СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc152690897)

[1. ЦЕЛЬ И ЗАДАЧИ УЧЕБНОЙ ПРАКТИКИ 4](#_Toc152690898)

[1.1 Основные требования по технике безопасности во время прохождения учебной практики 4](#_Toc152690899)

[1.2 Нормативно-правовые документы 5](#_Toc152690900)

[2. ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ПОСТАНОВКА ЗАДАЧИ 7](#_Toc152690901)

[2.1 Описание предметной области 7](#_Toc152690902)

[2.2 Описание групп пользователей информационной системы “Покупка и продажа квартир” 9](#_Toc152690903)

[2.3 Основные требования, предъявляемые к информационной системе “Покупка и продажа квартир” 10](#_Toc152690904)

[3. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ “ПОКУПКА И ПРОДАЖА КВАРТИР” 12](#_Toc152690905)

[3.1 Моделирование некоторых динамических аспектов системы 12](#_Toc152690906)

[3.1.1 Создание диаграммы вариантов использования 12](#_Toc152690907)

[3.1.2 Создание диаграммы деятельности 14](#_Toc152690908)

[3.2 Разработка функциональной структуры информационной системы. Карта навигации по информационной системе “Покупка и продажа квартир” 14](#_Toc152690909)

[3.3 Разработка макета информационной системы “Покупка и продажа квартир” 14](#_Toc152690910)

[4. РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ “ПОКУПКА И ПРОДАЖА КВАРТИР” 16](#_Toc152690911)

[4.1 Моделирование и разработка базы данных 16](#_Toc152690912)

[4.2 Разработка программных модулей информационной системы «Покупка и продажа квартир» 18](#_Toc152690913)

[5 РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ 43](#_Toc152690914)

[5.1 Назначение эксперимента. Выбор и обоснование методики проведения тестирования 43](#_Toc152690915)

[5.2 Технология тестирования 43](#_Toc152690916)

[5.3 Результаты проведения тестирования 45](#_Toc152690917)

[ЗАКЛЮЧЕНИЕ 48](#_Toc152690918)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 49](#_Toc152690919)

[ПРИЛОЖЕНИЕ А 50](#_Toc152690920)

[ПРИЛОЖЕНИЕ Б 67](#_Toc152690921)

# ВВЕДЕНИЕ

Организации по купле-продаже квартир играют важную роль в жизни человека. Они помогают людям приобрести собственное жилье, что является одной из базовых потребностей человека.

Организациям доступны обширные базы данных объектов недвижимости, что позволяет им быстро и эффективно находить подходящий вариант для покупателя.

Специалисты организаций по купле-продаже квартир могут объективно оценить стоимость объекта недвижимости, что позволяет покупателю не переплачивать.

Также они оказывают помощь в оформлении всех необходимых документов и проведении сделки, что позволяет покупателю сэкономить время и избежать возможных проблем.

В целом, организации по купле-продаже квартир позволяют людям сэкономить время, деньги и нервы, а также избежать возможных проблем при покупке квартиры.

Целью практики является проектирование и разработка прикладного программного обеспечения деятельности организации по купле-продаже.

Основные задачи учебной практики:

* анализ предметной области;
* развить готовность выполнять поставленные профессиональные задачи;
* разработка программных модулей¸ оптимизированных для работы;
* тестирование разработанных программных модулей на соответствие функциональным требованиям.

Интерфейс программы обеспечит удобство для клиентов, ускорит процедуры оформления договоров, а также предоставит возможность отслеживания статуса купли-продажи в реальном времени.

Программное обеспечение обеспечит надежное хранение и защиту конфиденциальной информации клиентов и транзакций, соблюдая все стандарты безопасности данных.

1. ЦЕЛЬ И ЗАДАЧИ УЧЕБНОЙ ПРАКТИКИ

Разработка программных модулей для системы покупки и продажи квартир направлена на повышение эффективности, надежности и прозрачности процессов купли-продажи. Целью является разработка и проектирование информационной системы, способствующей оптимизации работы и обеспечивающей высокий уровень безопасности и качества обслуживания клиентов.

Основные задачи:

* Автоматизация Учета Квартир и Проведения Договоров: Разработка модулей для автоматизации учета квартир и проведения договоров купли-продажи;
* Управление Договорами: Реализация функционала по управлению подписания договоров и мониторинга информации о специалистах, клиентах и их квартирах.
* Обеспечение Безопасности Данных: Разработка модулей для защиты конфиденциальности данных клиентов, агентов и квартир, чтобы предотвратить несанкционированный доступ.

Цель и задачи в совокупности направлены на повышение эффективности обслуживания и обеспечение безопасности и надежности подписания договоров между клиентами и организациями через современные информационные технологии.

* 1. Основные требования по технике безопасности во время прохождения учебной практики

К работе в компьютерном классе допускаются лица, ознакомленные с данной инструкцией по технике безопасности и правилам поведения. Необходимо неукоснительно соблюдать правила по технике безопасности, т.к. нарушение этих правил может привести к поражению электрическим током, вызвать возгорание и навредить вашему здоровью.

* Работа обучающихся в компьютерном классе разрешается только в присутствии преподавателя (лаборанта).
* Во время занятий посторонние лица могут находиться в классе только с разрешения преподавателя.
* Во время перемен между уроками проводится обязательное проветривание компьютерного кабинета с обязательным выходом обучающихся из класса.

Каждый обучающийся в ответе за состояние своего рабочего места и сохранность размещенного на нем оборудования.

Обучающийся, допустивший нарушение по технике безопасности и правилам поведения в компьютерном классе, может быть привлечен к дисциплинарной ответственности. Если нарушение техники безопасности связано с причинением имущественного ущерба, обучающийся несет и материальную ответственность в установленном законом порядке.

* 1. Нормативно-правовые документы

Оформление и содержание технического задания должно соответствовать требованиям стандарта ГОСТ 19.201-78. ЕСПД. Общие положения.

Техническое задание оформляют в соответствии с ГОСТ 19.106-78 на листах формата 11 и 12 по ГОСТ 2.301-68, как правило, без заполнения полей листа. Номера листов (страниц) проставляются в верхней части листа над текстом.

Лист утверждения и титульный лист оформляют в соответствии с ГОСТ 19.104-78.

Информационную часть (аннотацию и содержание), лист регистрации изменений допускается в документ не включать.

Для внесения изменений или дополнений в техническое задание на последующих стадиях разработки про1раммы или программного изделия выпускают дополнение к нему.

Согласование и утверждение дополнения к техническому заданию проводят в том же порядке, который установлен для технического задания.

Техническое задание должно содержать следующие разделы:

* введение;
* основания для разработки;
* назначение разработки;
* требования к программе или программному изделию;
* требования к программной документации;
* технико-экономические показатели;
* стадии и этапы разработки;
* порядок контроля и приемки;
* в техническое задание допускается включать приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

В главе рассмотрены следующие вопросы:

* Техника безопасности во время прохождения учебной практики: основные требования, правила и меры предосторожности.
* Нормативно-правовые документы, регламентирующие разработку: основные законы, постановления и инструкции.
* Цели и задачи учебной практики: общие и конкретные задачи, которые должны быть выполнены студентами.
* Тематика отрасли и её тенденции развития: обзор отрасли, её основные направления и тенденции развития.

1. ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ПОСТАНОВКА ЗАДАЧИ

Предметная область информационной системы — это материальная система или система, характеризующая элементы материального мира, информация о которой хранится и обрабатывается. Предметная область рассматривается как некоторая совокупность реальных объектов и связей между ними.

В рамках учебной практики было дано следующее задание: «Разработать прикладное программное обеспечение деятельности OOO «Центр оценки и продажи недвижимости». Одним из источников прибыли этой организации является покупка и продажа квартир. Центр оценки имеет большой штат специалистов, позволяющий этой организации проводить сделки купли-продажи на высоком профессиональном уровне. Владелец квартиры, желающий ее продать, заключает договор с Центром, в котором указывается сумма, срок продажи и процент отчислений в пользу Центра оценки и продажи недвижимости в случае успешного проведения сделки. Один клиент может заключить с Центром более одного договора купли-продажи одновременно, если он владеет несколькими квартирами. Обмен квартир специалисты центра непосредственно не производят. Для этих целей используется вариант купли-продажи.». Для начала выполнения данного задания необходимо проанализировать предметную область.

* 1. Описание предметной области

Проанализировав предметную область и данные предоставленные вариантом учебной практики в Приложение А – рисунок 1, можно сделать вывод о необходимости 8 таблиц (сущностей) со следующими названиями: Клиенты, Договоры, Агенты, Квартиры, Дома, Продление, План квартир, Фотографии домов.

Таблица (сущность) Клиенты включает в себя следующие поля (атрибуты): Регистрационный номер, Фамилия, Имя, Отчество, Телефон клиента, Адрес клиента.

Таблица (сущность) Договоры включает в себя следующие поля (атрибуты): Регистрационный номер договора, Регистрационный номер, Код агента, Код квартиры, Начало действий договора, Окончание действий договора, Стоимость квартиры, Дополнительные условия, Вознаграждение Центра оценки, где поле Регистрационный номер является полем внешнего ключа и соединен с полем первичного ключа таблицы (сущности) Клиент по полю Регистрационный номер, соединенные связью «один-ко-многим», а так же, где поле Код агента является полем внешнего ключа и соединен с полем первичного ключа таблицы (сущности) Агенты по полю Код агента соединенные связью «один-ко-многим». Поле Код квартиры является полем внешнего ключа и соединен с полем первичного ключа таблицы (сущности) Квартиры по полу Код квартиры, соединенные связью «один-ко-многим».

Таблица (сущность) Агенты включает в себя следующие поля (атрибуты):

Фамилия, Имя, Отчество, Окончание действий.

Таблица (сущность) Квартиры включает в себя следующие поля (атрибуты): Код квартиры, Код дома, Документ на право собственности, Наличие приватизации, Общая площадь, Этаж квартиры, Тип планировки, Тип санузла, Адрес квартиры, Жилая площадь, Площадь кухни, Код плана квартиры, где поле Код дома является полем внешнего ключа и соединен с полем первичного ключа таблицы (сущности) Дома по полю Код дома, соединенные связью «один-ко-многим», а так же, где поле Код плана квартиры является полем внешнего ключа и соединен с полем первичного ключа таблицы (сущности) План квартир по полю Код плана квартиры соединенные связью «один-ко-многим».

Таблица (сущности) План квартир включает в себя следующие поля (атрибуты): Код плана квартиры, План квартиры.

Таблица (сущность) Дома включает в себя следующие поля (атрибуты): Код дома, Тип дома, Этажей в доме, Район города, Инфраструктура территории, Код фотографии, где поле Код фотографии является полем внешнего ключа и соединен с полем первичного ключа таблицы (сущности) Фотографии домов по полю Код фотографии, соединенные связью «один-ко-многим».

Таблица (сущность) Фотографии домов включает себя следующие поля (атрибуты): Код фотографии, Фотография здания.

Таблица (сущность) Продление включает в себя следующие поля (атрибуты): Код продления, Продление срока действия договора, Код договора, где поле Код договора является полем внешнего ключа и соединен с полем первичного ключа таблицы (сущности) Договоры по полю Код договора, соединенные связью «один-ко-многим».

По вышеперечисленной информации выполнена логическая модель данных представлена в Приложение А – рисунок 2.

* 1. Описание групп пользователей информационной системы “Покупка и продажа квартир”

Потенциальные пользователи, которые будут взаимодействовать с информационной системой:

Гость – просмотр данных.

Оператор – сопровождает информационную систему и обеспечивает её защиту, оптимизирует. Разрешено изменять таблицы базы данных.

Агент – получает доступ к своим данным, так же к подписанным договорам. Разрешено чтение;

Клиент – получает доступ к своим данным и к своему подписанному договору. Разрешено чтение;

Администратор – сопровождает информационную систему и обеспечивает её защиту, оптимизирует. По требованию предоставляет полномочия доступа пользователям. Разрешен полный доступ к базе данных.

* 1. Основные требования, предъявляемые к информационной системе “Покупка и продажа квартир”

Информационная система должна соответствовать следующим требованиям:

Гибкость – это способность к адаптации и дальнейшему развитию подразумевает возможность приспособления информационной системы к новым условиям, новым потребностям предприятия.

Выполнение этих условий возможно, если на этапе разработки информационной системы использовались общепринятые средства и методы документирования, так что по прошествии определенного времени сохранится возможность разобраться в структуре системы и внести в нее соответствующие изменения, даже если все разработчики или их часть по каким-либо причинам не смогут продолжить работу.

Надежность – функционирование без искажения информации, потери данных по «техническим причинам».

Требование надежности обеспечивается созданием резервных копий хранимой информации, выполнения операций протоколирования, поддержанием качества каналов связи и физических носителей информации, использованием современных программных и аппаратных средств. Сюда же следует отнести защиту от случайных потерь информации в силу недостаточной квалификации персонала.

Эффективность – это требование, при котором система с учетом выделенных ей ресурсов позволяет решать возложенные на нее задачи в минимальные сроки.

В любом случае оценка эффективности будет производиться заказчиком, исходя из вложенных в разработку средств и соответствия представленной информационной системы его ожиданиям.

Безопасность – это свойство системы, в силу которого посторонние лица не имеют доступа к информационным ресурсам организации, кроме тех, которые для них предназначены.

Защита информации от постороннего доступа обеспечивается управлением доступом к ресурсам системы, использованием современных программных средств защиты информации. В крупных организациях целесообразно создавать подразделения, основным направлением деятельности которых было бы обеспечение информационной безопасности, в менее крупных организациях назначать сотрудника, ответственного за данный участок работы.

В главе были рассмотрены следующие аспекты:

* Предметная область: что она собой представляет, как ее определить и описать.
* Группы пользователей: кто будет пользоваться информационной системой, какие задачи они будут решать с ее помощью и как их потребности влияют на проектирование информационной системы.
* Системные требования: какие технические характеристики и возможности должна иметь информационная система, чтобы удовлетворять потребности пользователей и обеспечивать ее эффективную работу.

1. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ “ПОКУПКА И ПРОДАЖА КВАРТИР”

Проектированием информационных систем называется многоступенчатый процесс их создания и/или модернизации путём применения упорядоченной совокупности методологий и инструментария. Проектирование (в отличие от моделирования) предполагает работу с пока несуществующим объектом и направлено на создание информационной системы.

* 1. Моделирование некоторых динамических аспектов системы

Диаграммы деятельности применяются для моделирования динамических аспектов поведения системы. Эти динамические аспекты могут включать деятельность на любом уровне абстракции в любом представлении системной архитектуры, включая классы (в том числе активные), интерфейсы, компоненты и узлы.

Использовать диаграммы деятельности для моделирования некоторых динамических аспектов системы вы можете в контексте почти любого моделируемого элемента. Однако чаще вы будете прибегать к таким диаграммам в контексте всей системы, подсистемы, операции или класса. Диаграмму деятельности можно присоединить к варианту использования, чтобы моделировать сценарий, и к кооперации, чтобы моделировать динамические аспекты поведения совокупности объектов.

При разработке программного обеспечения заблаговременное планирование и моделирование значительно упрощают программирование. На этапе создания концептуальной модели для описания функционального назначения используется диаграмма вариантов использования, а для описания логики процедур и бизнес-процессов используется диаграмма деятельности.

* + 1. Создание диаграммы вариантов использования

Перед началом разработки информационной системы необходимо создать визуальное представление о вариантах использования разрабатываемой системы.

Диаграмма вариантов использования является концептуальным представлением системы в процессе ее проектирования и разработки. Диаграмма вариантов использования описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования.

Цели построения диаграммы вариантов использования:

* определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования;
* сформулировать общие требования к функциональному проектированию системы;
* разработать исходную концептуальную модель системы для ее последующей реализации;
* подготовить документацию для взаимодействия разработчика системы с ее заказчиком и пользователями.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых, вариантов использования. При этом актером или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик.

В свою очередь, вариант использования служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером.

В приложении А на рисунке 3 представлена диаграмма вариантов использования информационной системы «Покупка и продажа квартир»

На диаграмме представлены такие актеры как Администратор, Гость, Клиент, Агент и Оператор.

* + 1. Создание диаграммы деятельности

В приложении А на рисунках 4 – 8 представлена диаграмма деятельности (по ролям) информационной системы «Покупка и продажа квартир».

* 1. Разработка функциональной структуры информационной системы. Карта навигации по информационной системе “Покупка и продажа квартир”

Карта переходов по информационной системе «Покупка и продажа квартир» представлена в приложении А на рисунке 9

* 1. Разработка макета информационной системы “Покупка и продажа квартир”

В приложении А на рисунке 10 представлено окно авторизации.

На этапе авторизации пользователь может ввести логин и пароль, чтобы продолжить работу в информационной системе.

В случае если пользователь еще не зарегистрирован в системе, он может сделать это, нажав на кнопку «Зарегистрироваться». В этом случае откроется окно регистрации, представленное в приложении А на рисунке 11.

На этапе регистрации пользователю необходимо придумать свой логин и пароль.

После того как пользователь успешно прошел авторизацию, перед ним откроется меню с таблицами которые доступны конкретному пользователю. В приложении А на рисунках 12 - 16 представлены все меню. Например, для гостя будут скрыты таблицы: Клиенты, Договоры, Квартиры, Дома и кнопки редактирования, добавления и удаления записи, так как данной группе пользователей доступен только просмотр данных.

Дизайн выполнен в простом стиле для того, чтобы пользователь с любым уровнем владения компьютером мог без препятствий работать с программным обеспечением.

В главе были разработаны следующие проектные документы:

* Диаграмма вариантов использования: описывает, как пользователи будут взаимодействовать с информационной системой.
* Диаграмма деятельности: описывает, как будет осуществляться выполнение функций информационной системы.
* Макеты: представляют собой визуальные представления интерфейса информационной системы.

1. РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ “ПОКУПКА И ПРОДАЖА КВАРТИР”

Прототип — это один из этапов разработки, который заключается в продумывании содержания и расположения важных элементов интерфейса. Прототип — это моделирование конечного продукта. Это интерактивный макет, который может иметь любую степень точности. Основная цель создания прототипов – проверить, насколько последователен путь пользователя, и выявить препятствия, которые могут возникнуть в процессе его взаимодействия с продуктом.

* 1. Моделирование и разработка базы данных

Исходя из анализа предметной области, можно выделить 8 таблиц (сущностей) со следующими названиями: Клиенты, Договоры, Агенты, Квартиры, Дома, Продление, План квартир, Фотографии домов.

В приложении А на рисунке 1 представлена логическая модель данных базы данных «Покупка и продажа квартир».

Создание базы данных происходит в MS SQL, в примере представлено создание базы данных в MS SQL Server.

С помощью раздела «Создание» и конструктора таблиц MS SQL созданы все таблицы и поля, представленные на логической модели данных.

Названия полей и их типы данных представлены в таблицах 1-8.

Таблица 1 – Поля таблицы «Клиенты»

|  |  |
| --- | --- |
| Название | Тип данных |
| Регистрационный номер | Числовой |
| Фамилия | Текстовый |
| Имя | Текстовый |
| Отчество | Текстовый |
| Телефон | Текстовый |
| Адрес клиента | Текстовый |

Таблица 2 – Поля таблицы «Агенты»

|  |  |
| --- | --- |
| Название | Тип данных |
| Код агента | Числовой |
| Фамилия | Текстовый |
| Имя | Текстовый |
| Отчество | Текстовый |
| Окончание действий | Дата |

Таблица 3 – Поля таблицы «Квартиры»

|  |  |
| --- | --- |
| Название | Тип данных |
| Код квартиры | Числовой |
| Код дома | Числовой |
| Документ на право собственности | Текстовый |
| Наличие приватизации | Логический |
| Общая площадь | Числовой |
| Этаж квартиры | Числовой |
| Тип планировки | Текстовый |
| Тип санузла | Текстовый |
| Адрес квартиры | Текстовый |
| Жилая площадь | Числовой |
| Площадь кухни | Числовой |
| Код плана квартиры | Числовой |

Таблица 4 – Поля таблицы «План квартир»

|  |  |
| --- | --- |
| Название | Тип данных |
| Код плана квартиры | Числовой |
| План квартиры | OLE |

Таблица 5 – Поля таблицы «Дома»

|  |  |
| --- | --- |
| Название | Тип данных |
| Код дома | Числовой |
| Тип дома | Текстовый |
| Этажей в доме | Числовой |
| Район города | Текстовый |
| Инфраструктура | Текстовый |
| Код фотографии | Числовой |

Таблица 6 – Поля таблицы «Фотографии домов»

|  |  |
| --- | --- |
| Название | Тип данных |
| Код фотографии | Числовой |
| Фотография здания | OLE |

Таблица 7 – Поля таблицы «Договоры»

|  |  |
| --- | --- |
| Название | Тип данных |
| Регистрационный номер договора | Числовой |
| Регистрационный номер | Числовой |
| Код агента | Числовой |
| Код квартиры | Числовой |
| Начало действий договора | Дата |
| Окончание действий договора | Дата |
| Стоимость квартиры | Денежный |
| Дополнительные условия | Текстовый |
| Вознаграждение Центра оценки | Денежный |

Таблица 8 – Поля таблицы «Продление»

|  |  |
| --- | --- |
| Название | Тип данных |
| Код продления | Числовой |
| Продление срока действия договора | Дата |
| Регистрационный номер договора | Числовой |

В приложении А на рисунке 17 представлены все созданные таблицы в обозревателе объектов базы данных.

После того, как таблицы созданы, необходимо создать между ними связи.

В приложении А на рисунке 18 представлена физическая модель данных, отражающая все таблицы и их поля, а также связи между таблицами.

* 1. Разработка программных модулей информационной системы «Покупка и продажа квартир»

Разработанные в MS Visual Studio формы продемонстрированы в приложении А на рисунках 10 – 16 и 19 – 25.

Свойства элементов формы Form2 (Авторизация) представлены в таблице 9

Таблица 9 – Свойства элементов формы Form2 (Авторизация)

| Объект | Свойство | Значение |
| --- | --- | --- |
| Form2 (Авторизация) | FormBorderStyle | Sizable |
| Text | Авторизация |
| StartPosition | CenterScreen |
| Size | 277; 361 |
| ShowIcon | True |

Событийно-управляемые процедуры в форме Form2 (Авторизация) представлены в таблице 10

Таблица 10 – Событийно-управляемые процедуры в форме Form2 (Авторизация)

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Form2 (Авторизация) | public partial class Form2 : Form  {  public Form2()  {  InitializeComponent();  }  } |
| Событие textBox1\_KeyPress\_1 (Логин) | private void textBox1\_KeyPress\_1(object sender, KeyPressEventArgs e)  {  textBox1.MaxLength = 30;  if (char.IsDigit(e.KeyChar))  {  e.Handled = true;  MessageBox.Show("Поле не может содержать цифры!", "Ошибка");  }  } |
| Событие textBox2\_KeyPress\_1 (Пароль) | private void textBox2\_KeyPress\_1(object sender, KeyPressEventArgs e)  {  textBox2.MaxLength = 15;  } |
| Кнопка Войти | int j = 0;  private void button1\_Click\_1(object sender, EventArgs e)  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog =уП01\_ИСПП5\_Работягова\_АА ; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter("select \* from Пользователи", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  Boolean flag = false;  for (int i = 0; i < dt.Rows.Count; i++)  if ((dt.Rows[i]["Логин"].ToString() == textBox1.Text) && (dt.Rows[i]["Пароль"].ToString() == textBox2.Text)) { j = i; flag = true; }  if (flag==true)  {    switch (dt.Rows[j]["Права доступа"].ToString())  {  case null:  {  MessageBox.Show("Такого аккаунта не существует", "Проверьте данные и попоробуйте снова", MessageBoxButtons.OK, MessageBoxIcon.Warning);  break;  }  case "Администратор":  {  this.Hide();  Form5 form5 = new Form5();  form5.Show();  form5.а("Администратор");    break;  }  case "Клиент":  {  this.Hide();  Form10 form10 = new Form10();  form10.Show();  break;  }  case "Агент":  {  this.Hide();  Form11 form11 = new Form11();  form11.Show();  break;  }  case "Оператор":  {  this.Hide();  Form5 form5 = new Form5();  form5.Show();  form5.а("Оператор");  break;  }    }  }  else  {  MessageBox.Show("Заполните все обязательные поля", "Заполнение полей", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  } |
| Кнопка Зарегистрироваться | private void button3\_Click\_1(object sender, EventArgs e)  {  this.Hide();  Form3 UsrFrm = new Form3();  UsrFrm.Show();  Form3 form3 = new Form3();  form3.load();  } |
| Кнопка Войти как гость | private void button2\_Click\_1(object sender, EventArgs e)  {  this.Hide();  Form13 form13 = new Form13();  form13.Show();    } |
| Флаговая кнопка Показать пароль | if (checkBox1.Checked)  {  textBox2.UseSystemPasswordChar = false;  }  else  {  textBox2.UseSystemPasswordChar = true;  } |

Свойства элементов формы Form3 (Регистрация) представлены в таблице 11.

Таблица 11 – Свойства элементов формы Form3 (Регистрация)

| Объект | Свойство | Значение |
| --- | --- | --- |
| Form3 (Регистрация) | FormBorderStyle | Sizable |
| Text | Регистрация |
| StartPosition | CenterScreen |
| Size | 219; 243 |
| ShowIcon | True |

Событийно-управляемые процедуры в форме Form3 (Регистрация) представлены в таблице 12

Таблица 12 – Событийно-управляемые процедуры в форме Form3 (Регистрация)

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Form3 (Регистрация) | public partial class Form3 : Form  {  public Form3()  {  InitializeComponent();  }  } |
| Событие textBox1\_KeyPress (Логин) | private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)  {  textBox1.MaxLength = 30;  if (char.IsDigit(e.KeyChar))  {  e.Handled = true;  MessageBox.Show("Поле не может содержать цифры!", "Ошибка");  }  } |
| Событие textBox2\_KeyPress (Пароль) | private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)  {  textBox2.MaxLength = 15;  } |
| Кнопка Отмена | private void button2\_Click(object sender, EventArgs e)  {  this.Hide();  Form2 form2 = new Form2();  form2.Show();  } |
| Кнопка Зарегистрироваться | private void button1\_Click(object sender, EventArgs e)  {  SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Работягова\_АА; Integrated Security = True");  sqlConnect.Open();  DataTable dt = new DataTable();  DataTable dta = new DataTable();  SqlDataAdapter daa = new SqlDataAdapter($"select \* from Пользователи", sqlConnect);  daa.Fill(dta);  string n = dta.Rows[dta.Rows.Count - 1]["Id"].ToString();  SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Пользователи (Id,Логин,Пароль) VALUES ('{int.Parse(n) + 1}','{textBox1.Text}','{textBox2.Text}');", sqlConnect);  da.Fill(dt);  пользователиTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи);  MessageBox.Show($"Регистрация пользователя с логином {textBox1.Text} выполнена.\nВы зарегистрированы как гость");  this.Hide();  Form2 form2 = new Form2();  form2.Show();  } |
| Событие Form3\_Load | private void Form3\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи);  } |

Свойства элементов формы Form5 (Меню администратора и оператора) представлены в таблице 13.

Таблица 13 – Свойства элементов формы Form5 (Меню администратора и оператора).

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Form5 (Меню администратора и оператора) | FormBorderStyle | Sizable |
| Text | Меню администратора |
| StartPosition | CenterScreen |
| Size | 285; 320 |
| ShowIcon | True |

Событийно-управляемые процедуры в форме Form5 (Меню администратора и оператора) представлены в таблице 14.

Таблица 14 – Событийно-управляемые процедуры в форме Form5 (Меню администратора и оператора).

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Form5 (Меню администратора) | public partial class Form5 : Form  {  public Form5()  {  InitializeComponent();  }  } |
| Событие a(Ограничение на кнопку Права доступа) | string a;  public void а(string b)  {  if (b == "Администратор")  {  button7.Visible = true;  }  else  {  button7.Visible = false;  }  } |
| Кнопка Данные о клиентах | private void button1\_Click\_1(object sender, EventArgs e)  {  Hide();  Form1 form1 = new Form1();  form1.i("Администратор");  form1.d("Администратор");  form1.Show();  } |
| Кнопка Данные о договорах | private void button3\_Click(object sender, EventArgs e)  {  Hide();  Form4 form4 = new Form4();  form4.i("Администратор");  form4.d("Администратор");  form4.Show();  } |
| Кнопка Данные об агентах | private void button6\_Click(object sender, EventArgs e)  {  Hide();  Form6 form6 = new Form6();  form6.i("Администратор");  form6.d("Администратор");  form6.Show();  } |
| Кнопка Данные о квартирах | private void button5\_Click(object sender, EventArgs e)  {  Hide();  Form7 form7 = new Form7();  form7.i("Администратор");  form7.Show();  } |
| Кнопка Данные о домах | private void button4\_Click(object sender, EventArgs e)  {  Hide();  Form8 form8 = new Form8();  form8.i("Администратор");  form8.d("Администратор");  form8.Show();  } |
| Кнопка Вернуться к авторизации | private void button8\_Click(object sender, EventArgs e)  {  Close();  Form2 form5 = new Form2();  form5.Show();  } |
| Кнопка Выход | private void button2\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Кнопка Права доступа | private void button7\_Click(object sender, EventArgs e)  {  Hide();  Form9 form5 = new Form9();  form5.i("Администратор");  form5.Show();  } |

Свойства элементов формы Form10 (Меню клиента) представлены в таблице 15.

Таблица 15 – Свойства элементов формы Form10 (Меню клиента)

| Объект | Свойство | Значение |
| --- | --- | --- |
| Form10 (Меню клиента) | FormBorderStyle | Sizable |
| Text | Меню клиента |
| StartPosition | CenterScreen |
| Size | 254; 292 |
| ShowIcon | True |

Событийно-управляемые процедуры в форме Form10 (Меню клиента) представлены в таблице 16.

Таблица 16 – Событийно-управляемые процедуры в форме Form10 (Меню клиента)

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Form10 (Меню клиента) | public partial class Form 10 : Form  {  public Form 10 ()  {  InitializeComponent();  }  } |
| Кнопка Данные о договорах | private void button3\_Click(object sender, EventArgs e)  {  Hide();  Form4 form4 = new Form4();  form4.i("Клиент");  form4.d("Клиент");  form4.Show();  } |
| Кнопка Данные об агентах | private void button6\_Click(object sender, EventArgs e)  {  Hide();  Form6 form6 = new Form6();  form6.i("Клиент");  form6.d("Клиент");  form6.Show();  } |
| Кнопка Данные о квартирах | private void button5\_Click(object sender, EventArgs e)  {  Hide();  Form7 form7 = new Form7();  form7.i("Клиент");  form7.d("Клиент");  form7.Show();  } |
| Кнопка Данные о домах | private void button4\_Click(object sender, EventArgs e)  {  Hide();  Form8 form8 = new Form8();  form8.i("Клиент");  form8.d("Клиент");  form8.Show();  } |
| Кнопка Вернуться к авторизации | private void button8\_Click(object sender, EventArgs e)  {  Close();  Form2 form10 = new Form 2();  Form10.Show();  } |
| Кнопка Выход | private void button2\_Click(object sender, EventArgs e)  {  Application.Exit();  } |

Свойства элементов формы Form11 (Меню агента) представлены в таблице 17.

Таблица 17 – Свойства элементов формы Form11 (Меню агента)

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Form11 (Меню агента) | FormBorderStyle | Sizable |
| Text | Меню агента |
| StartPosition | CenterScreen |
| Size | 270; 276 |
| ShowIcon | True |

Событийно-управляемые процедуры в форме Form11 (Меню агента) представлены в таблице 18.

Таблица 18 – Событийно-управляемые процедуры в форме Form11 (Меню агента)

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Form11 (Меню агента) | public partial class Form11 : Form  {  public Form11()  {  InitializeComponent();  }  } |
| Кнопка Данные о клиентах | private void button1\_Click\_1(object sender, EventArgs e)  {  Hide();  Form1 form1 = new Form1();  form1.i("Агент");  form1.d("Агент");  form1.Show();  } |
| Кнопка Данные о договорах | private void button3\_Click(object sender, EventArgs e)  {  Hide();  Form4 form4 = new Form4();  form4.i("Агент");  form4.d("Агент");  form4.Show();  } |
| Кнопка Данные о квартирах | private void button5\_Click(object sender, EventArgs e)  {  Hide();  Form7 form7 = new Form7();  form7.i("Агент");  form7.d("Агент");  form7.Show();  } |
| Кнопка Данные о домах | private void button4\_Click(object sender, EventArgs e)  {  Hide();  Form8 form8 = new Form8();  form8.i("Агент");  form8.d("Агент");  form8.Show();  } |
| Кнопка Вернуться к авторизации | private void button8\_Click(object sender, EventArgs e)  {    Close();  Form2 form8 = new Form2();  form8.Show();    } |
| Кнопка Выход | private void button2\_Click(object sender, EventArgs e)  {  Application.Exit();  } |

Свойства элементов формы Form13 (Меню гостя) представлены в таблице 19.

Таблица 19 – Свойства элементов формы Form13 (Меню гостя)

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Form13 (Меню гостя) | FormBorderStyle | Sizable |
| Text | Меню гостя |
| StartPosition | CenterScreen |
| Size | 278; 254 |
| ShowIcon | True |

Событийно-управляемые процедуры в форме Form13 (Меню гостя) представлены в таблице 20.

Таблица 20 – Событийно-управляемые процедуры в форме Form13 (Меню гостя)

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Form13 (Меню гостя) | public partial class Form 13 : Form  {  public Form 13 ()  {  InitializeComponent();  }  } |
| Кнопка Данные об агентах | private void button6\_Click(object sender, EventArgs e)  {  Hide();  Form6 form6 = new Form6();  form6.i("Гость");  form6.d("Гость");  form6.Show();  } |
| Кнопка Вернуться к авторизации | private void button8\_Click(object sender, EventArgs e)  {  Close();  Form2 form13 = new Form2();  form13.Show();  } |
| Кнопка Выход | private void button2\_Click(object sender, EventArgs e)  {  Application.Exit();  } |

Свойства элементов формы Form1 (Данные о клиентах) представлены в таблице 21.

Таблица 21 – Свойства элементов формы Form1 (Данные о клиентах)

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Form1 (Данные о клиентах) | FormBorderStyle | Sizable |
| Text | Данные о клиентах |
| StartPosition | CenterScreen |
| Size | 998; 302 |
| ShowIcon | True |

Событийно-управляемые процедуры в форме Form1 (Данные о клиентах) представлены в таблице 22.

Таблица 22 – Событийно-управляемые процедуры в форме Form1 (Данные о клиентах)

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Form1 (Данные о клиентах) | public partial class Form1 : Form  {  public Form1 ()  {  InitializeComponent();    }  } |
| Событие Form1\_Load | private void Form1\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Клиенты". При необходимости она может быть перемещена или удалена.  this.клиентыTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Клиенты);  } |
| Кнопка Обновить | private void button1\_Click(object sender, EventArgs e)  {  клиентыTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Клиенты);  } |
| Событие ограничение на кнопку Вернуться в меню | private void button2\_Click(object sender, EventArgs e)  {  if (dost == "Администратор")  {  Hide();  Form5 form5 = new Form5();  form5.Show();  }  else  {  Hide();  Form11 form11 = new Form11();  form11.Show();  }  } |
| Кнопка Отмена | private void CancelButton\_Click(object sender, EventArgs e)  {  клиентыBindingSource1.CancelEdit();  } |
| Событие ограничение на кнопки Удалить и Добавить | public void d(string b)  {  if (b == "Администратор")  {  bindingNavigatorAddNewItem.Visible = true;  bindingNavigatorDeleteItem.Visible = true;  }  else  {  bindingNavigatorAddNewItem.Visible = false;  bindingNavigatorDeleteItem.Visible = false;  }  } |

Свойства элементов формы Form4 (Данные о договорах) представлены в таблице 23.

Таблица 23 – Свойства элементов формы Form4 (Данные о договорах)

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Form4 (Данные о договорах) | FormBorderStyle | Sizable |
| Text | Данные о договорах |
| StartPosition | CenterScreen |
| Size | 979; 383 |
| ShowIcon | True |

Событийно-управляемые процедуры в форме Form4 (Данные о договорах) представлены в таблице 24.

Таблица 24 – Событийно-управляемые процедуры в форме Form4 (Данные о договорах)

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Form4 (Данные о договорах) | public partial class Form4 : Form  {  public Form4 ()  {  InitializeComponent();    }  } |
| Событие Form4\_Load | private void Form4\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Договоры". При необходимости она может быть перемещена или удалена.  this.договорыTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Договоры);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Продление". При необходимости она может быть перемещена или удалена.  this.продлениеTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Продление);  } |
| Кнопка Обновить | private void button1\_Click(object sender, EventArgs e)  {  продлениеTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Продление);  договорыTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Договоры);  } |
| Кнопка Отменить | private void CancelButton\_Click(object sender, EventArgs e)  {  fKПродлениеДоговорыBindingSource.CancelEdit();  договорыBindingSource.CancelEdit();  } |
| Событие ограничение на кнопки Удалить и добавить | public void d(string b)  {  if (b == "Администратор")  {  bindingNavigatorAddNewItem.Visible = true;  bindingNavigatorDeleteItem.Visible = true;  }  else  {  bindingNavigatorAddNewItem.Visible = false;  bindingNavigatorDeleteItem.Visible = false;  }  } |
| Событие ограничение на кнопку вернуться в меню | private void button2\_Click(object sender, EventArgs e)  {  if (dost == "Администратор")  {  Hide();  Form5 form5 = new Form5();  form5.Show();  }  else if (dost == "Клиент")  {  Hide();  Form10 form10 = new Form10();  form10.Show();  }  else  {  Hide();  Form11 form11 = new Form11();  form11.Show();  }  } |

Свойства элементов формы Form6 (Данные об агентах) представлены в таблице 25.

Таблица 25 – Свойства элементов формы Form6 (Данные об агентах)

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Form6 (Данные об агентах) | FormBorderStyle | Sizable |
| Text | Данные об агентах |
| StartPosition | CenterScreen |
| Size | 816; 281 |
| ShowIcon | True |

Событийно-управляемые процедуры в форме Form6 (Данные об агентах) представлены в таблице 26.

Таблица 26 – Событийно-управляемые процедуры в форме Form6 (Данные об агентах)

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Form6 (Данные об агентах) | public partial class Form6 : Form  {  public Form6()  {  InitializeComponent();  }  } |
| Событие Form6\_Load | private void Form6\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Агенты". При необходимости она может быть перемещена или удалена.  this.агентыTableAdapter2.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Агенты);  } |
| Кнопка Обновить | private void button1\_Click(object sender, EventArgs e)  {  агентыTableAdapter2.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Агенты);  } |
| Кнопка Отменить | private void CancelButton\_Click(object sender, EventArgs e)  {  агентыBindingSource.CancelEdit();  } |
| Событие ограничение на кнопки Удалить и Добавить | public void d(string b)  {  if (b == "Администратор")  {  bindingNavigatorAddNewItem.Visible = true;  bindingNavigatorDeleteItem.Visible = true;  }  else  {  bindingNavigatorAddNewItem.Visible = false;  bindingNavigatorDeleteItem.Visible = false;  }  } |
| Событие ограничение на кнопку Вернуться в меню | private void button2\_Click(object sender, EventArgs e)  {  if (dost == "Администратор")  {  Hide();  Form5 form5 = new Form5();  form5.Show();  }  else if (dost == "Клиент")  {  Hide();  Form10 form10 = new Form10();  form10.Show();  }  else  {  Hide();  Form13 form13 = new Form13();  form13.Show();  }  } |

Свойства элементов формы Form7 (Данные о квартирах) представлены в таблице 27.

Таблица 27 – Свойства элементов формы Form7 (Данные о квартирах)

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Form7 (Данные о квартирах) | FormBorderStyle | Sizable |
| Text | Данные о квартирах |
| StartPosition | CenterScreen |
| Size | 728; 423 |
| ShowIcon | True |

Событийно-управляемые процедуры в форме Form7 (Данные о квартирах) представлены в таблице 28.

Таблица 28 – Событийно-управляемые процедуры в форме Form7 (Данные о квартирах)

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Form7 (Данные о квартирах) | public partial class Form7 : Form  {  public Form7()  {  InitializeComponent();  }} |
| Событие Form7\_Load | private void Form7\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Квартиры". При необходимости она может быть перемещена или удалена.  this.квартирыTableAdapter2.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Квартиры);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet2.Квартиры". При необходимости она может быть перемещена или удалена.  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.План\_квартир". При необходимости она может быть перемещена или удалена.  this.план\_квартирTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.План\_квартир);  sqlConnect.Open();  } |
| Кнопка Обновить | private void button1\_Click(object sender, EventArgs e)  {    switch (c)  {  case 0:  план\_квартирTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.План\_квартир);  план\_квартирTableAdapter1.Fill(уП01\_ИСПП5\_Работягова\_ААDataSet3.План\_квартир); break;  case 1:  квартирыTableAdapter2.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Квартиры);  квартирыTableAdapter2.Fill(уП01\_ИСПП5\_Работягова\_ААDataSet3.Квартиры); break;  }  } |
| PictureSet | private void pictureSet\_Click(object sender, EventArgs e)  {  pictureSet.SizeMode = PictureBoxSizeMode.StretchImage;  } |
| Событие ограничение на кнопки Удалить и Добавить | public void d(string b)  {  if (b == "Администратор")  {  bindingNavigatorAddNewItem.Visible = true;  bindingNavigatorDeleteItem.Visible = true;  }  else  {  bindingNavigatorAddNewItem.Visible = false;  bindingNavigatorDeleteItem.Visible = false;  }  } |
| Событие ограничение на кнопку Вернуться в меню | private void button2\_Click(object sender, EventArgs e)  {  if (dost == "Администратор")  {  Hide();  Form5 form5 = new Form5();  form5.Show();  }  else if (dost == "Клиент")  {  Hide();  Form10 form10 = new Form10();  form10.Show();  }  else  {  Hide();  Form13 form13 = new Form13();  form13.Show();  }  } |
| Кнопка Найти | private void button3\_Click(object sender, EventArgs e)  {  OpenFileDialog load = new OpenFileDialog();  if (load.ShowDialog() == DialogResult.OK)  {  pictureSet.ImageLocation = load.FileName;  SqlDataAdapter a = new SqlDataAdapter($"Update [План квартир] set [План квартиры] = '{load.FileName}' where [Код плана квартиры] = '{dataGridView2.Rows[dataGridView2.SelectedCells[0].RowIndex].Cells[0].Value}'", sqlConnect);  DataSet dt = new DataSet();  a.Fill(dt);  this.план\_квартирTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.План\_квартир);  }  } |
| Подключение к базе данных | SqlConnection sqlConnect = new SqlConnection(“Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Работягова\_АА; Integrated Security = True”); |
| Кнопка Добавить | private void bindingNavigatorAddNewItem\_Click(object sender, EventArgs e)  {  OpenFileDialog load = new OpenFileDialog();  if (load.ShowDialog() == DialogResult.OK)  {  pictureSet.ImageLocation = load.FileName;  SqlDataAdapter a = new SqlDataAdapter($"Insert into [План квартир] Values( '{load.FileName}')", sqlConnect);  DataSet dt = new DataSet();  a.Fill(dt);  this.план\_квартирTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.План\_квартир);  }  } |
| Событие dataGridView2\_CellMouseClick(Таблица План квартир) | private void dataGridView2\_CellMouseClick(object sender, DataGridViewCellMouseEventArgs e)  {  try  {  pic = dataGridView2.Rows[dataGridView2.SelectedCells[0].RowIndex].Cells[1].Value.ToString();  pictureSet.Image = Image.FromFile(pic);  }  catch  {  }  } |

Свойства элементов формы Form8 (Данные о домах) представлены в таблице 29.

Таблица 29 – Свойства элементов формы Form8 (Данные о домах)

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Form8 (Данные о домах) | FormBorderStyle | Sizable |
| Text | Данные о домах |
| StartPosition | CenterScreen |
| Size | 714; 412 |
| ShowIcon | True |

Событийно-управляемые процедуры в форме Form8 (Данные о домах) представлены в таблице 30.

Таблица 30 – Событийно-управляемые процедуры в форме Form9 (Данные о домах)

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Form8 (Данные о домах) | public partial class Form8 : Form  {  public Form8()  {  InitializeComponent();  }  } |
| Событие Form8\_Load | private void Form8\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Дома". При необходимости она может быть перемещена или удалена.  this.домаTableAdapter2.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Дома);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Фотографии\_домов". При необходимости она может быть перемещена или удалена.  this.фотографии\_домовTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Фотографии\_домов);  } |
| Кнопка Обновить | private void button1\_Click(object sender, EventArgs e)  {  switch (c)  {  case 0:  фотографии\_домовTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Фотографии\_домов);  фотографии\_домовTableAdapter1.Fill(уП01\_ИСПП5\_Работягова\_ААDataSet3.Фотографии\_домов); break;  case 1:  домаTableAdapter2.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Дома);  домаTableAdapter2.Fill(уП01\_ИСПП5\_Работягова\_ААDataSet3.Дома); break;  }  } |
| PictureSet | private void pictureSet\_Click(object sender, EventArgs e)  {  pictureSet.SizeMode = PictureBoxSizeMode.StretchImage;  } |
| Событие ограничение на кнопки Удалить и Добавить | public void d(string b)  {  if (b == "Администратор")  {  bindingNavigatorAddNewItem.Visible = true;  bindingNavigatorDeleteItem.Visible = true;  }  else  {  bindingNavigatorAddNewItem.Visible = false;  bindingNavigatorDeleteItem.Visible = false;  }  } |
| Событие ограничение на кнопку Вернуться в меню | private void button2\_Click(object sender, EventArgs e)  {  if (dost == "Администратор")  {  Hide();  Form5 form5 = new Form5();  form5.Show();  }  else if (dost == "Клиент")  {  Hide();  Form10 form10 = new Form10();  form10.Show();  }  else  {  Hide();  Form11 form11 = new Form11();  form11.Show();  }  } |
| Кнопка Найти | private void button3\_Click(object sender, EventArgs e)  {  OpenFileDialog load = new OpenFileDialog();  if (load.ShowDialog() == DialogResult.OK)  {  pictureSet.ImageLocation = load.FileName;  SqlDataAdapter a = new SqlDataAdapter($"Update [Фотографии домов] set [Фотография здания] = '{load.FileName}' where [Код фотографии] = '{dataGridView2.Rows[dataGridView2.SelectedCells[0].RowIndex].Cells[0].Value}'", sqlConnect);  DataSet dt = new DataSet();  a.Fill(dt);  this.фотографии\_домовTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Фотографии\_домов);  }  } |
| Подключение к базе данных | SqlConnection sqlConnect = new SqlConnection(“Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Работягова\_АА; Integrated Security = True”); |
| Кнопка Добавить | private void bindingNavigatorAddNewItem\_Click(object sender, EventArgs e)  {  OpenFileDialog load = new OpenFileDialog();  if (load.ShowDialog() == DialogResult.OK)  {  pictureSet.ImageLocation = load.FileName;  SqlDataAdapter a = new SqlDataAdapter($"Insert into [Фотографии домов] Values( '{load.FileName}')", sqlConnect);  DataSet dt = new DataSet();  a.Fill(dt);  this.фотографии\_домовTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Фотографии\_домов);  }  } |
| Событие dataGridView2\_CellMouseClick(Таблица Фотографии домов) | private void dataGridView2\_CellMouseClick(object sender, DataGridViewCellMouseEventArgs e)  {  try  {  pic = dataGridView2.Rows[dataGridView2.SelectedCells[0].RowIndex].Cells[1].Value.ToString();  pictureSet.Image = Image.FromFile(pic);  }  catch  {  }  } |

Свойства элементов формы Form9 (Права доступа) представлены в таблице 31.

Таблица 31 – Свойства элементов формы Form9 (Права доступа)

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Form9 (Права доступа) | FormBorderStyle | Sizable |
| Text | Права доступа |
| StartPosition | CenterScreen |
| Size | 478; 253 |
| ShowIcon | True |

Событийно-управляемые процедуры в форме Form9 (Права доступа) представлены в таблице 32.

Таблица 32 – Событийно-управляемые процедуры в форме Form9 (Права доступа)

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Form9 (Права доступа) | public partial class Form9 : Form  {  public Form8()  {  InitializeComponent();  }  } |
| Событие Form9\_Load | private void Form9\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи);  } |
| Кнопка Обновить | private void button1\_Click(object sender, EventArgs e)  {  пользователиTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи);  } |
| Кнопка Вернутся в меню | private void button2\_Click(object sender, EventArgs e)  {  if (dost == "Администратор")  {  Hide();  Form5 form5 = new Form5();  form5.Show();  }  } |

В главе были завершены следующие этапы разработки информационной системы «Покупка и продажа квартир»:

Разработка базы данных: определена структура базы данных, созданы таблицы, индексы и другие объекты базы данных.

Разработка программных модулей: определены функции и алгоритмы каждого программного модуля, написан код модулей.

1. РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И  
   ТЕСТОВЫХ СЦЕНАРИЕВ

Тестирование программного обеспечения – процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Тестовый сценарий – это неавтоматизированный или автоматизированный сценарий, содержащий инструкции по реализации тестового набора. Тестовый сценарий может быть написан вручную (для выполнения человеком) либо полностью или частично автоматизирован.

* 1. Назначение эксперимента. Выбор и обоснование методики проведения тестирования

Основные цели тестирования программного обеспечения:

* предоставление информации о качестве ПО конечному заказчику;
* повышение качества ПО;
* предотвращение появления дефектов.

Цели тестирования могут отличаться, в зависимости от этапа разработки программного обеспечения, на котором оно проводится. К примеру, на этапе кодирования целью тестирования будет вызов как можно большего количества сбоев в работе программы, что позволит локализовать и исправить дефекты.

В то же время, при приемочном тестировании необходимо показать, что система работает правильно. В период сопровождения, тестирование в основном необходимо для того, чтобы убедится в отсутствии новых багов, появившихся во время внесения изменений.

Главная же задача тестирования – поиск дефектов.

В данном разделе проведены два вида тестирования. Провести анализ тестирования и выбрать, как будет проходить тестирование ИС.

В данном разделе проведены два вида тестирования. Провести анализ тестирования и выбрать, как будет проходить тестирование ИС.

Модульное тестирование. Цель: проверить, что код работает именно так, как должен (при заданных входных параметрах выдает предполагаемый результат).

Анализ покрытия кода тестами. Цель: проверить, что весь код отрабатывает при модульном тестировании, что нет, не участвующих в тестировании участков кода.

Анализ стилистики кода. Цель: проверка кода на соблюдение стандартов разработки (отраслевых, корпоративных, принятых на проекте).

Анализ производительности. Цель: проанализировать производительность ключевых операций, потребление памяти, утечку памяти.

* 1. Технология тестирования

Рассмотрим, какие технологии можно применить для модульного тестирования и покрытия кода тестами.

Как в ходе выполнения проекта, так и при развитии системы, изменения в ее алгоритмы вносятся в соответствии с техническими заданиями, хотя данный документ может иметь различные названия. В технических заданиях, как правило, описываются критерии приемки: как мы и заказчик убедимся в том, что заказанный функционал работает верно.

В качестве примера можно привести такой упрощенный критерий:

* Исходные данные: в систему внесен клиент, которому, например, нужно подписать договор с конкретным агентом.
* Действия пользователя: пользователь создает новую запись в таблице договоры, ищет нужно агента, вносит в базу данных квартиру и т п.
* В результате: отображен клиент, подходящий ему агент и добавлена квартира в базу данных.
* Вручную такое тестирование, как правило, выполняется единожды, поскольку выполнять каждый раз полный комплект тестов очень трудоемко.
* Нет гарантии того, что тесты, описанные в техническом задании, покрывают функциональность полностью. Т.е. даже если тесты описаны и проводятся регулярно, вероятно будет существовать наличие не протестированного функционала.

Автоматизированное тестирование лишено данных недостатков. Конечно, возникает отдельная задача кодирования данного теста на основании критериев приемки, но это скорее дисциплинирующий фактор…

Прежде чем начинать юзабилити-тестирование необходимо собрать группу пользователей, которые будут тестировать данную информационную систему. Количество привлеченных пользователей должно быть не менее пяти человек. Пользователям будет представлен список некоторых вопросов, относящихся к графическому интерфейсу.

В таблице 33 представлены сведения о пользователях.

Таблица 33 – Информация о пользователях

|  |  |  |  |
| --- | --- | --- | --- |
| ФИО | Пол | Возраст | Образование |
| Васильева Анастасия Витальевна | Женский | 18 | СПО |
| Демешко Екатерина Олеговна | Женский | 18 | СПО |
| Шипицына Ольга Ивановна | Женский | 18 | СПО |
| Дацюк Виктория Денисовна | Женский | 18 | СПО |
| Евдокимов Антон Андреевич | Мужской | 19 | СПО |

* 1. Результаты проведения тестирования

В ходе тестирования был применен метод нагрузочного тестирования и юзабилити-тестирования, которые проверяют работу при нагрузке системы и сделать простым и удобным в пользовании информационной системы.

Суть тестирования заключается в проверке работы всех обязательных функций.

Результаты проведения нагрузочного тестирования

Для проведения нагрузочного тестирования составлена таблица 34, в которой указано количество запущенных копий программного продукта, нагрузка на процесс и оперативную память.

Таблица 34 – Результаты нагрузочного тестирования

|  |  |  |
| --- | --- | --- |
| Число копий | Нагрузка на процессор (%) | Нагрузка на память (%) |
| 1 | 1,8 | 0,5 |
| 2 | 2,1 | 1 |
| 3 | 2,5 | 1,5 |
| 4 | 3,5 | 2 |
| 5 | 5 | 2,5 |

Исходя из результатов нагрузочного тестирования, можно прийти к выводу, что нагрузка, оказываемая на центральный процессор и память, не имеет критичных значений. Это значит, что показатель производительности информационной системы на приемлемом уровне, то есть сбои и длительная загрузка не должны помешать пользователю при работе с данным программным продуктом.

Таблица 23 – Результаты тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вопрос | Васильева Анастасия Витальевна | Демешко Екатерина Олеговна | Шипицына Ольга Ивановна | Дацюк Виктория Денисовна | Евдокимов Антон Андреевич |
| Присутствуют ли такие дизайн-элементы, которые неприятны для зрения? | нет | нет | нет | нет | нет |
| Присутствует ли отвлекающая информация? | нет | нет | нет | нет | нет |
| Понятна ли структура приложения на интуитивном уровне? | да | да | да | да | да |
| Смогли ли вы найти интересующую вас информацию без затруднений? | да | да | да | нет | да |
| Понятен ли смысл приложения? | да | да | да | да | да |
| Общая оценка | 7/10 | 7/10 | 9/10 | 9/10 | 9/10 |

Исходя из средней оценки информационной системы, рассчитанной на основе общих оценок всех пользователей с учетом ответов на все поставленные вопросы, можно сделать вывод о том, что внешние характеристики графического интерфейса находятся на довольно высоком уровне. Также в информационной системе нет помех для пользователя в виде отвлекающей информации, так как все отображаемые данные находятся в рамках предметной области.

В результате проведенных тестов было установлено, что программное обеспечение соответствует требованиям качества.

Модульное тестирование показало, что все важные части программного обеспечения работают правильно. Анализ покрытия кода тестами показал, что все важные части программного обеспечения покрыты тестами. Анализ стилистики кода выявил потенциальные проблемы с качеством кода, которые были устранены. Анализ производительности показал, что программное обеспечение работает эффективно. Нагрузочное тестирование показало, что нагрузка на процессор и память не превышает допустимых значений.

Таким образом, программное обеспечение может быть безопасно запущено в эксплуатацию.

ЗАКЛЮЧЕНИЕ

В результате учебной практики был проведен анализ предметной области, спроектирована и разработана информационная система «Покупка и продажа квартир», серверная часть которой была реализована в MS SQL, а клиентская часть в MS Visual Studio.

В MS SQL была создана база данных для хранения информации о клиентах, агентах, подписанных договорах и о квартирах.

Клиентская часть, реализованная в MS Visual Studio, отражает все хранящиеся в базе данных таблицы.

Цели учебной практики были достигнуты путем проведения анализа предметной области, проектирования информационной системы и разработки программных модулей.

Это позволяет сделать вывод, что поставленные цели и задачи выполнены в полном объеме.

В дальнейшем данная информационная система может быть модернизирована путем добавления формы личного кабинета, создания версии для слабовидящих людей или улучшения и изменения дизайна.

.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Microsoft SQL Server [Электронный доступ]. — Режим доступа: [Microsoft SQL Server — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server) (Дата обращения: 10.09.2023 г.).
2. Microsoft Visio [Электронный доступ]. — Режим доступа: [Microsoft Visio — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Microsoft_Visio) (Дата обращения: 07.08.2023 г.).
3. Microsoft Visual Studio [Электронный доступ]. — Режим доступа: [Руководство по разработке модулей расширений на C# — Павел Еремеев (pvs-studio.ru)](https://pvs-studio.ru/ru/blog/posts/csharp/a0082/) (Дата обращения: 29.08.2013 г.).
4. ГОСТ 19.201-78. ЕСПД. Общие положения. [Электронный доступ]. — Режим доступа: [ГОСТ 19.201-78.ЕСПД. Техническое задание. Требования к содержанию и оформлению (swrit.ru)](https://www.swrit.ru/doc/espd/19.201-78.pdf?ysclid=lpi5u6naom706376064) (Дата обращения: 01.10.1981 г.).
5. ГОСТ 20886-85. Организация данных в системах обработки данных. Термины и определения [Электронный доступ]. — Режим доступа: [ГОСТ 20886-85. Организация данных в системах обработки данных. Термины и определения (rostest.info)](https://rostest.info/gost/001.001.040.035/gost-20886-85/#:~:text=%D0%93%D0%9E%D0%A1%D0%A2%2020886%2D85.%20%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B2,%D0%BD%D0%B0%D1%83%D1%87%D0%BD%D0%BE%2D%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%) (Дата обращения: 22.05.2013 г.).
6. ГОСТ Р. 59793–2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания [Электронный доступ]. — Режим доступа: [ГОСТ Р. 59793-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания (internet-law.ru)](https://internet-law.ru/gosts/gost/77858/?ysclid=lifap277mg519765664) (Дата обращения: 30.04.2022 г.).
7. Дебби Валковски. Microsoft Office Visio 2003 для «чайников» = Visio 2003 For Dummies. — М: [«Диалектика»](https://ru.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D0%B0%D0%BB%D0%B5%D0%BA%D1%82%D0%B8%D0%BA%D0%B0_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2006. — С. 336.
8. Майо Д. Самоучитель Microsoft Visual Studio 2010 = Microsoft Visual Studio 2010: A Beginner's Guide (A Beginners Guide). — C: [«БХВ-Петербург»](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%A5%D0%92-%D0%9F%D0%B5%D1%82%D0%B5%D1%80%D0%B1%D1%83%D1%80%D0%B3_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2010. — С. 464.
9. Роберт Э. Уолтерс, Майкл Коулс. SQL Server 2008: ускоренный курс для профессионалов = Accelerated SQL Server 2008. — М: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2008. — С. 768.

ПРИЛОЖЕНИЕ А



Рисунок 1 – Данные предоставленные вариантом учебной практики

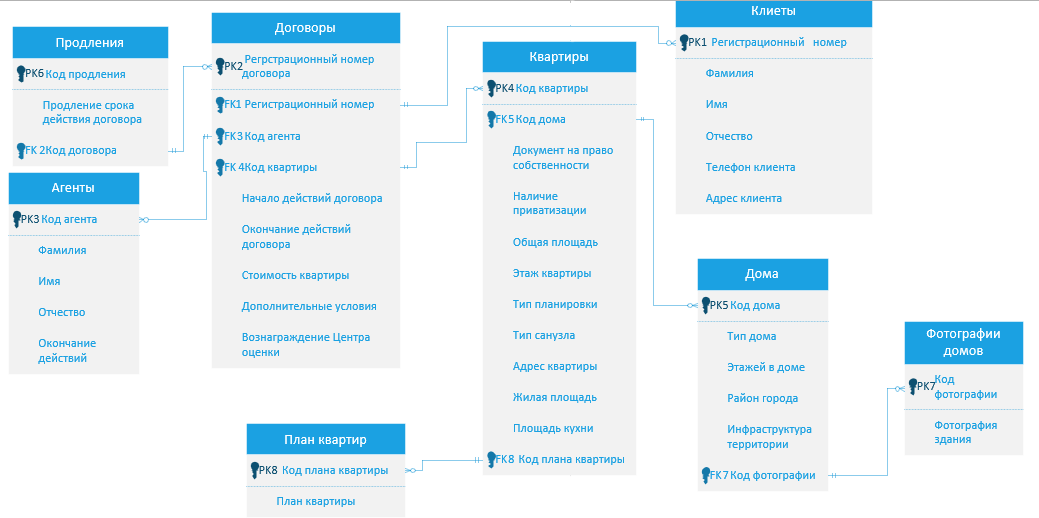


Рисунок 2 – Логическая модель данных

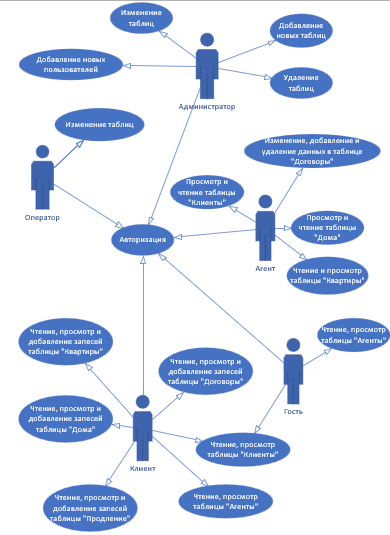


Рисунок 3 **–** Диаграмма вариантов использования

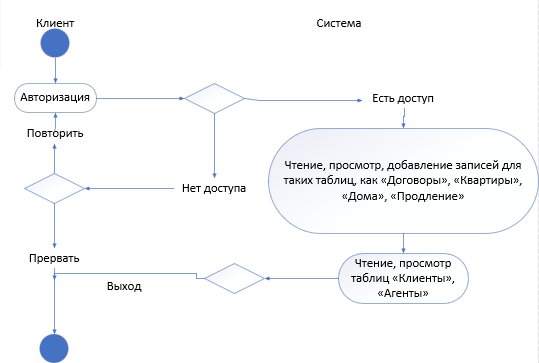


Рисунок 4 – Диаграмма деятельности «Клиент»

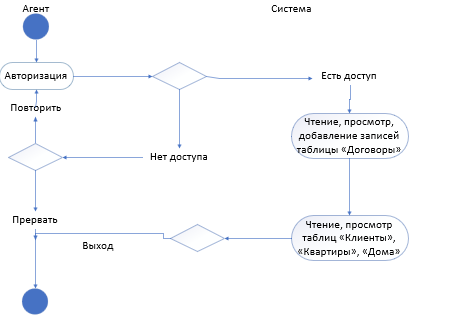


Рисунок 5 – Диаграмма деятельности «Агент»

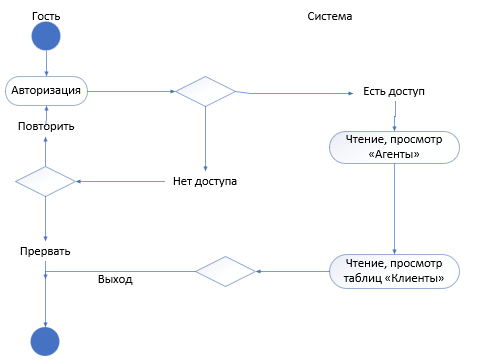


Рисунок 6 – Диаграмма деятельности «Гость»

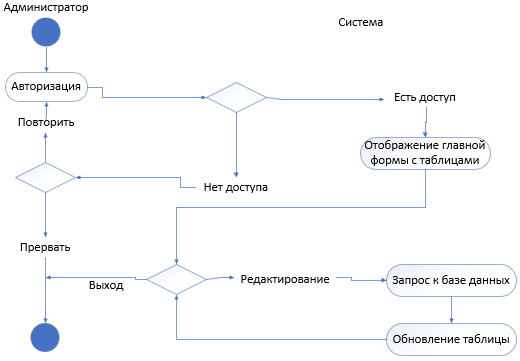


Рисунок 7 – Диаграмма деятельности «Администратор»



Рисунок 8 – Диаграмма деятельности «Оператор»

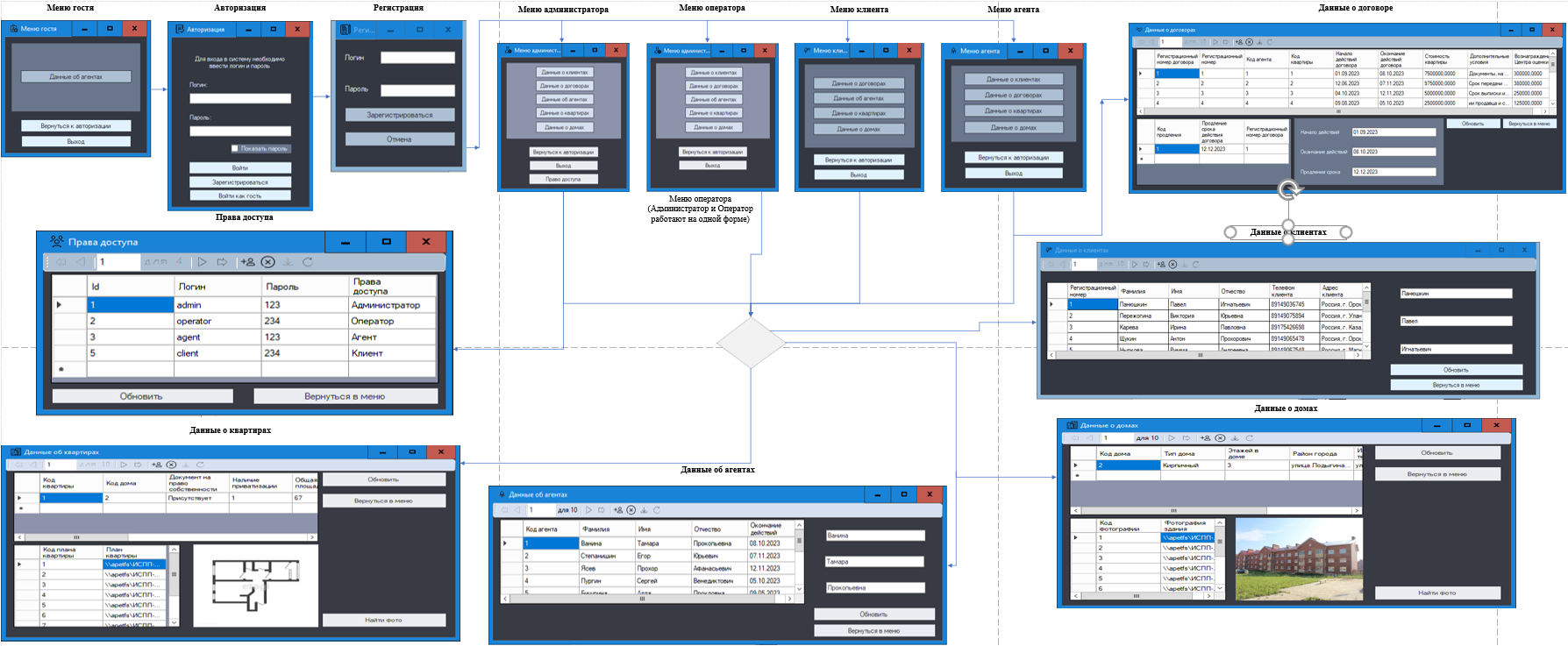


Рисунок 9 – Карта переходов

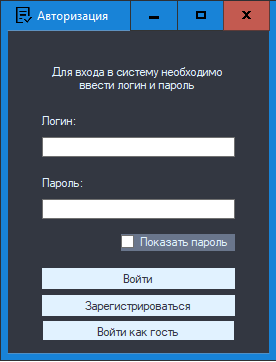


Рисунок 10 – Форма “Авторизации”

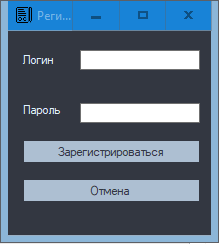


Рисунок 11 – Форма “Регистрация”

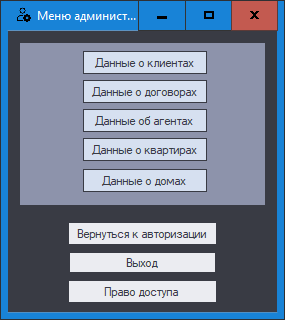


Рисунок 12 – Форма “Меню администратора”

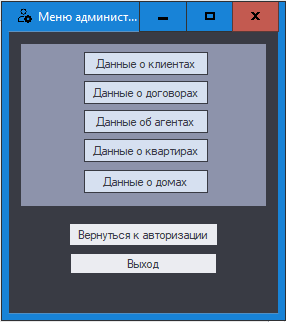


Рисунок 13 – Форма “Меню оператора”

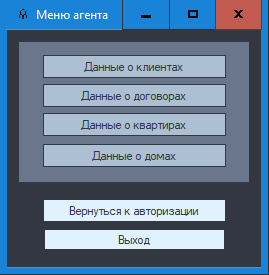


Рисунок 14 – Форма “Меню агента”

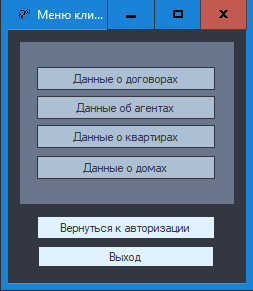


Рисунок 15 – Форма “Меню клиента”

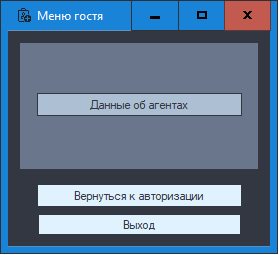


Рисунок 16 – Форма “Меню гостя”

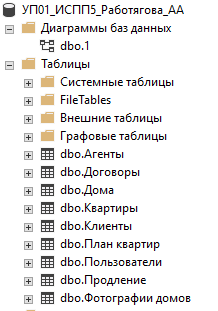


Рисунок 17 – Созданные таблицы

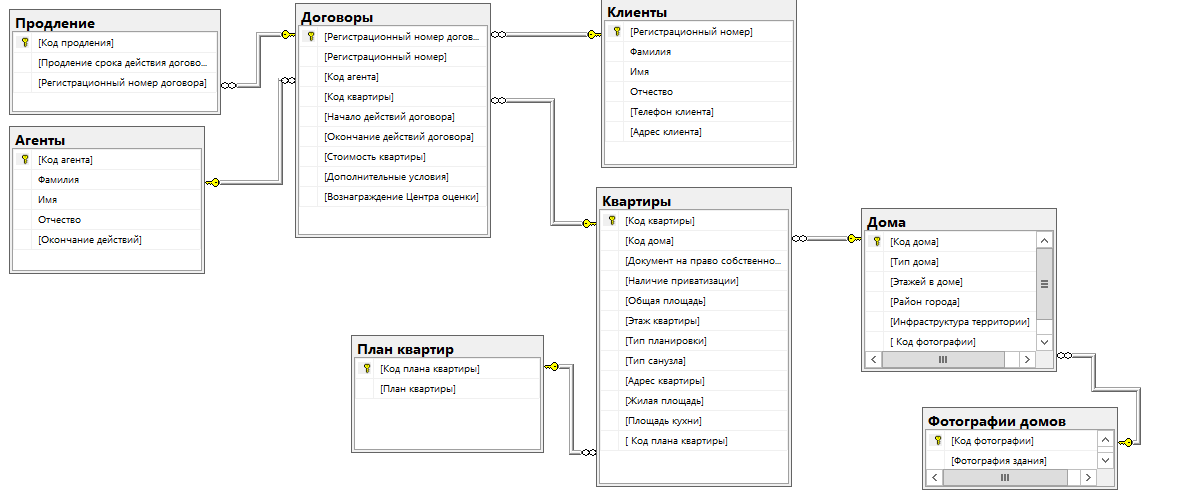


Рисунок 18 – Физическая модель базы данных

Форма “”

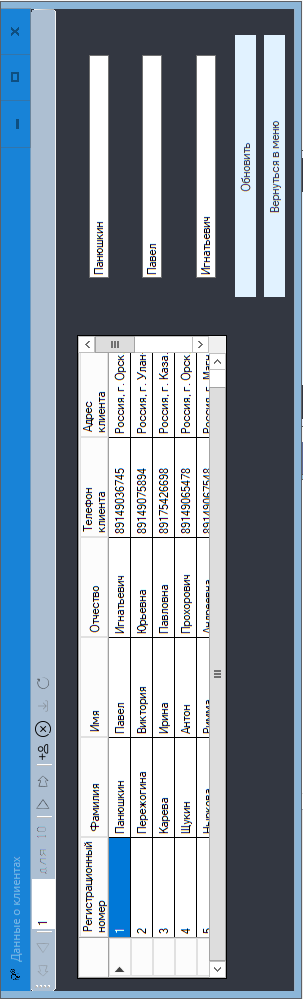


Рисунок 19 – Форма “Данные о клиентах”

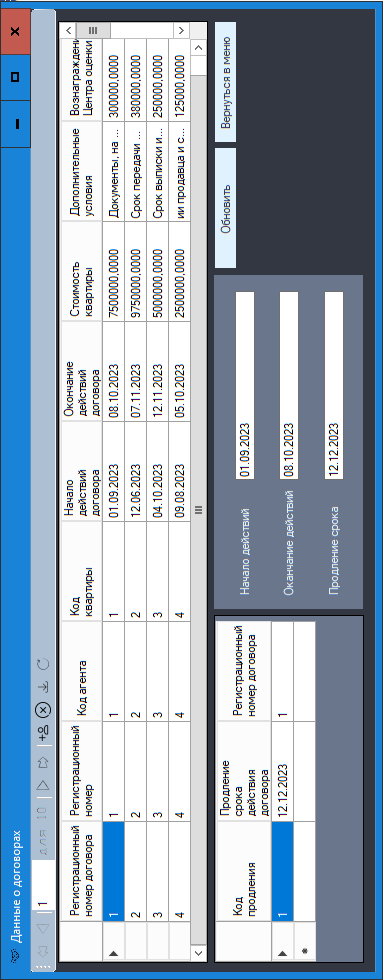


Рисунок 20 – Форма “Данные о договорах”

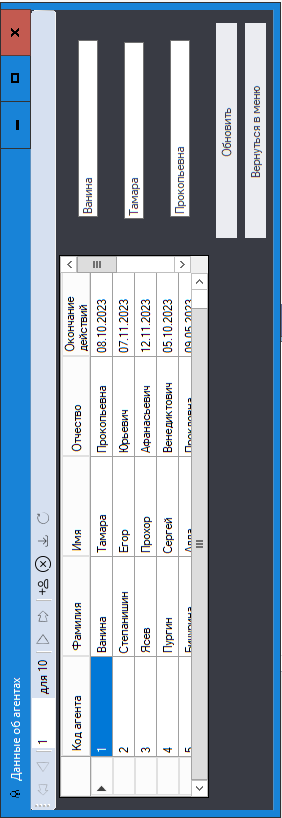


Рисунок 21 – Форма “Данные об агентах”

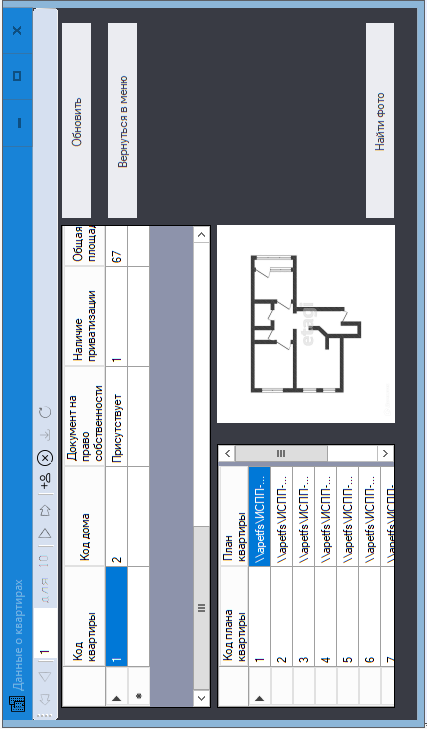


Рисунок 22 – Форма “Данные о квартирах”

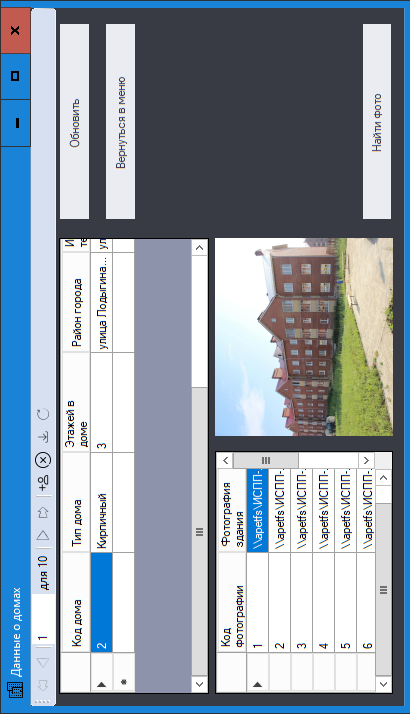


Рисунок 23 – Форма “Данные о домах”

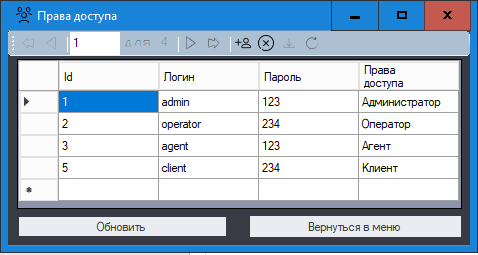


Рисунок 24 – Форма “Права доступа”

ПРИЛОЖЕНИЕ Б

Листинг кода формы “Авторизация”:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.IO;

using System.Drawing.Imaging;

namespace УП\_МДК\_01\_01

{

public partial class Form2 : Form

{

static public string loginActive;

static public string whois;

public Form2()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button3\_Click\_1(object sender, EventArgs e)

{

this.Hide();

Form3 UsrFrm = new Form3();

UsrFrm.Show();

Form3 form3 = new Form3();

form3.load();

}

int j = 0;

private void button1\_Click\_1(object sender, EventArgs e)

{

SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog =уП01\_ИСПП5\_Работягова\_АА ; Integrated Security = True");

sqlConnect.Open();

SqlDataAdapter da = new SqlDataAdapter("select \* from Пользователи", sqlConnect);

DataTable dt = new DataTable();

da.Fill(dt);

Boolean flag = false;

for (int i = 0; i < dt.Rows.Count; i++)

if ((dt.Rows[i]["Логин"].ToString() == textBox1.Text) && (dt.Rows[i]["Пароль"].ToString() == textBox2.Text)) { j = i; flag = true; }

if (flag==true)

{

switch (dt.Rows[j]["Права доступа"].ToString())

{

case null:

{

MessageBox.Show("Такого аккаунта не существует", "Проверьте данные и попоробуйте снова", MessageBoxButtons.OK, MessageBoxIcon.Warning);

break;

}

case "Администратор":

{

this.Hide();

Form5 form5 = new Form5();

form5.Show();

form5.а("Администратор");

break;

}

case "Клиент":

{

this.Hide();

Form10 form10 = new Form10();

form10.Show();

break;

}

case "Агент":

{

this.Hide();

Form11 form11 = new Form11();

form11.Show();

break;

}

case "Оператор":

{

this.Hide();

Form5 form5 = new Form5();

form5.Show();

form5.а("Оператор");

break;

}

}

}

else

{

MessageBox.Show("Заполните все обязательные поля", "Заполнение полей", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void textBox1\_KeyPress\_1(object sender, KeyPressEventArgs e)

{

textBox1.MaxLength = 30;

if (char.IsDigit(e.KeyChar))

{

e.Handled = true;

MessageBox.Show("Поле не может содержать цифры!", "Ошибка");

}

}

private void textBox2\_KeyPress\_1(object sender, KeyPressEventArgs e)

{

textBox2.MaxLength = 15;

}

private void button2\_Click\_1(object sender, EventArgs e)

{

this.Hide();

Form13 form13 = new Form13();

form13.Show();

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked)

{

textBox2.UseSystemPasswordChar = false;

}

else

{

textBox2.UseSystemPasswordChar = true;

}

}

}

}

Листинг кода формы “Регистрация”:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace УП\_МДК\_01\_01

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

Form2 form2 = new Form2();

form2.Show();

}

private void Form3\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи);

}

private void button1\_Click(object sender, EventArgs e)

{

SqlConnection sqlConnect = new SqlConnection("Data Source=sql;Initial Catalog = уП01\_ИСПП5\_Работягова\_АА; Integrated Security = True");

sqlConnect.Open();

DataTable dt = new DataTable();

DataTable dta = new DataTable();

SqlDataAdapter daa = new SqlDataAdapter($"select \* from Пользователи", sqlConnect);

daa.Fill(dta);

string n = dta.Rows[dta.Rows.Count - 1]["Id"].ToString();

SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Пользователи (Id,Логин,Пароль) VALUES ('{int.Parse(n) + 1}','{textBox1.Text}','{textBox2.Text}');", sqlConnect);

da.Fill(dt);

пользователиTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи);

MessageBox.Show($"Регистрация пользователя с логином {textBox1.Text} выполнена.\nВы зарегистрированы как гость");

this.Hide();

Form2 form2 = new Form2();

form2.Show();

}

public void load()

{

this.пользователиTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи);

dataGridView1.Update();

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

textBox1.MaxLength = 30;

if (char.IsDigit(e.KeyChar))

{

e.Handled = true;

MessageBox.Show("Поле не может содержать цифры!", "Ошибка");

}

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

textBox2.MaxLength = 15;

}

}

}

Листинг кода формы “Меню администратора”:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace УП\_МДК\_01\_01

{

public partial class Form5 : Form

{

public Form5()

{

InitializeComponent();

}

string a;

public void а(string b)

{

if (b == "Администратор")

{

button7.Visible = true;

}

else

{

button7.Visible = false;

}

}

private void button1\_Click\_1(object sender, EventArgs e)

{

Hide();

Form1 form1 = new Form1();

form1.i("Администратор");

form1.d("Администратор");

form1.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

Hide();

Form4 form4 = new Form4();

form4.i("Администратор");

form4.d("Администратор");

form4.Show();

}

private void button6\_Click(object sender, EventArgs e)

{

Hide();

Form6 form6 = new Form6();

form6.i("Администратор");

form6.d("Администратор");

form6.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

Hide();

Form7 form7 = new Form7();

form7.i("Администратор");

form7.d("Администратор");

form7.Show();

}

private void button7\_Click(object sender, EventArgs e)

{

Hide();

Form9 form5 = new Form9();

form5.i("Администратор");

form5.Show();

}

private void button8\_Click(object sender, EventArgs e)

{

Close();

Form2 form5 = new Form2();

form5.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button4\_Click(object sender, EventArgs e)

{

Hide();

Form8 form8 = new Form8();

form8.i("Администратор");

form8.Show();

}

}

}

Листинг кода формы “Меню клиента”:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace УП\_МДК\_01\_01

{

public partial class Form10 : Form

{

public Form10()

{

InitializeComponent();

}

private void button6\_Click(object sender, EventArgs e)

{

Hide();

Form6 form6 = new Form6();

form6.i("Клиент");

form6.d("Клиент");

form6.Show();

}

private void button8\_Click(object sender, EventArgs e)

{

Close();

Form2 form10 = new Form2();

form10.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

Hide();

Form4 form4 = new Form4();

form4.i("Клиент");

form4.d("Клиент");

form4.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

Hide();

Form7 form7 = new Form7();

form7.i("Клиент");

form7.d("Клиент");

form7.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

Hide();

Form8 form8 = new Form8();

form8.i("Клиент");

form8.d("Клиент");

form8.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

Листинг кода формы “Меню агента”:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace УП\_МДК\_01\_01

{

public partial class Form11 : Form

{

public Form11()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Hide();

Form1 form1 = new Form1();

form1.i("Агент");

form1.d("Агент");

form1.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

Hide();

Form4 form4 = new Form4();

form4.i("Агент");

form4.d("Агент");

form4.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

Hide();

Form7 form7 = new Form7();

form7.i("Агент");

form7.d("Агент");

form7.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

Hide();

Form8 form8 = new Form8();

form8.i("Агент");

form8.d("Агент");

form8.Show();

}

private void button8\_Click(object sender, EventArgs e)

{

Close();

Form2 form11 = new Form2();

form11.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

Листинг кода формы “Меню гостя”:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace УП\_МДК\_01\_01

{

public partial class Form13 : Form

{

public Form13()

{

InitializeComponent();

}

private void button6\_Click(object sender, EventArgs e)

{

Hide();

Form6 form6 = new Form6();

form6.i("Гость");

form6.d("Гость");

form6.Show();

}

private void button8\_Click(object sender, EventArgs e)

{

Close();

Form2 form13 = new Form2();

form13.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

Листинг кода формы “Данные о клиентах”:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace УП\_МДК\_01\_01

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Клиенты". При необходимости она может быть перемещена или удалена.

this.клиентыTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Клиенты);

}

private void button1\_Click(object sender, EventArgs e)

{

клиентыTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Клиенты);

}

public string dost;

public void i(string b)

{

dost = b;

}

private void button2\_Click(object sender, EventArgs e)

{

if (dost == "Администратор")

{

Hide();

Form5 form5 = new Form5();

form5.Show();

}

else

{

Hide();

Form11 form11 = new Form11();

form11.Show();

}

}

public void d(string b)

{

if (b == "Администратор")

{

bindingNavigatorAddNewItem.Visible = true;

bindingNavigatorDeleteItem.Visible = true;

}

else

{

bindingNavigatorAddNewItem.Visible = false;

bindingNavigatorDeleteItem.Visible = false;

}

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

клиентыTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Клиенты);

}

private void CancelButton\_Click(object sender, EventArgs e)

{

клиентыBindingSource1.CancelEdit();

}

}

}

Листинг кода формы “Данные о договорах”:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace УП\_МДК\_01\_01

{

public partial class Form4 : Form

{

public Form4()

{

InitializeComponent();

}

private void Form4\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Договоры". При необходимости она может быть перемещена или удалена.

this.договорыTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Договоры);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Продление". При необходимости она может быть перемещена или удалена.

this.продлениеTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Продление);

}

public string dost;

public void i(string b)

{

dost = b;

}

private void button2\_Click(object sender, EventArgs e)

{

if (dost == "Администратор")

{

Hide();

Form5 form5 = new Form5();

form5.Show();

}

else if (dost == "Клиент")

{

Hide();

Form10 form10 = new Form10();

form10.Show();

}

else

{

Hide();

Form11 form11 = new Form11();

form11.Show();

}

}

public void d(string b)

{

if (b == "Администратор")

{

bindingNavigatorAddNewItem.Visible = true;

bindingNavigatorDeleteItem.Visible = true;

}

else

{

bindingNavigatorAddNewItem.Visible = false;

bindingNavigatorDeleteItem.Visible = false;

}

}

private void button1\_Click(object sender, EventArgs e)

{

продлениеTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Продление);

договорыTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Договоры);

}

private void SaveButton\_Click(object sender, EventArgs e)

{

продлениеTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Продление);

договорыTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Договоры);

}

private void CancelButton\_Click(object sender, EventArgs e)

{

fKПродлениеДоговорыBindingSource.CancelEdit();

договорыBindingSource.CancelEdit();

}

}

}

Листинг кода формы “Данные об агентах”:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace УП\_МДК\_01\_01

{

public partial class Form6 : Form

{

public Form6()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

агентыTableAdapter2.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Агенты);

}

private void Form6\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Агенты". При необходимости она может быть перемещена или удалена.

this.агентыTableAdapter2.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Агенты);

}

public string dost;

public void i(string b)

{

dost = b;

}

private void button2\_Click(object sender, EventArgs e)

{

if (dost == "Администратор")

{

Hide();

Form5 form5 = new Form5();

form5.Show();

}

else if (dost == "Клиент")

{

Hide();

Form10 form10 = new Form10();

form10.Show();

}

else

{

Hide();

Form13 form13 = new Form13();

form13.Show();

}

}

private void SaveButton\_Click(object sender, EventArgs e)

{

агентыTableAdapter2.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Агенты);

}

private void CancelButton\_Click(object sender, EventArgs e)

{

агентыBindingSource.CancelEdit();

}

public void d(string b)

{

if (b == "Администратор")

{

bindingNavigatorAddNewItem.Visible = true;

bindingNavigatorDeleteItem.Visible = true;

}

else

{

bindingNavigatorAddNewItem.Visible = false;

bindingNavigatorDeleteItem.Visible = false;

}

}

private void bindingNavigatorAddNewItem\_Click(object sender, EventArgs e)

{

}

}

}

Листинг кода формы “Данные о квартирах”:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using System.Reflection.Emit;

namespace УП\_МДК\_01\_01

{

public partial class Form7 : Form

{

public Form7()

{

InitializeComponent();

}

private void Form7\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Квартиры". При необходимости она может быть перемещена или удалена.

this.квартирыTableAdapter2.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Квартиры);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet2.Квартиры". При необходимости она может быть перемещена или удалена.

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.План\_квартир". При необходимости она может быть перемещена или удалена.

this.план\_квартирTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.План\_квартир);

sqlConnect.Open();

}

private void button1\_Click(object sender, EventArgs e)

{

switch (c)

{

case 0:

план\_квартирTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.План\_квартир);

план\_квартирTableAdapter1.Fill(уП01\_ИСПП5\_Работягова\_ААDataSet3.План\_квартир); break;

case 1:

квартирыTableAdapter2.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Квартиры);

квартирыTableAdapter2.Fill(уП01\_ИСПП5\_Работягова\_ААDataSet3.Квартиры); break;

}

}

public string dost;

public void i(string b)

{

dost = b;

}

private void button2\_Click(object sender, EventArgs e)

{

if (dost == "Администратор")

{

Hide();

Form5 form5 = new Form5();

form5.Show();

}

else if (dost == "Клиент")

{

Hide();

Form10 form10 = new Form10();

form10.Show();

}

else

{

Hide();

Form11 form11 = new Form11();

form11.Show();

}

}

public void d(string b)

{

if (b == "Администратор")

{

bindingNavigatorAddNewItem.Visible = true;

bindingNavigatorDeleteItem.Visible = true;

}

else

{

bindingNavigatorAddNewItem.Visible = false;

bindingNavigatorDeleteItem.Visible = false;

}

}

private void pictureSet\_Click(object sender, EventArgs e)

{

pictureSet.SizeMode = PictureBoxSizeMode.StretchImage;

}

SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Работягова\_АА; Integrated Security = True");

private void button3\_Click(object sender, EventArgs e)

{

OpenFileDialog load = new OpenFileDialog();

if (load.ShowDialog() == DialogResult.OK)

{

pictureSet.ImageLocation = load.FileName;

SqlDataAdapter a = new SqlDataAdapter($"Update [План квартир] set [План квартиры] = '{load.FileName}' where [Код плана квартиры] = '{dataGridView2.Rows[dataGridView2.SelectedCells[0].RowIndex].Cells[0].Value}'", sqlConnect);

DataSet dt = new DataSet();

a.Fill(dt);

this.план\_квартирTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.План\_квартир);

}

}

string pic;

int c;

private void dataGridView2\_CellMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

try

{

pic = dataGridView2.Rows[dataGridView2.SelectedCells[0].RowIndex].Cells[1].Value.ToString();

pictureSet.Image = Image.FromFile(pic);

}

catch

{

}

}

private void bindingNavigatorAddNewItem\_Click(object sender, EventArgs e)

{

OpenFileDialog load = new OpenFileDialog();

if (load.ShowDialog() == DialogResult.OK)

{

pictureSet.ImageLocation = load.FileName;

SqlDataAdapter a = new SqlDataAdapter($"Insert into [План квартир] Values( '{load.FileName}')", sqlConnect);

DataSet dt = new DataSet();

a.Fill(dt);

this.план\_квартирTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.План\_квартир);

}

}

}

}

Листинг кода формы “Данные о домах”:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using System.Reflection.Emit;

namespace УП\_МДК\_01\_01

{

public partial class Form8 : Form

{

public Form8()

{

InitializeComponent();

}

private void Form8\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Дома". При необходимости она может быть перемещена или удалена.

this.домаTableAdapter2.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Дома);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Фотографии\_домов". При необходимости она может быть перемещена или удалена.

this.фотографии\_домовTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Фотографии\_домов);

}

int c;

private void button1\_Click(object sender, EventArgs e)

{

switch (c)

{

case 0:

фотографии\_домовTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Фотографии\_домов);

фотографии\_домовTableAdapter1.Fill(уП01\_ИСПП5\_Работягова\_ААDataSet3.Фотографии\_домов); break;

case 1:

домаTableAdapter2.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Дома);

домаTableAdapter2.Fill(уП01\_ИСПП5\_Работягова\_ААDataSet3.Дома); break;

}

}

public string dost;

public void i(string b)

{

dost = b;

}

private void button2\_Click(object sender, EventArgs e)

{

if (dost == "Администратор")

{

Hide();

Form5 form5 = new Form5();

form5.Show();

}

else if (dost == "Клиент")

{

Hide();

Form10 form10 = new Form10();

form10.Show();

}

else

{

Hide();

Form11 form11 = new Form11();

form11.Show();

}

}

public void d(string b)

{

if (b == "Администратор")

{

bindingNavigatorAddNewItem.Visible = true;

bindingNavigatorDeleteItem.Visible = true;

}

else

{

bindingNavigatorAddNewItem.Visible = false;

bindingNavigatorDeleteItem.Visible = false;

}

}

private void pictureSet\_Click(object sender, EventArgs e)

{

pictureSet.SizeMode = PictureBoxSizeMode.StretchImage;

}

SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Работягова\_АА; Integrated Security = True");

private void bindingNavigatorAddNewItem\_Click(object sender, EventArgs e)

{

OpenFileDialog load = new OpenFileDialog();

if (load.ShowDialog() == DialogResult.OK)

{

pictureSet.ImageLocation = load.FileName;

SqlDataAdapter a = new SqlDataAdapter($"Insert into [Фотографии домов] Values( '{load.FileName}')", sqlConnect);

DataSet dt = new DataSet();

a.Fill(dt);

this.фотографии\_домовTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Фотографии\_домов);

}

}

string pic;

private void dataGridView2\_CellMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

try

{

pic = dataGridView2.Rows[dataGridView2.SelectedCells[0].RowIndex].Cells[1].Value.ToString();

pictureSet.Image = Image.FromFile(pic);

}

catch

{

}

}

private void button3\_Click(object sender, EventArgs e)

{

OpenFileDialog load = new OpenFileDialog();

if (load.ShowDialog() == DialogResult.OK)

{

pictureSet.ImageLocation = load.FileName;

SqlDataAdapter a = new SqlDataAdapter($"Update [Фотографии домов] set [Фотография здания] = '{load.FileName}' where [Код фотографии] = '{dataGridView2.Rows[dataGridView2.SelectedCells[0].RowIndex].Cells[0].Value}'", sqlConnect);

DataSet dt = new DataSet();

a.Fill(dt);

this.фотографии\_домовTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Фотографии\_домов);

}

}

}

}

Листинг кода формы “Права доступа”

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace УП\_МДК\_01\_01

{

public partial class Form9 : Form

{

public Form9()

{

InitializeComponent();

}

private void Form9\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter1.Fill(this.уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи);

}

private void button1\_Click(object sender, EventArgs e)

{

пользователиTableAdapter1.Update(уП01\_ИСПП5\_Работягова\_ААDataSet3.Пользователи);

}

public string dost;

public void i(string b)

{

dost = b;

}

private void button2\_Click(object sender, EventArgs e)

{

if (dost == "Администратор")

{

Hide();

Form5 form5 = new Form5();

form5.Show();

}

}

}

}