прохождение практики:***

1. получить задание;
2. пройти инструктаж по технике безопасности и охране труда;
3. выполнить все задания;
4. оформить отчет согласно требованиям.

***За время прохождения практики необходимо:***

**1)** Разработайте код программного продукта в соответствии с отраслевыми стандартами.

При открытии формы для редактирования все поля выбранного объекта должны быть подгружены в соответствующие поля из базы данных. После редактирования/добавления в БД данные в окне/форме системы должны быть обновлены.

Соблюдайте культуру кодирования.

1. *Название приложения*. Используйте соответствующие названия для ваших приложений и файлов. Так, например, наименование настольного приложения должно обязательно включать название компании- заказчика. Используйте пиктограмму компании в качестве пиктограмм всех форм.
2. *Файловая структура.* Файловая структура проекта должна отражать логику, заложенную в приложение. Например, все формы содержатся в одной директории, пользовательские визуальные компоненты – в другой, классы сущностей – в третьей.
3. *Структура проекта*. Каждая сущность должна быть представлена в программе как минимум одним отдельным классом. Классы должны быть небольшими, понятными и выполнять одну единственную функцию (Single responsibility principle). Для работы с разными сущностями используйте разные формы, где это уместно.
4. *Макет и технические характеристики*. Все компоненты системы должны иметь единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

– разметка и дизайн (предпочтение отдается масштабируемой компоновке;

– должно присутствовать ограничение на минимальный размер окна;

– должна присутствовать возможность изменения размеров окна, где это необходимо;

– увеличение размеров окна должно увеличивать размер контентной части, например, таблицы с данными из БД);

– группировка элементов (в логические категории);

– использование соответствующих элементов управления (например, выпадающих списков для отображения подстановочных значений из базы данных);

– расположение и выравнивание элементов (метки, поля для ввода и т.д.);

– последовательный переход фокуса по элементам интерфейса (по нажатию клавиши TAB);

– общая компоновка логична, понятна и проста в использовании;

– последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами в приложении (в том числе обратно, например, с помощью кнопки «Назад»);

– соответствующий заголовок на каждом окне приложения (не должно быть значений по умолчанию типа MainWindow, Form1 и т.п.).

1. *Руководство по стилю*. Все экранные формы пользовательского интерфейса должны иметь заголовок с логотипом (для форм – квадратный вариант, для отчетов – прямоугольный вариант) и название (за исключением простых диалогов).

Все визуальные компоненты должны быть выровнены, доступны, иметь соизмеримый масштаб и не оставлять много свободного пространства. Не допускайте орфографические и грамматические ошибки.

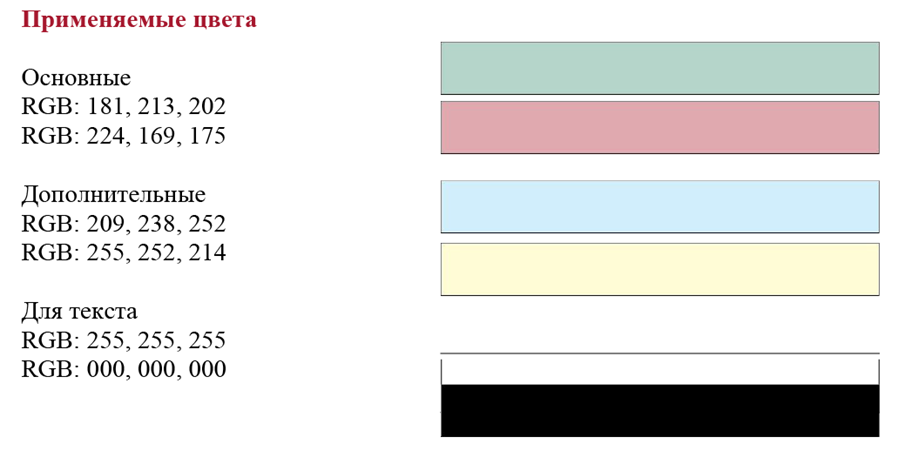
В таблицах рекомендуется применять дополнительные цвета для разного оформления четных и нечетных строк.

Для надписей используйте черный цвет, или белый (в том случае, если фон – темный); начертание: обычный, курсив или полужирный; размер: 11 – 18 пт.

Все элементы интерфейса должны быть логически сгруппированы вместе, чтобы система была более простой в использовании.

Во всей системе должны применяться одинаковые макетные решения.

К внешнему виду относится любой визуальный элемент, с которым контактирует конечный пользователь, включая экранные формы, отчеты, надписи и т.д.



1. *Обратная связь с пользователем*. Уведомляйте пользователя о совершаемых им ошибках или о запрещенных в рамках задания действиях, запрашивайте подтверждение перед удалением, предупреждайте о неотвратимых операциях, информируйте об отсутствии результатов поиска и т.п. Окна сообщений соответствующих типов (например, ошибка, предупреждение, информация) должны отображаться с соответствующим заголовком и пиктограммой. Текст сообщения должен быть полезным и информативным, содержать полную информацию о совершенных ошибках пользователя и порядок действий для их исправления. Также можно использовать визуальные подсказки для пользователя при вводе данных.
2. *Обработка ошибок*. Не позволяйте пользователю вводить некорректные значения в текстовые поля сущностей. Например, в случае несоответствия типа данных или размера поля введенному значению. Оповестите пользователя о совершенной им ошибке. При возникновении непредвиденной ошибки приложение не должно аварийно завершать работу.

Запрещено удаление сущностей, которое приведет к нарушению ограничений связей. При удалении любых сущностей необходимо либо спрашивать подтверждение пользователя, либо реализовать возможность отмены операции удаления.

1. *Оформление кода*. Идентификаторы переменных, методов и классов должны отражать суть и/или цель их использования, в том числе и наименования элементов управления (например, не должно быть значений по умолчанию типа Form1, button3). Идентификаторы должны соответствовать соглашению об именовании (Code Convention) и стилю CamelCase (для C# и Java) и snake\_case (для Python). Допустимо использование не более одной команды в строке. Максимальная длина строки 80 символов. Отступ составляет четыре пробела (без tab).

*Реализуйте переиспользуемые визуальные компоненты*. Не дублируйте логику – это отнимет много времени. Например, понадобится текстовое поле с валидацией целочисленного значения много более чем в одном месте. Другой пример, с каждой сущностью нужно осуществлять четыре одинаковых действия: чтение, обновление, создание и удаление. Где уместно используйте наследование и полиморфизм.

1. *Комментарии*. Используйте комментарии для пояснения неочевидных фрагментов кода. Запрещено комментирование кода. Хороший код воспринимается как обычный текст. Не используйте комментарии для пояснения очевидных действий. Комментарии должны присутствовать только в местах, которые требуют дополнительного пояснения. Используйте тип комментариев, который в дальнейшем позволит сгенерировать XML- документацию, с соответствующими тегами (например, param, return(s), summary и др.)
2. Создайте отчёт в формате **.xls**.

**2)** Определить наборы входных данных и выполнить функциональное тестирование модуля по определенному сценарию. Провести тестирование для проверки функциональности программы (хотя бы один тест на одну функцию). Использовать инструментальные средства для тестирования. Представить результаты тестирования в виде протокола тестирования, в соответствии со стандартами.

*Создайте форму для авторизации зарегистрированных пользователей*. Пользователи будут входить в систему, используя эту форму. Они будут вводить свой логин и пароль. При попытке входа введенные данные проверяются на совпадение в базе данных. Должно выдаваться сообщение об ошибке в случае неправильного ввода связки логин/пароль. После успешного входа пользователя в зависимости от их роли происходит перенаправление в соответствующие разделы. Когда пользователь нажимает на кнопку «Выход» в любом окне/форме системы, пользователи должны быть отключены от системы и совершен переход на *Форму авторизации*.

*Необходимо предусмотреть возможность регистрации*. При регистрации пароль должен отвечать следующим требованиям:

– Минимум 6 символов

– Минимум 1 прописная буква

– Минимум 1 цифра

– Минимум один символ из набора: ! @ # $ % ^.

**3)** Результаты работы предоставить в системе контроля версий.

**4)** Задание по стандартизации: оформить отчет по результатам практики согласно требованиям, к оформлению документации.

**5)** Задание по охране труда, технике безопасности и охране окружающей среды: изучить правила техники безопасности при работе за компьютером, вопросы охраны труда на месте прохождения практики.

Отчет по практике составить к **23.05.2025**

Задание выдал: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.Ю. Кабанова

Задание получил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.С. Казанкина

05.05.2025

СОДЕРЖАНИЕ

[Введение 7](#_Toc165156075)

[1 Разработка приложения 8](#_Toc165156076)

[1.1 Описание предметной области 8](#_Toc165156077)

[2 Подключение к базе данных 13](#_Toc165156079)

[3 Разработка приложения 19](#_Toc165156080)

[3.1 Разработка пользовательского интерфейса 19](#_Toc165156081)

[3.2 Файловая структура. 19](#_Toc165156082)

[3.3 Структура проекта. 19](#_Toc165156083)

[3.4 Оформление кода. 19](#_Toc165156084)

[3.5 Создание отчета в формате .xls 19](#_Toc165156085)

[4 Система контроля версий 20](#_Toc165156086)

[5 Тестирование 21](#_Toc165156087)

[5.1 Тест-требования 21](#_Toc165156088)

[5.2 Тест-план 21](#_Toc165156089)

[5.3 Результаты тестирования 21](#_Toc165156090)

[Заключение 22](#_Toc165156091)

[Список использованных источников 23](#_Toc165156092)

[Приложение А 24](#_Toc165156093)

введение

Учебная практика УП.01.01 по модулю ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем» проходит с 05 мая 2025 года по 23 мая 2025 года.

Целью практики является разработка кода программного продукта в соответствии с отраслевыми стандартами. Процесс должен включать в себя описание предметной области, подключение к базе данных (далее БД), разработку пользовательского интерфейса и структуры проекта, тестирование готового продукта и создание для него документации.

Приложение предусматривает возможность регистрации и авторизации и реализует интерфейс в соответствии с руководством по стилю.

1. разработка приложения

1.1 Описание предметной области

Темой учебной практики является разработка приложения для управления арендой недвижимости.

Для наглядного представления предметной области была создана ER-диаграмма в нотации Мартина. Диаграмма в нотации Мартина представлена на Рисунке 1.

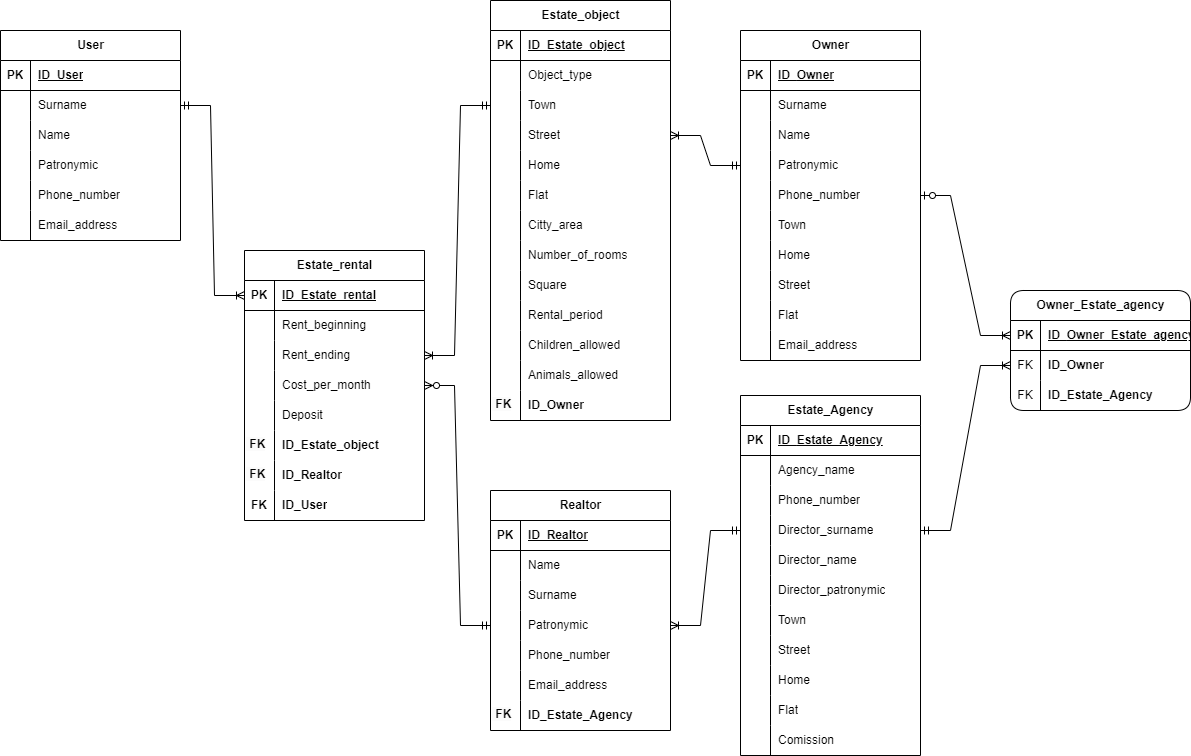


Рисунок 1

Словарь данных, отражающий структуру и содержимое базы данных представлен в Таблице 1.

Таблица 1 – Словарь данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User | | | | |
| Ключ | Поле | Тип | Обязательное (да/нет) | Примечание |
| PK | ID\_User | INT | Да | Первичный ключ таблицы User |
|  | Surname | NVARCHAR(30) | Да | Фамилия |
|  | Name | NVARCHAR(30) | Да | Имя |
|  | Patronymic | NVARCHAR(30) | Нет | Отчество |
|  | Phone\_number | CHAR(11) | Да | Номер телефона |
|  | Email\_address | VARCHAR(30) | Да | Адрес электронной почты |
| Estate\_rental | | | | |
| PK | ID\_Estate\_rental | INT | Да | Первичный ключ таблицы Estate\_rental |
|  | Rent\_beginning | DATE | Да | Начало аренды |
|  | Rent\_ending | DATE | Нет | Конец аренды |
|  | Cost\_per\_month | DECIMAL(10,2) | Да | Плата за месяц |
|  | Deposit | DECIMAL(10,2) | Да | Залог |
| FK | ID\_Estate\_object | INT | Да | Внешний ключ на таблицу Estate\_object |
| FK | ID\_Realtor | INT | Нет | Внешний ключ на таблицу Realtor |
| FK | ID\_User | INT | Да | Внешний ключ на таблицу User |
| Realtor | | | | |
| PK | ID\_Realtor | INT | Да | Первичный ключ таблицы Realtor |
|  | Surname | NVARCHAR(30) | Да | Фамилия |
|  | Name | NVARCHAR(30) | Да | Имя |
|  | Patronymic | NVARCHAR(30) | Нет | Отчество |
|  | Phone\_number | CHAR(11) | Да | Номер телефона |
|  | Email-address | VARCHAR(30) | Да | Адрес электронной почты |
| FK | ID\_Estate\_agency | INT | Да | Внешний ключ на таблицу Estate\_agency |

Продолжение таблицы

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Estate\_agency | | | | | | | |
| PK | ID\_Estate\_agency | | | INT | Да | | Первичный ключ таблицы Estate\_agency |
|  | Agency\_name | | | NVARCHAR(30) | Да | | Название агенстства |
|  | Phone\_number | | | CHAR(11) | Да | | Номер телефона |
|  | Director\_surname | | | NVARCHAR(30) | Да | | Фамилия директора |
|  | Director\_name | | | NVARCHAR(30) | Да | | Имя директора |
|  | Director\_patronymic | | | NVARCHAR(30) | Нет | | Отчество директора |
|  | Town | | | NVARCHAR(30) | Да | | Город |
|  | Street | | | NVARCHAR(30) | Да | | Улица |
|  | Home | | | NVARCHAR(30) | Да | | Дом |
|  | Flat | | | INT | Да | | Квартира |
|  | Comission | | | DECIMAL(2,1) | Да | | Комиссия |
| Owner | | | | | | | |
| PK | ID\_ Owner | | | INT | Да | | Первичный ключ таблицы Owner |
|  | Surname | | | NVARCHAR(30) | Да | | Фамилия |
|  | Name | | | NVARCHAR(30) | Да | | Имя |
|  | Patronymic | | | NVARCHAR(30) | Нет | | Отчество |
|  | Phone\_number | | | CHAR(11) | Да | | Номер телефона |
|  | Town | | | NVARCHAR(30) | Да | | Город |
|  | Street | | | NVARCHAR(30) | Да | | Улица |
|  | Home | | | NVARCHAR(30) | Да | | Дом |
|  | Flat | | | INT | Да | | Квартира |
|  | Email\_address | | | VARCHAR(30) | Да | | Адрес электронной почты |
| Estate\_object | | | | | | | |
| PK | | ID\_ Estate\_object | INT | | Да | Первичный ключ таблицы Estate\_object | |
|  | | Object\_type | NVARCHAR(20) | | Да | Тип объекта (дом, квартира, офим) | |

Продолжение таблицы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Town | NVARCHAR(30) | Да | Город |
|  | Street | NVARCHAR(30) | Да | Улица |
|  | Home | NVARCHAR(30) | Да | Дом |
|  | Flat | INT | Нет | Квартира |
|  | City\_area | NVARCHAR(30) | Да | Район города |
|  | Number\_of\_rooms | INT | Да | Количество комнат |
|  | Square | DECIMAL(10,2) | Да | Площадь |
|  | Rental\_period | INT | Да | Минимальный срок аренды (в дянх) |
| FK | ID\_Owner | INT | Да | Внешний ключ на таблицу Owner |
|  | Children\_allowed | BIT | Да | Дети разрешены |
|  | Animals\_allowed | BIT | Да | Животные разрешены |
| Owner\_Estate\_agency | | | | |
| PK | ID\_ Owner\_Estate\_agency | INT | Да | Первичный ключ таблицы Owner\_Estate\_agency |
|  | ID\_Owner | INT | Да | Внешний ключ на таблицу Owner |
|  | ID\_Estate\_agency | INT | Нет | Внешний ключ на таблицу Estate\_agency |

Для работы с приложением предполагается наличие следующих входных параметров:

* Пользователь: фамилия, имя, отчество, адрес электронной почты, номер телефона.
* Аренда недвижимости: начало аренды, конец аренды, стоимость за месяц, залог.
* Объект недвижимости: тип объекта, район, город, улица, дом, квартира, количество комнат, площадь, минимальный срок аренды, проживание с детьми, проживание с животными.
* Собственник: фамилия, имя, отчество, номер телефона, город, улица, дом, квартира.
* Агентство недвижимости: название, номер телефона, фамилия владельца, имя владельца, отчество владельца, город, улица, дом, квартира, комиссия.
* Риелтор: фамилия, имя, отчество, номер телефона, адрес электронной почты.

Также наличие следующих выходных параметров:

* Список пользователей.
* Список одобренных аренд.
* Список объектов недвижимости.
* Список собственников.
* Список агентств.
* Список риелторов.

Доступ к таблицам будет осуществляться в соответствии с ролями пользователей.

2 подключение к базе данных

Для подключения БД к проекту необходимо в верхнем меню перейти в раздел «Средства» и выбрать пункт «Подключение к базе данных». Выпадающий список «Средства» представлен на Рисунке 2.

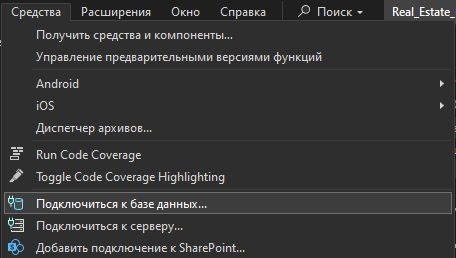


Рисунок 2 – Выпадающий список «Средства»

Откроется окно «Добавить подключение», в которое нужно ввести имя сервера и выбрать базу данных из списка. Окно «Добавить подключение» представлено на Рисунке 3.

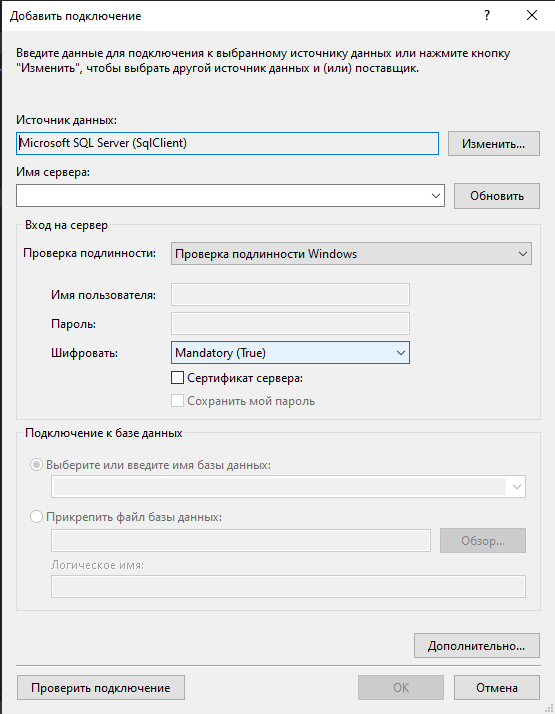


Рисунок 3 – Окно «Добавить подключение»

Для того, чтобы получить имя сервера, нужно открыть Microsoft SQL Server Manager Studio. При входе в приложение открывается окно «Соединение с сервером». Необходимо скопировать строчку «Имя сервера». Окно «Соединение с сервером» представлено на Рисунке 4.

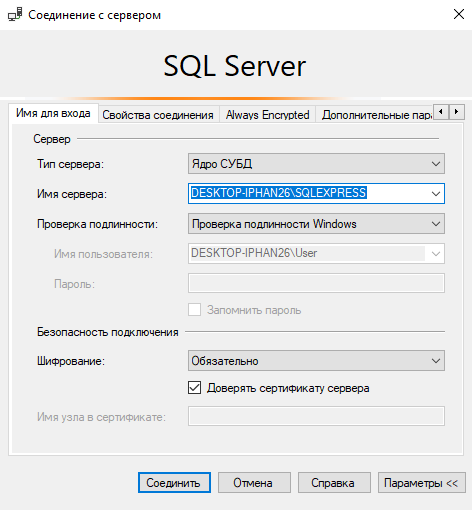


Рисунок 4 – Окно «Соединение с сервером»

Скопированную строчку вставляем в поле «Имя сервера» в окне «Добавить подключение» и выбираем нужную БД из раскрывающегося списка. Окно «Добавить подключение» с выбранной БД представлено на Рисунке 5.

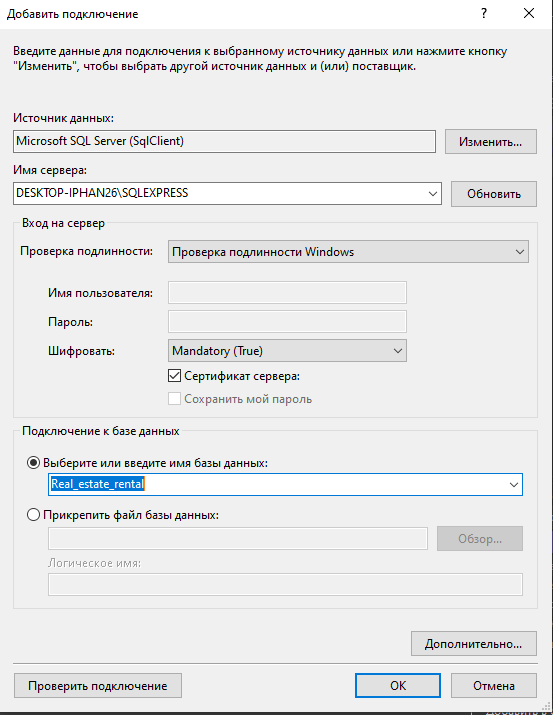


Рисунок 5 – Окно «Добавить подключение» с выбранной БД

Важно перейти в раздел «Дополнительно» и в открывшемся окне «Расширенные свойства» указать «Trust Server Certificate=true». Нажимаем «ОК». Окно «Расширенные свойства» представлено на Рисунке 6.

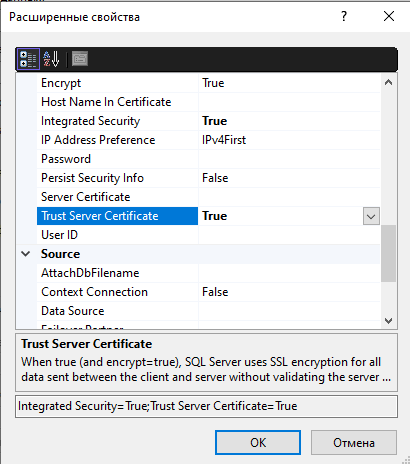


Рисунок 6 – Окно «Расширенные свойства»

Далее в окне «Добавить подключение нажимаем «ОК». Выбранная БД отобразится в активных подключениях. Активные подключения представлены на Рисунке 7.

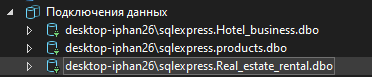


Рисунок 7 – Активные подключения

Для автоматического создания классов для каждой сущности нужно скопировать строку подключения и вставить в соответствующую команду. Для этого выбираем необходимое подключение, нажимаем правой кнопкой мыши и выбираем в раскрывшемся меню пункт «Свойства». Раскрывающееся меню представлено на Рисунке 8.

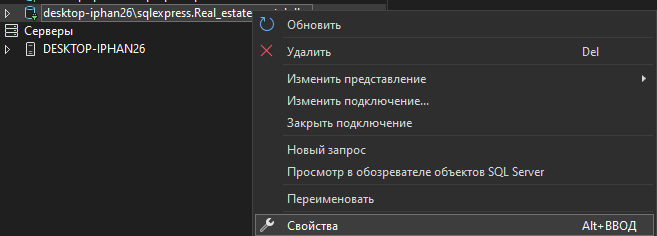


Рисунок 8 – Раскрывающееся меню

В окне «Свойства» находим пункт «Строка подключения» и копируем из него значение. Окно «Свойства» представлено на Рисунке 9.

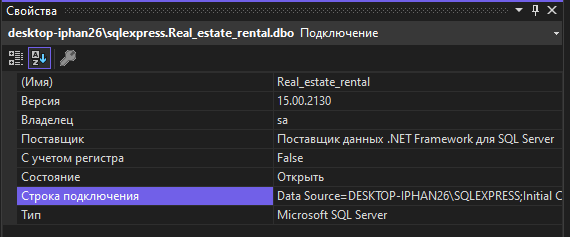


Рисунок 9 – Окно «Свойства»

Далее нужно установить необходимые для подключения пакеты. Команды для установки представлены в Листинге 2.1.

Листинг 2.1

Install-Package Microsoft.EntityFrameworkCore.SqlServer

Install-Package Microsoft.EntityFrameworkCore.Tools

Install-Package Microsoft.EntityFrameworkCore.Design

Команда для подключения к БД и автоматического создания классов моделей представлена в Листинге 2.2

Листинг 2.2

Scaffold-DbContext "Data Source=DESKTOP-IPHAN26\SQLEXPRESS;Initial Catalog=Real\_estate\_rental;Integrated Security=True;Trust Server Certificate=True" Microsoft.EntityFrameworkCore.SqlServer

Результат выполнения команды представлен на Рисунке 10.

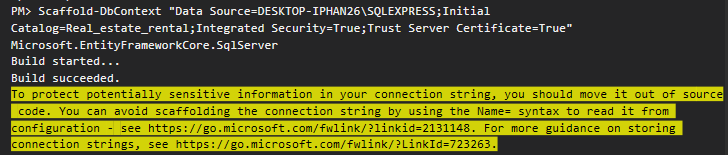


Рисунок 10 – Результат выполнения команды

Автоматически созданные классы и контекст подключения представлены на Рисунке 11.

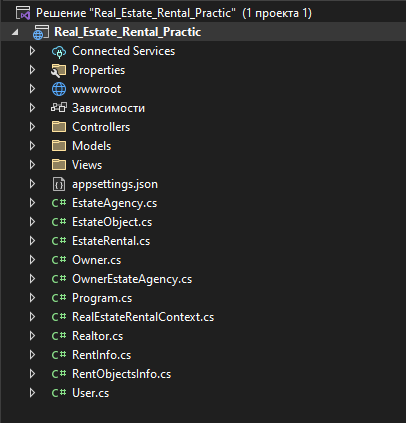


Рисунок 11 – Созданные классы и контекст подключения

Таким образом было осуществлено подключение к разработанной БД и автоматически созданы все необходимые для работы классы.

3 разработка приложения

3.1 Разработка пользовательского интерфейса

Входные и выходные значения, описание интерфейса со скриншотами

3.2 Файловая структура.

3.3 Структура проекта.

3.4 Оформление кода.

Реализация переиспользуемых визуальных компонентов.

3.5 Создание отчета в формате .xls

4 система контроля версий

Описание системы контроля версий

Скриншоты репозитория и работы с ним пошагово

5 тестирование

5.1 Тест-требования

Рассмотреть все возможные случаи для проверки

5.2 Тест-план

Тестовый пример привести для каждого пункта тест-требования

Описание теста, входные данные, ожидаемые выходные данные, сценарий

5.3 Результаты тестирования

Для каждого тестового примера указать входные и выходные данные и подтвердить скриншотом

заключение

В процессе прохождения учебной практики УП.01.01 по модулю ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем» были рассмотрены и реализованы …

список использованных источников

1)

приложение А

Указать ссылку на репозиторий

Код главной формы