МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное бюджетное учреждение

"Слободской колледж педагогики и социальных отношений"

**КУРСОВОЙ ПРОЕКТ**

по ПМ 01 «Разработка программных модулей» на тему:

**Разработка программного модуля для автоматизации работы бизнес-процессов агентства недвижимости**

Выполнил: Лутошкин Андрей Юрьевич

Специальность 09.02.07

Информационные системы и программирование

Группа 21П-1

Форма обучения: очная

Руководитель: Калинин Арсений Олегович

Дата защиты курсового проекта:

Председатель ПЦК:

Оценка за защиту курсового проекта:

Слободской

2024

**ОГЛАВЛЕНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc185436616)

[**1.** **АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ** 5](#_Toc185436617)

[**2.** **РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ** 12](#_Toc185436618)

[**3.** **ОПИСАНИЕ АЛГОРИТМОВ И ФУНКЦИОНАЛА ПРОГРАММЫ** 18](#_Toc185436619)

[**4.** **ТЕСТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ** 21](#_Toc185436620)

[**5.** **РУКОВОДСТВО ОПЕРАТОРА** 26](#_Toc185436621)

[**ЗАКЛЮЧЕНИЕ** 35](#_Toc185436622)

[**СПИСОК ЛИТЕРАТУРЫ** 36](#_Toc185436623)

[**ПРИЛОЖЕНИЕ** 38](#_Toc185436624)

[**Приложение 1** 39](#_Toc185436625)

[**Приложение 2** 41](#_Toc185436626)

[**Приложение 3** 53](#_Toc185436627)

# **ВВЕДЕНИЕ**

В условиях стремительного развития информационных технологий и возрастающей конкуренции, агентства недвижимости сталкиваются с необходимостью оптимизации своей деятельности и повышения качества обслуживания клиентов. Традиционные методы работы, основанные на ручном ведении документации, личных встречах и телефонных переговорах, уже не отвечают современным требованиям рынка. Поиск подходящих объектов, организация просмотров, оформление сделок и взаимодействие с клиентами занимают значительное время и требуют существенных затрат ресурсов. Зачастую, рутинные операции, такие как ввод данных, составление договоров, контроль платежей и формирование отчетов, отвлекают агентов от более важных задач, связанных с привлечением новых клиентов и развитием бизнеса. Неэффективное управление информацией может привести к потере потенциальных сделок, снижению лояльности клиентов и ухудшению репутации агентства. Автоматизация бизнес-процессов становится необходимым условием для успешной работы агентства недвижимости в современных условиях. Внедрение специализированного программного обеспечения позволяет решить ряд ключевых задач:

* **Оптимизация работы с объектами недвижимости:** быстрый поиск и подбор объектов по заданным критериям, удобное хранение и обновление информации, автоматическое формирование описаний и презентаций.
* **Улучшение взаимодействия с клиентами:** автоматизация обработки заявок, планирование и учет просмотров, персонализированные коммуникации, быстрый доступ к истории взаимодействия.
* **Повышение эффективности сделок:** автоматизация подготовки договоров и другой документации, контроль сроков и этапов сделки, учет финансовых операций.
* **Упрощение отчетности и аналитики:** автоматическое формирование отчетов по различным параметрам, анализ эффективности работы агентов, мониторинг ключевых показателей производительности.

**Цель данной курсового проекта** – разработка программного модуля для автоматизации ключевых бизнес-процессов агентства недвижимости, ориентированного на повышение эффективности работы и улучшение качества обслуживания клиентов.

**Объект исследования**: бизнес-процессы агентства недвижимости, связанные с управлением объектами, взаимодействием с клиентами и сопровождением сделок.

**Предмет исследования:**

* программный модуль для автоматизации бизнес-процессов агентства недвижимости, реализованный на платформе 1С: Предприятие.

**Методы исследования:** системный анализ, функциональное моделирование, объектно-ориентированное программирование.

В работе использовалась платформа 1С: Предприятие, которая широко распространена в России и обладает мощными возможностями для создания бизнес-приложений. Выбор данной платформы обусловлен ее доступностью, гибкостью и наличием широкого спектра инструментов для разработки учетных систем.

Информационную базу исследования составили официальные нормативно-правовые источники, регулирующие деятельность библиотек, научные статьи и публикации, посвященные проблемам автоматизации библиотек, методические рекомендации по учету библиотечных фондов, официальная документация по платформе 1С: Предприятие.

# **АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

**1.1 Общая характеристика предметной области:**

Агентство недвижимости осуществляет деятельность на рынке недвижимости, предоставляя посреднические услуги между продавцами/арендодателями и покупателями/арендаторами. Ключевыми аспектами работы агентства являются: эффективное управление информацией об объектах недвижимости, взаимодействие с клиентами (контрагентами) и сотрудниками, а также юридическое сопровождение сделок. Автоматизация этих процессов позволяет повысить эффективность работы агентства, минимизировать риски и улучшить качество обслуживания.

**1.2 Процессы агентства недвижимости:**

В соответствии с техническим заданием, ключевыми процессами, подлежащими автоматизации, являются:

* **Учет базы недвижимости (здания):**
  + Добавление здания: ввод информации о типе объекта, статусе новостройки, ответственном сотруднике, адресе, технических характеристиках (тип здания, этажность, площадь участка).
  + Редактирование данных о здании: изменение существующей информации.
  + Удаление здания: снятие здания с учета.
* **Учет базы объектов (квартиры, помещения и т.д.):**
  + Добавление объекта: ввод детальной информации об объекте, включая тип операции (продажа/аренда), связь со зданием, характеристики объекта (площадь, ремонт, коммуникации), правовой статус, информацию о собственнике и агенте.
  + Редактирование данных об объекте: изменение существующей информации.
  + Удаление объекта: снятие объекта с учета.
* **Учет базы сотрудников:**
  + Добавление сотрудника: ввод персональных данных, контактной информации, данных о должности и образовании.
  + Редактирование данных о сотруднике: изменение существующей информации.
  + Удаление сотрудника: снятие сотрудника с учета.
* **Учет базы контрагентов (клиенты):**
  + Добавление контрагента: ввод персональных данных, контактной информации.
  + Редактирование данных о контрагенте: изменение существующей информации.
  + Удаление контрагента: снятие контрагента с учета (не рекомендуется, лучше отмечать, как неактивного).
* **Выполнение операций (сделки):**
  + Формирование заявки: создание заявки на оказание услуги с указанием типа операции, объекта, цены, агента.
  + Формирование договора: создание договора на основании заявки с указанием всех необходимых реквизитов и возможностью печати.

Реализована ER-диаграмма предметной области на рисунке 1 и рисунке 2.

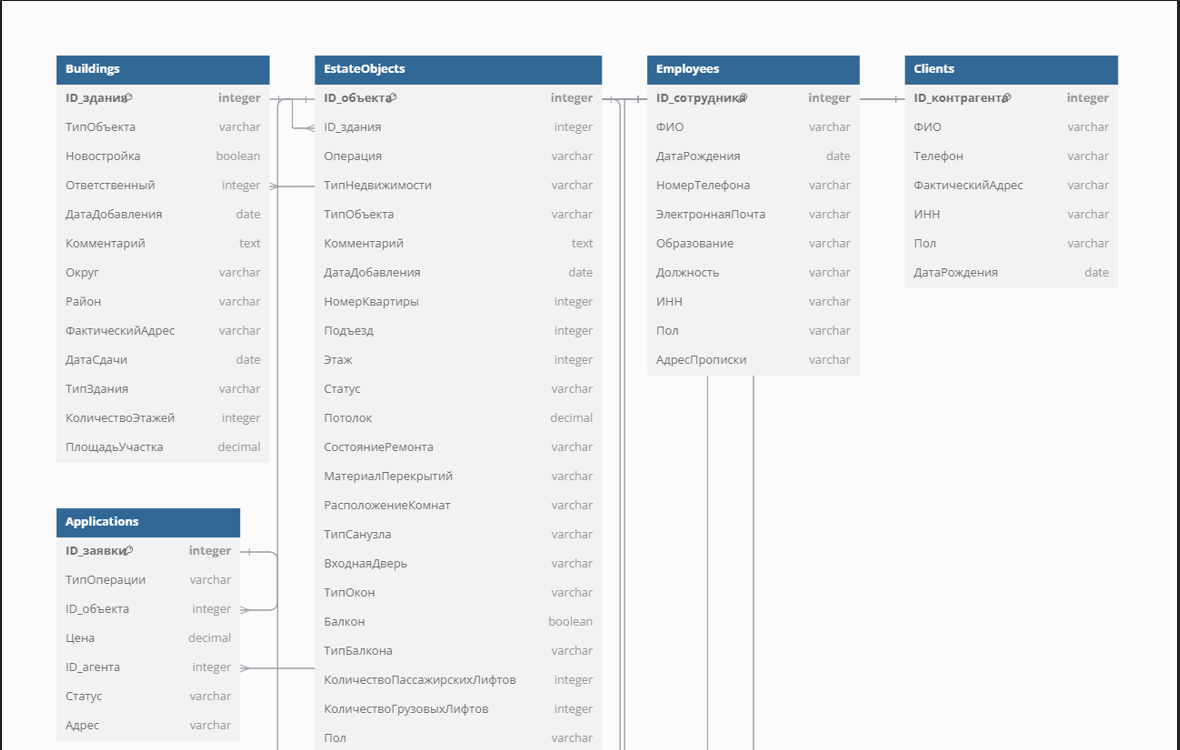
****

Рисунок 1 - ER-диаграмма предметной области

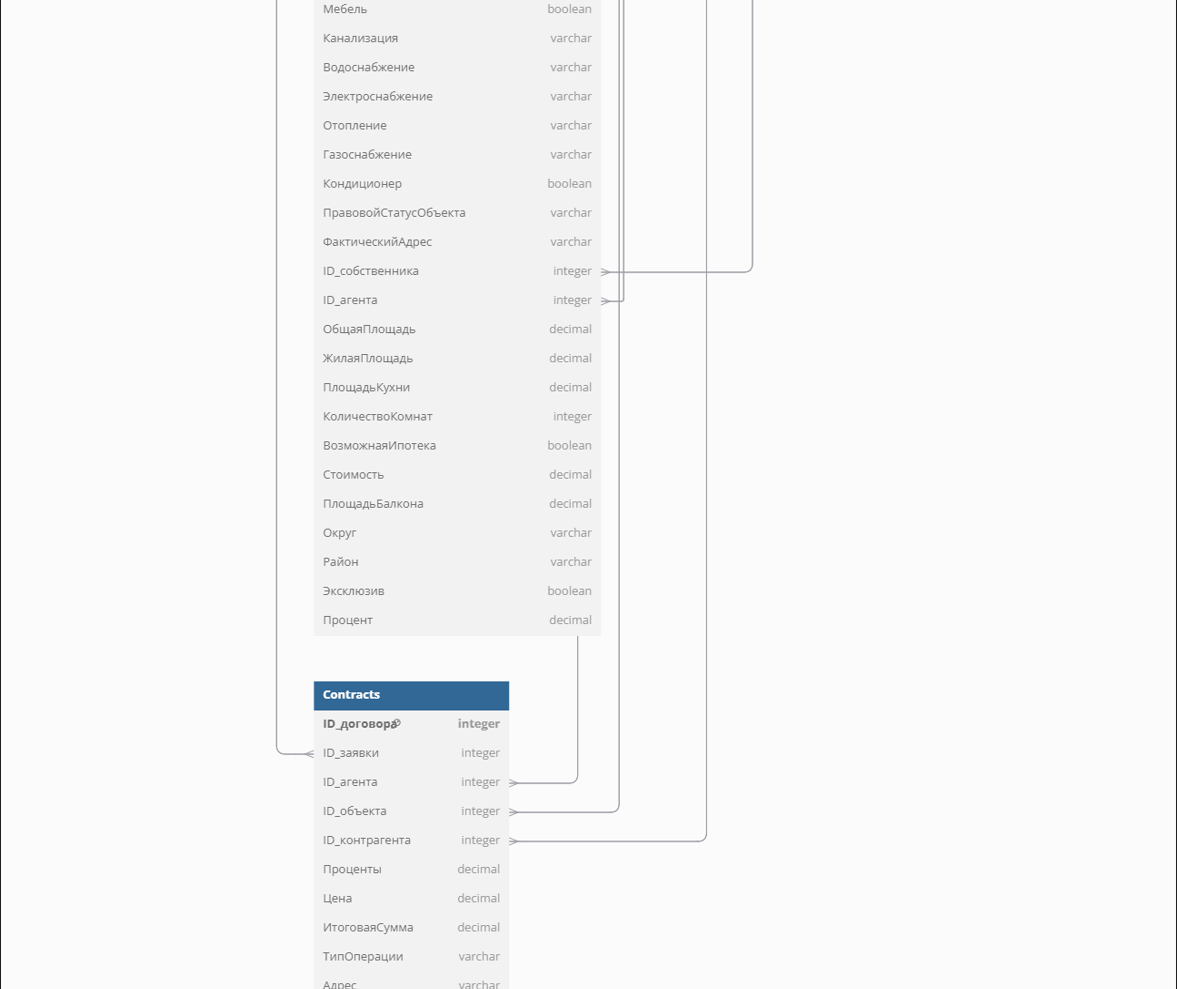
****

Рисунок 2 – ERD-диаграмма предметной области

**1.3 Существующие методы автоматизации работы бизнес-процессов агентства недвижимости:**

Сегодня на рынке представлено множество решений для автоматизации работы агентств недвижимости, от специализированных программных продуктов до комплексных CRM-систем. Их можно классифицировать следующим образом:

**1.3.1 Специализированные CRM для недвижимости:**

* **Преимущества:** узкая специализация позволяет учесть специфику работы агентства недвижимости, предлагая функционал для управления объектами, клиентами, сделками, маркетингом и аналитикой. Часто интегрируются с порталами недвижимости.
* **Недостатки:** могут быть дороже универсальных CRM и иметь меньше возможностей для интеграции с другими системами.
* **Примеры:** Real Estate CRM, Propertybase, IXACT Contact.

**1.3.2 Универсальные CRM-системы:**

* **Преимущества:** широкий функционал, гибкость настройки, возможность интеграции с различными сервисами. Более доступная цена по сравнению со специализированными решениями.
* **Недостатки:** требуют настройки и адаптации под специфику работы агентства недвижимости. Могут содержать лишний функционал, который не используется.
* **Примеры:** Bitrix24, AmoCRM, Salesforce.

**1.3.3 Отраслевые ERP-системы:**

* **Преимущества:** позволяют автоматизировать не только операционные процессы (работа с клиентами, сделки), но и управленческий учет, финансы, кадры. Обеспечивают комплексный подход к управлению агентством.
* **Недостатки:** Высокая стоимость внедрения и обслуживания, сложность настройки и обучения персонала.
* **Примеры:** SAP Real Estate Management, Microsoft Dynamics 365.

**1.3.4 Индивидуальная разработка:**

* **Преимущества:** позволяет создать систему, полностью соответствующую потребностям конкретного агентства. Максимальная гибкость и возможность реализации любого функционала.
* **Недостатки:** Высокая стоимость разработки, длительные сроки, необходимость в квалифицированных специалистах.

**1.3.5 Комбинация различных подходов:** например, использование специализированной CRM для работы с клиентами и сделками, и отдельной системы для учета финансов.

**1.4 Проблемы ручного учета:**

Ручной учет в агентстве недвижимости, несмотря на кажущуюся простоту, создает ряд серьезных проблем, которые негативно сказываются на эффективности работы и качестве обслуживания клиентов. Среди основных проблем можно выделить:

* **Трудоемкость:** ручной ввод данных отнимает много времени, которое можно потратить на привлечение клиентов и заключение сделок.
* **Риск ошибок:** высокая вероятность опечаток и неточностей, ведущих к финансовым и юридическим проблемам.
* **Сложность анализа:** трудно искать информацию и анализировать данные для принятия решений, формирование отчетов трудоемко.
* **Проблемы с контролем:** сложно отслеживать статус сделок, историю взаимодействия с клиентами и изменения данных.
* **Дублирование информации:** риск путаницы и ошибок из-за дублирования данных.
* **Низкая безопасность данных:** бумажное хранение уязвимо для несанкционированного доступа и утечек.
* **Ограниченная масштабируемость:** ручной учет не эффективен при росте агентства и объема данных.

**1.5 Обоснование необходимости автоматизации:**

Автоматизация бизнес-процессов агентства недвижимости решает ряд проблем ручного учета, повышая эффективность и конкурентоспособность. Основные преимущества:

* **Повышение эффективности агентов:** освобождение от рутины (ввод данных, отчеты) для фокуса на привлечении клиентов и сделках.
* **Улучшение обслуживания клиентов:** быстрый доступ к информации, автоматическое формирование документов, оперативный сервис.
* **Снижение рисков и ошибок:** минимизация ошибок ручного ввода, предотвращение финансовых и юридических проблем.
* **Улучшение контроля и управления:** эффективный контроль бизнес-процессов, отслеживание сделок, анализ эффективности агентов.
* **Повышение безопасности данных:** разграничение прав доступа, защита конфиденциальной информации.
* **Масштабируемость бизнеса:** адаптация к росту агентства и объема данных.
* **Конкурентное преимущество:** оптимизация деятельности, повышение прибыльности и укрепление позиций на рынке.

**Обзор аналогов.**

**Real Estate CRM:** Специализированная CRM-система, предназначенная для автоматизации работы агентств недвижимости. Включает в себя функционал для управления объектами недвижимости (добавление, редактирование, поиск), работы с клиентами (ведение базы контактов, история взаимодействий), сопровождения сделок (отслеживание этапов сделки, формирование документов), а также инструменты маркетинга и аналитики.

**Плюсы**:

* Узкая специализация и ориентация на рынок недвижимости.
* Широкий набор специализированных функций.
* Интеграция с порталами недвижимости.

**Минусы**:

* Стоимость может быть высокой для небольших агентств.
* Ограниченные возможности интеграции с другими системами.

**Bitrix24:** Универсальная CRM-система, которая может быть адаптирована для использования агентствами недвижимости. Предлагает широкий набор инструментов для управления продажами, маркетингом, проектами и взаимодействием с клиентами.

**Плюсы:**

* Гибкость настройки и широкий функционал.
* Наличие бесплатного тарифа.
* Большое количество интеграций.

**Минусы:**

* Требует настройки и доработки для полного соответствия потребностям агентства недвижимости.

**1.6 Роли**

Реализована UML диаграмма вариантов использования агенства недвижимости на рисунке 3.

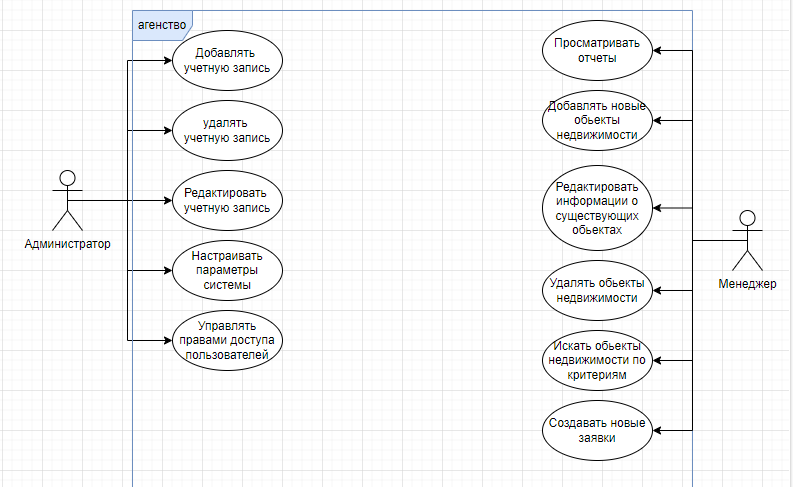


Рисунок 3 - UML диаграмма вариантов использования

**Роль:** Администратор

* Добавление новых пользователей
* Редактирование информации о пользователях
* Удаление учетных записей пользователей
* Настройка параметров системы
* Управление правами доступа пользователей

**Роль:** Менеджер

* Просмотр отчетов
* Добавление новых объектов недвижимости
* Редактирование информации о существующих объектах
* Поиск объектов по различным критериям

# **РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ**

**1. Введение.**

В данном разделе представлено техническое задание (ТЗ) на разработку программного модуля для автоматизации бизнес-процессов агентства недвижимости [1]. Техническое задание содержит подробное описание требований к модулю, его функциональным возможностям, а также технических характеристик. Техническое задание является основным документом, регламентирующим процесс разработки программного обеспечения и определяющим его соответствие потребностям заказчика.

**2. Назначение и область применения.**

* **Назначение:** разрабатываемый программный модуль предназначен для автоматизации бизнес-процессов агентства недвижимости, предоставляя удобный инструмент для хранения справочной информации, управления заявками и договорами и оптимизации рабочих процессов.
* **Область применения:** модуль может быть использован в различных агентствах недвижимости, в которых необходимо хранить информацию, а также управлять договорами и заявками.

**3. Функциональные требования.**

Программный модуль должен обеспечивать выполнение следующих функций:

* **Учет базы о недвижимости.**
  + **Добавление нового здания в базу данных:** модуль должен позволять добавлять новые здания в базу данных с указанием всех необходимых реквизитов, таких как тип объекта, является ли новостройкой, ответственный, дата добавления, комментарий, округ, район, фактический адрес, дата сдачи, тип здания, количество этажей, площадь участка. Форма для добавления здания показана на рисунке 4.

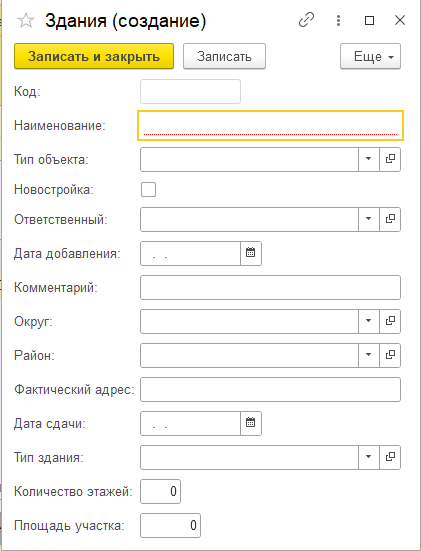


Рисунок 4 – Форма добавления здания

* + Редактирование данных о здании: должна быть предусмотрена возможность изменения данных о здании, если это необходимо.
  + **Удаление здания из базы данных:** в случае сноса здания или переквалификации земли, модуль должен позволять удалять его из базы данных.
* **Учет базы об объектах.**
  + **Добавление нового объекта в базу данных:** модуль должен позволять добавлять новые объекты в базу данных с указанием всех необходимых реквизитов, таких операция, здание, тип недвижимости, тип объекта, комментарий, дата добавления, номер квартиры, подъезд, этаж, статус, потолок, состояние ремонта, материал перекрытий, расположение комнат, тип санузла, входная дверь, тип окон, балкон, тип балкона, количество лифтов, пол, мебель, канализация, водоснабжение, электроснабжение, отопление, газоснабжение, кондиционер, правовой статус объекта, фактический адрес, собственник, агент, общая площадь . Форма для добавления объекта показана на рисунке 5.

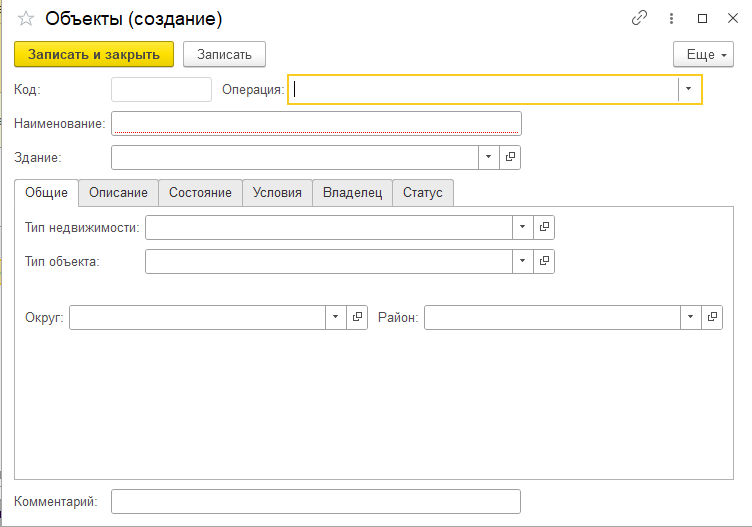


Рисунок 5 - Форма добавления обьекта

* + Редактирование данных об объекте: должна быть предусмотрена возможность изменения данных об объекте, если это необходимо.
  + **Удаление объекта из базы данных:** в случаи переквалификации объекта, модуль должен позволять удалять его из базы данных.
* **Учет базы о сотрудниках.**
  + **Добавление нового сотрудника в базу данных:** модуль должен позволять добавлять новых сотрудников в базу данных с указанием всех необходимых реквизитов, таких как ФИО, дата рождения, номер телефона, электронная почта, образование, должность, ИНН, адрес прописки. Форма для добавления сотрудника показана на рисунке 6.

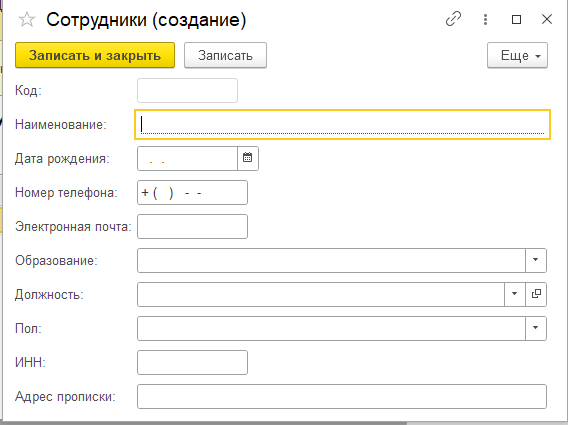


Рисунок 6 – Форма добавления сотрудника

* + Редактирование данных о сотруднике: должна быть предусмотрена возможность изменения данных об объекте, если это необходимо.
  + **Удаление сотрудника из базы данных:** в случае увольнения сотрудника, модуль должен позволять удалять его из базы данных.
* **Учет базы об контрагентах.**
  + **Добавление нового контрагента в базу данных:** модуль должен позволять добавлять новых контрагентов в базу данных с указанием всех необходимых реквизитов, таких как ФИО, номер телефона, фактический адрес, ИНН, пол, дата рождения. Форма для добавления контрагента показана на рисунке 5.
  + Редактирование данных об контрагенте: должна быть предусмотрена возможность изменения данных об объекте, если это необходимо.
  + **Удаление контрагенте из базы данных:** модуль должен позволять удалять его из базы данных.
* **Выполнение операция.**
  + **Формирование заявки на выполнение услуги:** модуль должен позволять создавать заявки на выполнение услуг с указанием типа операции, объекта, цены, агента, процентов, статуса, адреса.
  + **Формирование договора услуг:** при выполнении заявки на оказание услуг модуль должен позволять формировать договор услуг с указанием заявки, агента, объекта, процентов, контрагента, цены, итоговой суммы, типа операции, адреса, а также выводить печатную форму.
  + **Печать договора услуг:** модуль должен позволять печатать договор услуг по созданному макету.

**4. Требования к пользовательскому интерфейсу.**

* **Интерфейс программы должен быть интуитивно понятным и удобным в использовании:** разработка пользовательского интерфейса должна быть ориентирована на простоту и удобство использования, чтобы даже неподготовленный пользователь мог легко освоить работу с модулем.
* **Требования** к пользовательскому интерфейсу должны соответствовать принципам эргономичности и удобства использования.

**5. Требования к техническому обеспечению.**

* **Операционная система:** Windows 7/8/10/11.
* **Платформа разработки:** 1С: Предприятие 8.3.
* **База данных:** Встроенная база данных 1С: Предприятие.

**6. Требования к надежности и безопасности.**

* **Устойчивость к ошибкам:** программа должна быть устойчива к ошибкам пользователей и сбоям в работе системы, обеспечивая целостность данных и предотвращая потерю информации.
* **Резервное копирование:** должна быть предусмотрена система резервного копирования данных, позволяющая восстановить базу данных в случае сбоя или потери информации.
* **Авторизация и разграничение прав доступа:** доступ к программе должен быть ограничен системой авторизации, которая будет идентифицировать пользователя и предоставлять ему доступ только к тем функциям, которые соответствуют его роли.

**7. Стадии и этапы разработки.**

* **Анализ и проектирование:** 1 месяц.
* **Разработка программного кода:** 2 месяца.
* **Тестирование и отладка:** 1 месяц.
* **Внедрение и сопровождение:** по мере необходимости.

**Вывод:** разработанное техническое задание на создание программного модуля для автоматизации бизнес-процессов агентства недвижимости содержит подробное описание функциональных и технических требований к разрабатываемой системе. Данное ТЗ послужит основой для дальнейшей разработки, тестирования и внедрения программного модуля, обеспечивая его соответствие потребностям заказчика и эффективную автоматизацию бизнес-процессов агентства недвижимости

# **ОПИСАНИЕ АЛГОРИТМОВ И ФУНКЦИОНАЛА ПРОГРАММЫ**

Разработанный программный модуль для автоматизации бизнес-процессов агентства недвижимости реализован на платформе 1С: Предприятие 8.3 с использованием встроенного языка "1С" (Приложение 1). Модуль предоставляет удобный пользовательский интерфейс и включает набор объектов метаданных: справочники ("Здания", "Объекты", "Контрагенты", "Сотрудники"), документы ("Заявка", "Договор"), регистры, а также программные модули, определяющие логику работы системы. Алгорим работы программы представлен в виде блок-схемы рисунок 7.

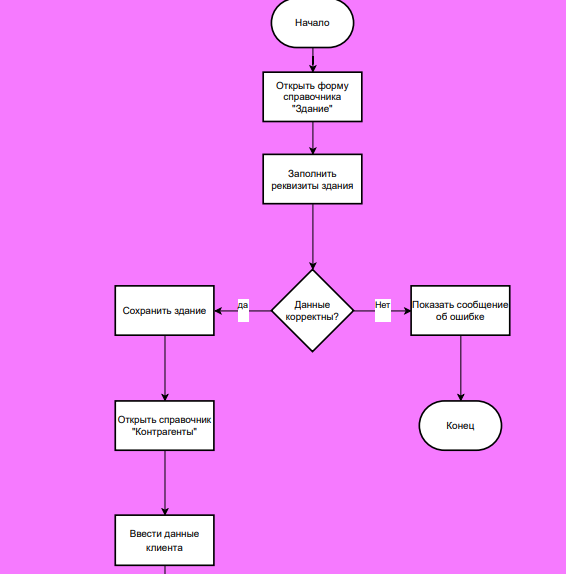


Рисунок 7 – Начало блок-Схемы

Продолжение блок схемы находится в приложении 1.

**Функциональные модули:**

Программа включает следующие функциональные модули (подсистемы):

* **Управление недвижимостью:** отвечает за хранение и обработку информации о зданиях и объектах недвижимости. Реализовано с помощью справочников "Здания" и "Объекты".
* **Работа с клиентами:** обеспечивает ведение базы клиентов (контрагентов) и истории взаимодействий с ними. Основан на справочнике "Контрагенты".
* **Управление сделками:** позволяет создавать заявки на покупку/аренду недвижимости, формировать договоры и отслеживать статус сделок. Использует документы "Заявка" и "Договор".
* **Формирование отчетов:** обеспечивает генерацию различных отчетов, необходимых для анализа деятельности агентства.

**Алгоритмы работы основных функций:**

Ниже представлены алгоритмы работы основных функций каждого модуля.

**Модуль "Управление недвижимостью":**

* **Добавление здания:**
  1. Пользователь открывает форму справочника "Здания" в режиме добавления.
  2. Пользователь заполняет реквизиты здания (наименование, адрес, тип здания и т.д.).
  3. Система проверяет корректность введенных данных.
  4. При успешной проверке система сохраняет новое здание в справочник.
  5. В случае ошибки система выдает сообщение об ошибке.
* **Редактирование здания:** аналогично добавлению, но с предварительным выбором редактируемого здания.
* **Удаление здания:**
  1. Пользователь выбирает здание для удаления.
  2. Система проверяет, нет ли связанных с этим зданием объектов недвижимости.
  3. Если связанных объектов нет, система удаляет здание.
  4. Если есть связанные объекты, система выдает сообщение об ошибке и предлагает сначала удалить связанные объекты.
* **Добавление объекта, Редактирование объекта, Удаление объекта:** Алгоритмы аналогичны работе со зданиями, с учетом связи объекта со зданием.

**Модуль "Работа с клиентами":**

* **Добавление клиента:** (аналогично добавлению здания)
* **Редактирование клиента:** (аналогично редактированию здания)
* **Удаление клиента:** (аналогично удалению здания, но с проверкой на наличие связанных сделок)

**Модуль "Управление сделками":**

* **Создание заявки:** (аналогично добавлению здания, с выбором клиента и объекта)
* **Формирование договора:**
  1. Пользователь выбирает заявку.
  2. Система создает новый документ "Договор" на основе данных заявки.
  3. Система заполняет реквизиты договора (клиент, объект, стоимость, сроки и т.д.).
  4. Пользователь проверяет и подписывает договор.
* **Отслеживание статуса сделки:** Система позволяет изменять статус сделки (например, "Новая", "В обработке", "Завершена").

**Модуль "Формирование отчетов":**

* Система предоставляет набор стандартных отчетов (например, "Отчет по продажам", "Отчет по клиентам").
* Пользователь выбирает тип отчета и задает необходимые параметры.
* Система формирует отчет на основе данных из базы данных.

# **ТЕСТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ**

Для проведения тестирования программного модуля была разработана серия тестов, направленных на выявление ошибок и проверку функциональности программы. Тестирование проводилось в процессе разработки, и в результате были обнаружены несколько проблем, которые были успешно решены.

Было произведено тестирование добавление контрагента в таблице 1.

Таблица 1

|  |  |
| --- | --- |
| Поле | Значение |
| TestCase # | 1 |
| Приоритет | Высокий |
| Название | Проверка добавления контрагента |
| Резюме | Проверка корректности добавления нового контрагента в справочник |
| Шаги | 1.Перейти в справочник " контрагент ".  2.Добавить нового контрагента.  3.Заполнить все обязательные поля  4.Сохранить. |
| Данные | Наименование: " Лутошкин Андрей ",  Телефон: ”+7(999)123 22-33”  Фактический Адрес:” Город Слободской улица Ленина”  ИНН: ”123445657564”  Пол:”Мужской”  Дата Рождения:”01.05.2000” |
| Ожидаемый | Контрагент успешно добавлен в справочник. |
| Фактический | Контрагент успешно добавлен в справочник. |

Продолжение таблицы 1

|  |  |
| --- | --- |
| Предпосылки | Справочник "Контрагент " существует и доступен для записи. |
| Постусловия | Контрагент добавлен в справочник. |
| Статус | Pass |
| Комментарии | |  | | --- | | Тест проверяет базовую функциональность добавления нового контрагента. | |

Таблица 2

|  |  |
| --- | --- |
| Поле | Значение |
| TestCase # | 2 |
| Приоритет | Высокий |
| Название | Генерация отчета |
| Резюме | Проверка корректности формирования отчета |
| Шаги | 1.Перейти в раздел "Отчеты".  2.Выбрать отчет "Обьекты".  4.Сформировать отчет. |
| Данные | - |
| Ожидаемый | Сформирован отчет, содержащий информацию о обьектах которые есть на данный момент у агенства |
| Фактический | Сформирован отчет, содержащий информацию о обьектах которые есть на данный момент у агенства |
| Предпосылки | В базе данных есть записи о обьектах |
| Постусловия | Отчет сформирован и сохранен. |
| Статус | Pass | |
| Комментарии | |  | | --- | | Тест проверяет корректность формирования отчета | | |

Таблица 3

|  |  |
| --- | --- |
| Поле | Значение |
| TestCase # | 3 |
| Приоритет | Высокий |
| Название | Редактирование объекта |
| Резюме | Проверка корректности редактирования данных существующего объекта недвижимости. |
| Шаги | 1. Найти объект в списке. 2. Открыть форму редактирования. 3. Изменить цену. 4. Нажать "Сохранить". |
| Данные | Новая цена: 1 200 000 руб. |
| Ожидаемый | Данные объекта успешно изменены. |
| Фактический | Данные объекта успешно изменены. |
| Предпосылки | В базе данных есть объект для редактирования. |
| Постусловия | Изменения отображаются в списке объектов и карточке объекта. |
| Статус | Pass |
| Комментарии | Тест проверяет работоспособность функции редактирования |

Таблица 4

|  |  |
| --- | --- |
| Поле | Значение |
| TestCase # | 4 |
| Приоритет | Высокий |
| Название | Удаление объекта |

Продолжение таблицы 4

|  |  |
| --- | --- |
| Резюме | Проверка корректности удаления объекта недвижимости из базы данных. |
| Шаги | 1. Найти объект в списке. 2. Выбрать "Удалить". 3. Подтвердить удаление. |
| Данные |  |
| Ожидаемый | Объект успешно удален из базы данных. |
| Фактический | Объект успешно удален из базы данных. |
| Предпосылки | В базе данных есть объект для удаления. |
| Постусловия | Объект больше не отображается в списке. |
| Статус | Pass |
| Комментарии | Тест проверяет работоспособность функции удаления |

Таблица 5

|  |  |
| --- | --- |
| Поле | Значение |
| TestCase # | 5 |
| Приоритет | Высокий |
| Название | Создание договора |
| Резюме | Проверка корректности формирования договора на основе заявки. |
| Шаги | 1. Открыть заявку. 2. Нажать "Создать договор". 3. Проверить данные договора. 4. Нажать "Сохранить". |
| Данные | - |
| Ожидаемый | Договор успешно создан и содержит корректные данные из заявки (клиент, объект, тип услуги, цена). |

Продолжение таблицы 5

|  |  |
| --- | --- |
| Фактический | Договор успешно создан и содержит корректные данные из заявки (клиент, объект, тип услуги, цена). |
| Предпосылки | Существует заявка на услугу. |
| Постусловия | Договор отображается в списке договоров. |
| Статус | Pass |
| Комментарии | Тест проверяет функцию создания договора |

* 1. **РУКОВОДСТВО ОПЕРАТОРА**

**1. Введение**

Данное руководство предназначено для операторов, работающих с программным модулем для автоматизации бизнес-процессов агентства недвижимости, разработанным на платформе 1С: Предприятие 8.3. Руководство описывает основные функции модуля и порядок работы с ним.

**2. Функциональное назначение программы**

Программный модуль предназначен для автоматизации следующих бизнес-процессов агентства недвижимости:

* Учет базы недвижимости (здания и объекты)
  + Добавление здания
  + Редактирование здания
  + Удаление здания
  + Добавление объекта
  + Редактирование объекта
  + Удаление объекта
* Учет базы сотрудников
  + Добавление сотрудника
  + Редактирование сотрудника
  + Удаление сотрудника
* Учет базы контрагентов
  + Добавление контрагента
  + Редактирование контрагента
  + Удаление контрагента
* Выполнение операций (формирование заявок и договоров)
  + Формирование заявки
  + Формирование договора
  + Печать договора

**3. Минимальные системные требования**

* Операционная система: Windows 7/8/10/11
* Платформа: 1С:Предприятие 8.3
* База данных: Встроенная база данных 1С:Предприятие

**4. Запуск программы**

Для запуска программы необходимо открыть 1С:Предприятие и выбрать соответствующую базу данных. После запуска программы появится главное окно приложения (Рисунок8).

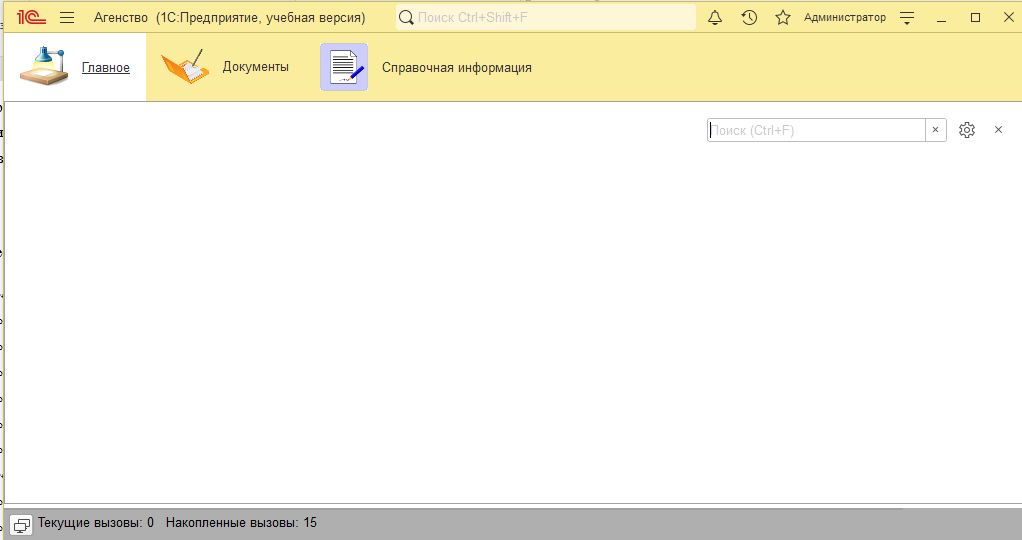


Рисунок 8 - Стартовое окно программы

**5. Работа с модулем**

**5.1. Учет базы недвижимости**

* **Добавление здания:** В разделе "Недвижимость" выбрать "Здания" и нажать кнопку "Добавить". Заполнить все необходимые поля в форме добавления здания (тип объекта, новостройка, ответственный, дата добавления, комментарий, округ, район, адрес, дата сдачи, тип здания, количество этажей, площадь участка) (Рисунок 9).

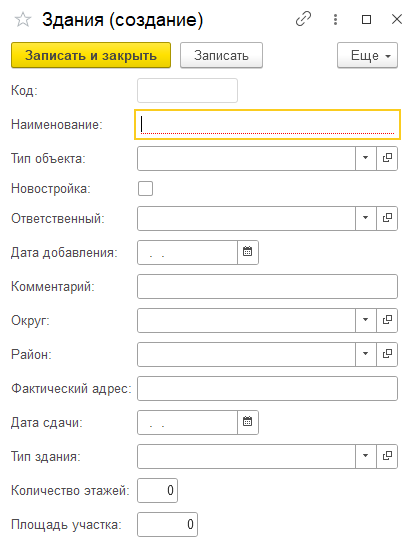


Рисунок 9 – Окно добавления здания

* **Редактирование здания:** Выбрать здание из списка и нажать кнопку "Изменить". Внести необходимые изменения и сохранить (Рисунок 10).

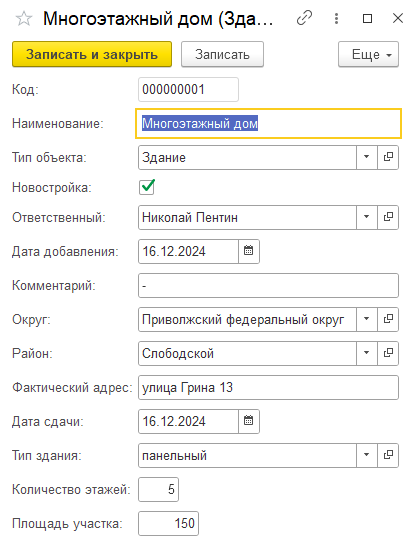


Рисунок 10 – Окно редактирования здания

* **Удаление здания:** Выбрать здание из списка и нажать кнопку "Удалить". Подтвердить удаление (Рисунок 11).

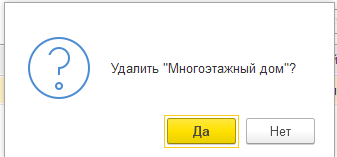


Рисунок 11 – Уведомление о удалении здания

* **Добавление объекта:** в разделе "Недвижимость" выбрать "Объекты" и нажать кнопку "Добавить". Заполнить все необходимые поля в форме добавления объекта (операция, здание, тип недвижимости, тип объекта, комментарий, дата добавления, номер квартиры, подъезд, этаж, статус, потолок, состояние ремонта, материал перекрытий, расположение комнат, тип санузла, входная дверь, тип окон, балкон, тип балкона, количество лифтов, пол, мебель, канализация, водоснабжение, электроснабжение, отопление, газоснабжение, кондиционер, правовой статус объекта, адрес, собственник, агент, общая площадь) (Рисунок 12).

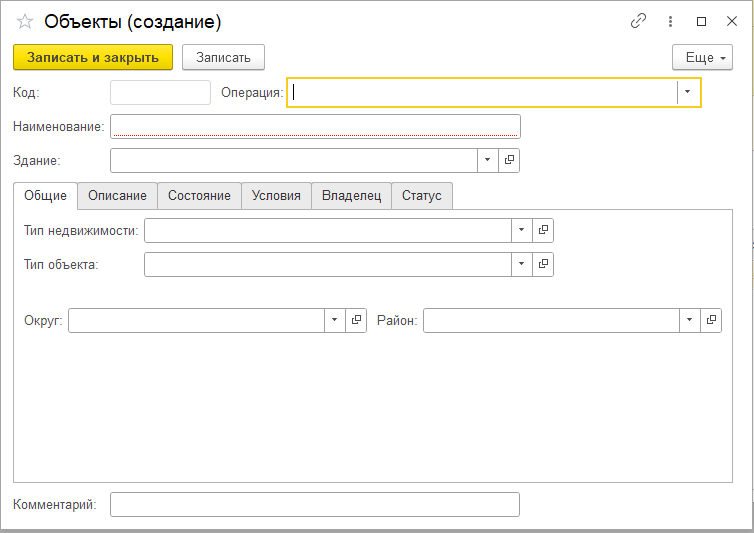


Рисунок 12 – Окно добавления объекта

* **Редактирование объекта:** Выбрать объект из списка и нажать кнопку "Изменить". Внести необходимые изменения и сохранить (Рисунок 13).

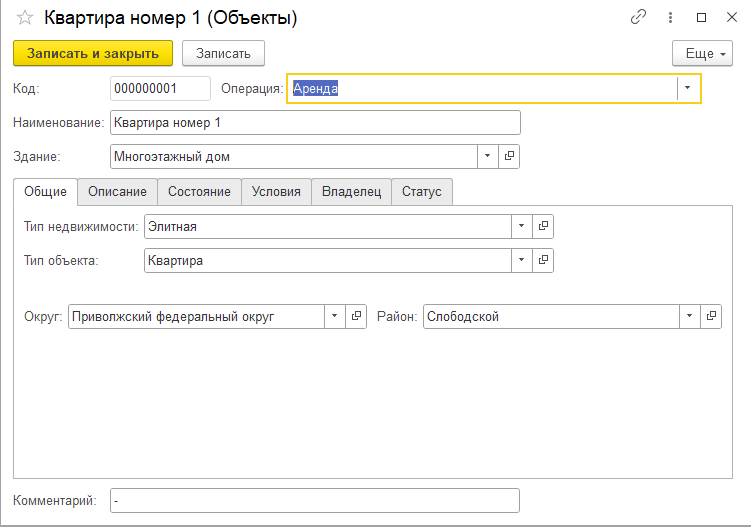


Рисунок 13 – Окно редактирования объекта

* **Удаление объекта:** Выбрать объект из списка и нажать кнопку "Удалить". Подтвердить удаление (Рисунок 14).

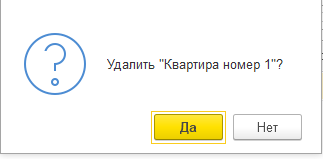


Рисунок 14 – Сообщение о удалении объекта

**5.2. Учет базы сотрудников**

* **Добавление сотрудника:** в разделе "Сотрудники" нажать кнопку "Добавить". Заполнить все необходимые поля в форме добавления сотрудника (ФИО, дата рождения, номер телефона, электронная почта, образование, должность, ИНН, адрес прописки) (Рисунок 15).

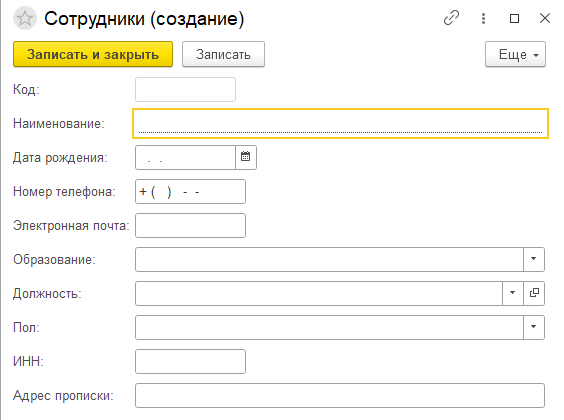


Рисунок 15 – Окно добавления сотрудника

* **Редактирование сотрудника:** Выбрать сотрудника из списка и нажать кнопку "Изменить". Внести необходимые изменения и сохранить (Рисунок 16).

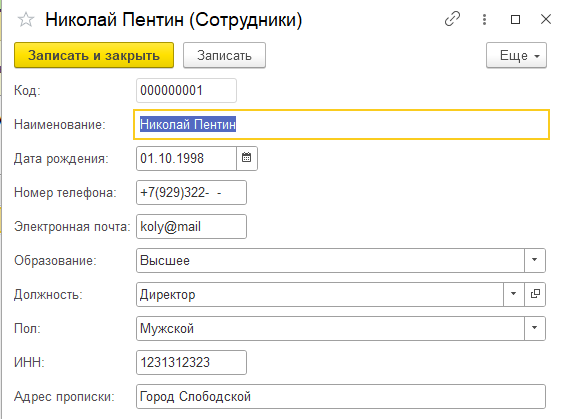


Рисунок 16 – Окно редактирования сотрудника

* **Удаление сотрудника:** Выбрать сотрудника из списка и нажать кнопку "Удалить". Подтвердить удаление (Рисунок 17).

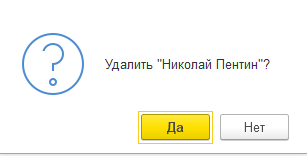


Рисунок 17 – Сообщение о удалении сотрудника

**5.3. Учет базы контрагентов**

* **Добавление контрагента:** в разделе "Контрагенты" нажать кнопку "Добавить". Заполнить все необходимые поля в форме добавления контрагента (ФИО, номер телефона, адрес, ИНН, пол, дата рождения) (Рисунок 18).

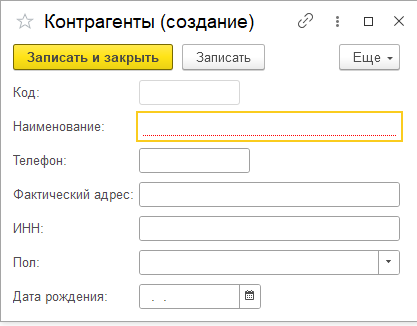


Рисунок 18 – Окно добавления контрагента

* **Редактирование контрагента:** Выбрать контрагента из списка и нажать кнопку "Изменить". Внести необходимые изменения и сохранить (Рисунок 19).

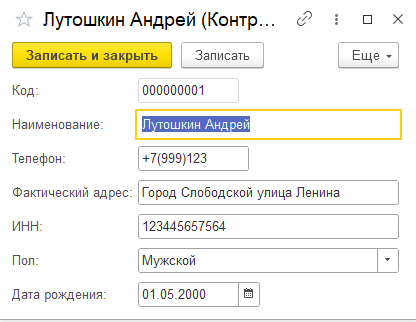


Рисунок 19 – Окно редактирования контрагента

* **Удаление контрагента:** Выбрать контрагента из списка и нажать кнопку "Удалить". Подтвердить удаление (Рисунок 20).

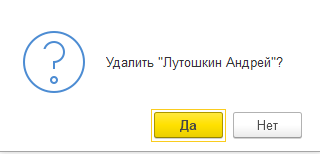


Рисунок 20 – Сообщение о удалении контрагента

**5.4. Выполнение операций**

* **Формирование заявки:** в разделе "Операции" выбрать "Заявки" и нажать кнопку "Добавить". Заполнить все необходимые поля в форме заявки (тип операции, объект, цена, агент, проценты, статус, адрес) (Рисунок 21).

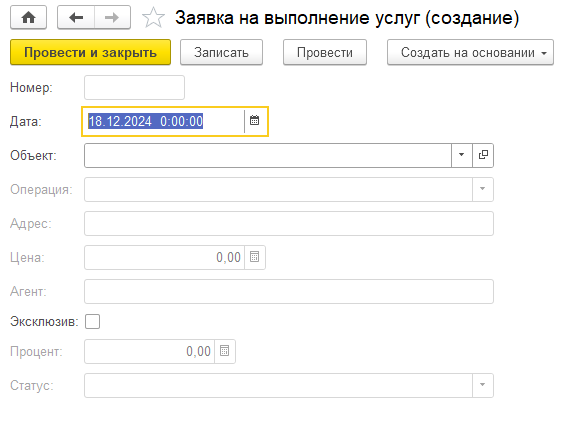


Рисунок 21 – Окно создания заявки на выполнение услуг

* **Формирование договора:** Выбрать заявку и нажать кнопку "Создать договор". Заполнить все необходимые поля в форме договора (заявка, агент, объект, проценты, контрагент, цена, итоговая сумма, тип операции, адрес) (Рисунок 22).

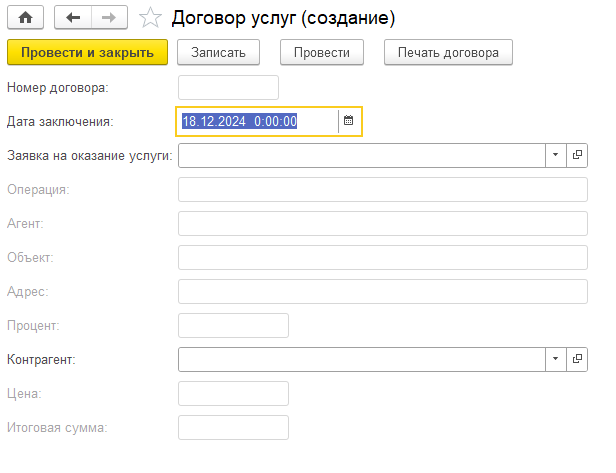


Рисунок 22 – Окно создания договора услуг

* **Печать договора:** После формирования договора нажать кнопку "Печать"(Рисунок 23).



Рисунок 23 – Окно печати договора на оказание услуг

**6. Резервное копирование**

Регулярно выполняйте резервное копирование базы данных, следуя инструкциям администратора системы.

**7. Выход из программы**

Для выхода из программы закройте окно 1С:Предприятие.

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения курсового проекта был разработан программный модуль для автоматизации бизнес-процессов агентства недвижимости. Основные задачи, поставленные в начале работы, были успешно решены. Созданный модуль предоставляет инструмент для эффективного управления объектами недвижимости, взаимодействия с клиентами и сопровождения сделок.

В ходе разработки были реализованы следующие ключевые функции:

* Автоматизирован учет объектов недвижимости, включая добавление, редактирование и удаление информации.
* Реализовано ведение базы клиентов (контрагентов) и истории взаимодействий с ними.
* Создан механизм формирования заявок на покупку/аренду недвижимости и генерации договоров.
* Реализована система отчетов для анализа деятельности агентства.
* Разработаны роли пользователей ("Администратор", "Менеджер") с разграничением прав доступа.

Разработанный модуль обладает удобным и интуитивно понятным интерфейсом, что обеспечивает простоту его использования сотрудниками агентства. Проведенное тестирование подтвердило корректность работы реализованных функций и стабильность системы.

Таким образом, созданный программный модуль представляет собой эффективное решение для автоматизации ключевых бизнес-процессов агентства недвижимости, позволяя сократить временные затраты, минимизировать риски ошибок и повысить качество обслуживания клиентов.

**СПИСОК ЛИТЕРАТУРЫ**

1. ГОСТ 21552-84. Единая система программной документации. Общие технические требования, приемка, методы испытаний, маркировка, упаковка, транспортирование и хранение, введение. 01.01.19876. – М.: Изд-во стандартов, 1980. – 4 с.
2. ГОСТ 19.103-77. Единая система программной документации. Обозначение программ и программных документов, введение. 01.07.1978. – М.: Изд-во стандартов, 1978. – 2 с.
3. ГОСТ 19.104-78. Единая система программной документации. Основные надписи, введение. 01.01.1978. – М.: Изд-во стандартов, 1980. – 6 с.
4. ГОСТ 19.106-78. Единая система программной документации. Требования к программным документам, выполненным печатным способом, введение. 01.01.1980. – М.: Изд-во стандартов, 1989. – 6 с.
5. ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению, введение. 01.01.1980. – М.: Изд-во стандартов, 1988. – 3 с.
6. ГОСТ 19.401-78. Единая система программной документации. Текст программы. Требования к содержанию и оформлению, введение. 01.01.1980. – М.: Изд-во стандартов, 1988. – 2 с.
7. ГОСТ 19.505-79. Единая система программной документации. Руководство оператора. Требования к содержанию и оформлению, введение. 01.01.1980. – М.: Изд-во стандартов, 1987. – 6 с.
8. ГОСТ 7.82-2001. СИБИД. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления, введение. 01.07.2002. –М.: Госстандарт России, 2001. – 23 с.
9. ГОСТ Р 7.0.5-2008. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая ссылка. Общие требования и правила составления, введение. 01.01.2009. – М.: Госстандарт России,2009. – 19 с.
10. Тимофеев, Г. Конфигурирование и администрирование 1С: Предприятия / Г. Тимофеев. - М.: Феникс, 2020. - 15 c.
11. Масуев, М. А. Проектирование предприятий автомобильного транспорта/М. А. Масуев. -М.: Академия, 2017. -224 с.
12. Филимонова, Е. В. 1С:Предприятие 7.7. Бухгалтерия и торговля в вопросах и ответах. Учебно-практическое пособие / Е.В. Филимонова, Н.А. Кириллова. - М.: Феникс, 2020. - 384 c.
13. Харитонов, С. А. Бухгалтерский и налоговый учет в "1С:Бухгалтерии 8" (редакция 2.0) / С.А. Харитонов. - М.: 1С-Паблишинг, 2018. - 680 c.
14. Харитонов, С. А. Настольная книга по оплате труда и ее расчету в "1С:Зарплата и управление персоналом 8" / С.А. Харитонов. - М.: 1С-Паблишинг, 2019. - 648 c.
15. Харитонов, С. А. Секреты профессиональной работы с "1С: Бухгалтерией 8" (редакция 2. 0). Учет расчетов по НДС / С.А. Харитонов, О.В. Шерст. - М.: 1С-Паблишинг, 2020. - 792 c.
16. Хрусталёва, Е.Ю. Разработка сложных отчётов в "1 С: Предприятие 8". Система компоновки данных / Е.Ю. Хрусталёва - 2-е изд. - М.: 000 "1С-Паблишинг", 2012. - 484 с.
17. Видеолекции по программированию 1С 8.2 [Электронный ресурс]. - Режим доступа: http://betbookit.net. - Дата доступа: 03.05.2014.
18. Чистов, Д. В. Практикум по программе "1С:Упрощенка 8" / Д.В. Чистов, С.А. Харитонов. - М.: 1С-Паблишинг, 2020. - 436 c.

# **ПРИЛОЖЕНИЯ**

## **Приложение 1**

Описание алгоритмов и функционирования программы

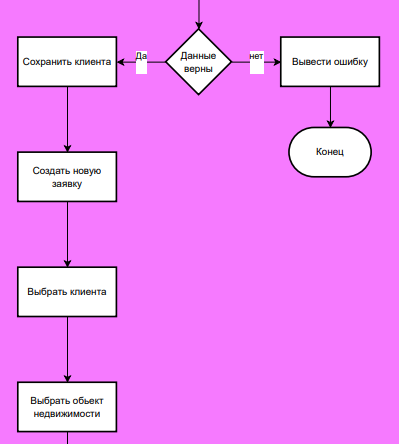
****

Рисунок 24 - Продолжение блок-схемы

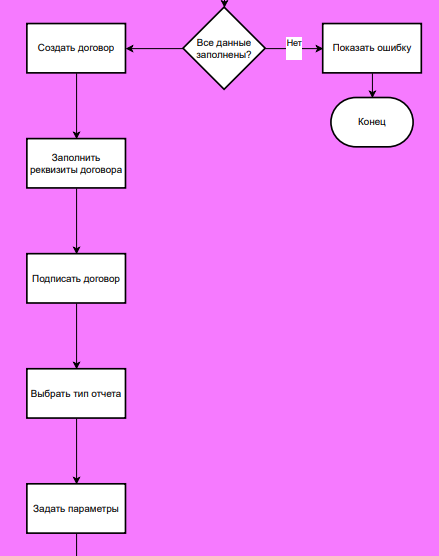


Рисунок 25 - Продолжение блок-схемы

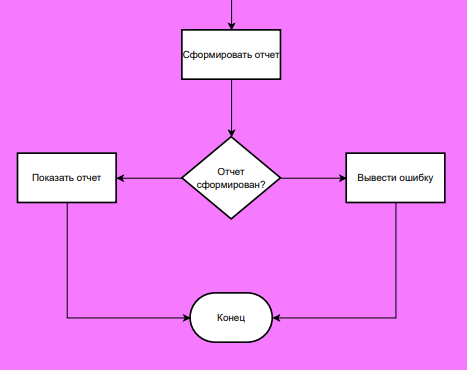


Рисунок 26 - Продолжение блок-схемы

## **Приложение 2**

Форма элемента справочника пользователи

#Область ОбработчикиСобытийФормы

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

УстановитьУсловноеОформление();

Если Параметры.Свойство("АвтоТест") Тогда // Возврат при получении формы для анализа.

Возврат;

КонецЕсли;

Если НЕ Пользователи.ЭтоПолноправныйПользователь(, ОбщегоНазначенияПовтИсп.РежимРаботыПрограммы().Локальный) Тогда

ВызватьИсключение НСтр("ru = 'Недостаточно прав для открытия списка пользователей информационной базы.'");

КонецЕсли;

Пользователи.НайтиНеоднозначныхПользователейИБ();

ТипыПользователей.Добавить(Тип("СправочникСсылка.Пользователи"));

Если ПолучитьФункциональнуюОпцию("ИспользоватьВнешнихПользователей") Тогда

ТипыПользователей.Добавить(Тип("СправочникСсылка.ВнешниеПользователи"));

КонецЕсли;

ПоказатьТолькоОбработанныеВКонфигураторе = Истина;

ЗаполнитьПользователейИБ();

КонецПроцедуры

&НаКлиенте

Процедура ОбработкаОповещения(ИмяСобытия, Параметр, Источник)

Если ИмяСобытия = "ДобавленПользовательИБ"

ИЛИ ИмяСобытия = "ИзмененПользовательИБ"

ИЛИ ИмяСобытия = "УдаленПользовательИБ"

ИЛИ ИмяСобытия = "ОчищеноСопоставлениеСНесуществующимПользователемИБ" Тогда

ЗаполнитьПользователейИБ();

КонецЕсли;

КонецПроцедуры

#КонецОбласти

#Область ОбработчикиСобытийЭлементовШапкиФормы

&НаКлиенте

Процедура ПоказатьТолькоОбработанныеВКонфигуратореПриИзменении(Элемент)

ЗаполнитьПользователейИБ();

КонецПроцедуры

#КонецОбласти

#Область ОбработчикиСобытийЭлементовТаблицыФормыПользователиИнфБазы

&НаКлиенте

Процедура ПользователиИнфБазыПриАктивизацииСтроки(Элемент)

ТекущиеДанные = Элементы.ПользователиИнфБазы.ТекущиеДанные;

Если ТекущиеДанные = Неопределено Тогда

МожноУдалить = Ложь;

МожноСопоставить = Ложь;

МожноПерейтиКПользователю = Ложь;

МожноОтменитьСопоставление = Ложь;

Иначе

МожноУдалить = Не ЗначениеЗаполнено(ТекущиеДанные.Ссылка);

МожноСопоставить = Не ЗначениеЗаполнено(ТекущиеДанные.Ссылка);

МожноПерейтиКПользователю = ЗначениеЗаполнено(ТекущиеДанные.Ссылка);

МожноОтменитьСопоставление = ЗначениеЗаполнено(ТекущиеДанные.Ссылка);

КонецЕсли;

Элементы.ПользователиИнфБазыУдалить.Доступность = МожноУдалить;

Элементы.ПользователиИнфБазыПерейтиКПользователю.Доступность = МожноПерейтиКПользователю;

Элементы.ПользователиИнфБазыКонтекстноеМенюПерейтиКПользователю.Доступность = МожноПерейтиКПользователю;

Элементы.ПользователиИнфБазыСопоставить.Доступность = МожноСопоставить;

Элементы.ПользователиИнфБазыСопоставитьСНовым.Доступность = МожноСопоставить;

Элементы.ПользователиИнфБазыОтменитьСопоставление.Доступность = МожноОтменитьСопоставление;

КонецПроцедуры

&НаКлиенте

Процедура ПользователиИнфБазыПередУдалением(Элемент, Отказ)

Отказ = Истина;

Если Не ЗначениеЗаполнено(Элементы.ПользователиИнфБазы.ТекущиеДанные.Ссылка) Тогда

УдалитьТекущегоПользователяИБ(Истина);

КонецЕсли;

КонецПроцедуры

#КонецОбласти

#Область ОбработчикиКомандФормы

&НаКлиенте

Процедура Обновить(Команда)

ЗаполнитьПользователейИБ();

КонецПроцедуры

&НаКлиенте

Процедура Сопоставить(Команда)

СопоставитьПользователяИБ();

КонецПроцедуры

&НаКлиенте

Процедура СопоставитьСНовым(Команда)

СопоставитьПользователяИБ(Истина);

КонецПроцедуры

&НаКлиенте

Процедура ПерейтиКПользователю(Команда)

ОткрытьПользователяПоСсылке();

КонецПроцедуры

&НаКлиенте

Процедура ОтменитьСопоставление(Команда)

Если Элементы.ПользователиИнфБазы.ТекущиеДанные = Неопределено Тогда

Возврат;

КонецЕсли;

Кнопки = Новый СписокЗначений;

Кнопки.Добавить("ОтменитьСопоставление", НСтр("ru = 'Отменить сопоставление'"));

Кнопки.Добавить("ОставитьСопоставление", НСтр("ru = 'Оставить сопоставление'"));

ПоказатьВопрос(

Новый ОписаниеОповещения("ОтменитьСопоставлениеПродолжение", ЭтотОбъект),

НСтр("ru = 'Отмена сопоставления пользователя информационной базы с пользователем в справочнике.

|

|Отмена сопоставления требуется крайне редко - только если сопоставление было выполнено некорректно, например,

|при обновлении информационной базы, поэтому не рекомендуется отменять сопоставление по любой другой причине.'"),

Кнопки,

,

"ОставитьСопоставление");

КонецПроцедуры

#КонецОбласти

#Область СлужебныеПроцедурыИФункции

&НаСервере

Процедура УстановитьУсловноеОформление()

УсловноеОформление.Элементы.Очистить();

//

Элемент = УсловноеОформление.Элементы.Добавить();

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.ПолноеИмя.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.Имя.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.АутентификацияСтандартная.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.АутентификацияОС.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.ПользовательОС.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.АутентификацияOpenID.Имя);

ОтборЭлемента = Элемент.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));

ОтборЭлемента.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("ПользователиИнфБазы.ДобавленВКонфигураторе");

ОтборЭлемента.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;

ОтборЭлемента.ПравоеЗначение = Истина;

Элемент.Оформление.УстановитьЗначениеПараметра("ЦветТекста", ЦветаСтиля.ЦветОсобогоТекста);

//

Элемент = УсловноеОформление.Элементы.Добавить();

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.ПолноеИмя.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.Имя.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.АутентификацияСтандартная.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.АутентификацияОС.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.ПользовательОС.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.АутентификацияOpenID.Имя);

ГруппаОтбора1 = Элемент.Отбор.Элементы.Добавить(Тип("ГруппаЭлементовОтбораКомпоновкиДанных"));

ГруппаОтбора1.ТипГруппы = ТипГруппыЭлементовОтбораКомпоновкиДанных.ГруппаИли;

ОтборЭлемента = ГруппаОтбора1.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));

ОтборЭлемента.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("ПользователиИнфБазы.ИзмененВКонфигураторе");

ОтборЭлемента.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;

ОтборЭлемента.ПравоеЗначение = Истина;

ОтборЭлемента = ГруппаОтбора1.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));

ОтборЭлемента.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("ПользователиИнфБазы.УдаленВКонфигураторе");

ОтборЭлемента.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;

ОтборЭлемента.ПравоеЗначение = Истина;

Элемент.Оформление.УстановитьЗначениеПараметра("ЦветТекста", ЦветаСтиля.НедоступныеДанныеЦвет);

//

Элемент = УсловноеОформление.Элементы.Добавить();

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.Имя.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.АутентификацияСтандартная.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.АутентификацияОС.Имя);

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.АутентификацияOpenID.Имя);

ОтборЭлемента = Элемент.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));

ОтборЭлемента.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("ПользователиИнфБазы.УдаленВКонфигураторе");

ОтборЭлемента.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;

ОтборЭлемента.ПравоеЗначение = Истина;

Элемент.Оформление.УстановитьЗначениеПараметра("Текст", НСтр("ru = '<Нет данных>'"));

Элемент.Оформление.УстановитьЗначениеПараметра("Формат", "Л=ru; БЛ=Нет; БИ=Да");

//

Элемент = УсловноеОформление.Элементы.Добавить();

ПолеЭлемента = Элемент.Поля.Элементы.Добавить();

ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.АутентификацияОС.Имя);

ОтборЭлемента = Элемент.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));

ОтборЭлемента.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("ПользователиИнфБазы.ПользовательОС");

ОтборЭлемента.ВидСравнения = ВидСравненияКомпоновкиДанных.НеЗаполнено;

Элемент.Оформление.УстановитьЗначениеПараметра("Формат", "Л=ru; БЛ=; БИ=Да");

КонецПроцедуры

&НаСервере

Процедура ЗаполнитьПользователейИБ()

ПустойУникальныйИдентификатор = Новый УникальныйИдентификатор("00000000-0000-0000-0000-000000000000");

Если Элементы.ПользователиИнфБазы.ТекущаяСтрока <> Неопределено Тогда

Строка = ПользователиИнфБазы.НайтиПоИдентификатору(Элементы.ПользователиИнфБазы.ТекущаяСтрока);

Иначе

Строка = Неопределено;

КонецЕсли;

ТекущийИдентификаторПользователяИБ =

?(Строка = Неопределено, ПустойУникальныйИдентификатор, Строка.ИдентификаторПользователяИБ);

ПользователиИнфБазы.Очистить();

ИдентификаторыНесуществующихПользователейИБ.Очистить();

ИдентификаторыНесуществующихПользователейИБ.Добавить(ПустойУникальныйИдентификатор);

Запрос = Новый Запрос;

Запрос.УстановитьПараметр("ПустойУникальныйИдентификатор", ПустойУникальныйИдентификатор);

Запрос.Текст =

"ВЫБРАТЬ

| Пользователи.Ссылка КАК Ссылка,

| Пользователи.Наименование КАК ПолноеИмя,

| Пользователи.ИдентификаторПользователяИБ,

| ЛОЖЬ КАК ЭтоВнешнийПользователь

|ИЗ

| Справочник.Пользователи КАК Пользователи

|ГДЕ

| Пользователи.ИдентификаторПользователяИБ <> &ПустойУникальныйИдентификатор

|

|ОБЪЕДИНИТЬ ВСЕ

|

|ВЫБРАТЬ

| ВнешниеПользователи.Ссылка,

| ВнешниеПользователи.Наименование,

| ВнешниеПользователи.ИдентификаторПользователяИБ,

| ИСТИНА

|ИЗ

| Справочник.ВнешниеПользователи КАК ВнешниеПользователи

|ГДЕ

| ВнешниеПользователи.ИдентификаторПользователяИБ <> &ПустойУникальныйИдентификатор";

Выгрузка = Запрос.Выполнить().Выгрузить();

Выгрузка.Индексы.Добавить("ИдентификаторПользователяИБ");

Выгрузка.Колонки.Добавить("Сопоставлен", Новый ОписаниеТипов("Булево"));

ПользователиИБ = ПользователиИнформационнойБазы.ПолучитьПользователей();

Для Каждого ПользовательИБ Из ПользователиИБ Цикл

СвойстваПользовательИБ = Пользователи.НовоеОписаниеПользователяИБ();

Пользователи.ПрочитатьПользователяИБ(ПользовательИБ.УникальныйИдентификатор, СвойстваПользовательИБ);

ИзмененВКонфигураторе = Ложь;

Строка = Выгрузка.Найти(СвойстваПользовательИБ.УникальныйИдентификатор, "ИдентификаторПользователяИБ");

Если Строка <> Неопределено Тогда

Строка.Сопоставлен = Истина;

Если Строка.ПолноеИмя <> СвойстваПользовательИБ.ПолноеИмя Тогда

ИзмененВКонфигураторе = Истина;

КонецЕсли;

КонецЕсли;

Если ПоказатьТолькоОбработанныеВКонфигураторе

И Строка <> Неопределено

И Не ИзмененВКонфигураторе Тогда

Продолжить;

КонецЕсли;

НоваяСтрока = ПользователиИнфБазы.Добавить();

НоваяСтрока.ПолноеИмя = СвойстваПользовательИБ.ПолноеИмя;

НоваяСтрока.Имя = СвойстваПользовательИБ.Имя;

НоваяСтрока.АутентификацияСтандартная = СвойстваПользовательИБ.АутентификацияСтандартная;

НоваяСтрока.АутентификацияОС = СвойстваПользовательИБ.АутентификацияОС;

НоваяСтрока.ИдентификаторПользователяИБ = СвойстваПользовательИБ.УникальныйИдентификатор;

НоваяСтрока.ПользовательОС = СвойстваПользовательИБ.ПользовательОС;

НоваяСтрока.АутентификацияOpenID = СвойстваПользовательИБ.АутентификацияOpenID;

Если Строка = Неопределено Тогда

// Пользователя ИБ нет в справочнике.

НоваяСтрока.ДобавленВКонфигураторе = Истина;

Иначе

НоваяСтрока.Ссылка = Строка.Ссылка;

НоваяСтрока.СопоставленСВнешнимПользователем = Строка.ЭтоВнешнийПользователь;

НоваяСтрока.ИзмененВКонфигураторе = ИзмененВКонфигураторе;

КонецЕсли;

КонецЦикла;

Отбор = Новый Структура("Сопоставлен", Ложь);

Строки = Выгрузка.НайтиСтроки(Отбор);

Для каждого Строка Из Строки Цикл

НоваяСтрока = ПользователиИнфБазы.Добавить();

НоваяСтрока.ПолноеИмя = Строка.ПолноеИмя;

НоваяСтрока.Ссылка = Строка.Ссылка;

НоваяСтрока.СопоставленСВнешнимПользователем = Строка.ЭтоВнешнийПользователь;

НоваяСтрока.УдаленВКонфигураторе = Истина;

ИдентификаторыНесуществующихПользователейИБ.Добавить(Строка.ИдентификаторПользователяИБ);

КонецЦикла;

Отбор = Новый Структура("ИдентификаторПользователяИБ", ТекущийИдентификаторПользователяИБ);

Строки = ПользователиИнфБазы.НайтиСтроки(Отбор);

Если Строки.Количество() > 0 Тогда

Элементы.ПользователиИнфБазы.ТекущаяСтрока = Строки[0].ПолучитьИдентификатор();

КонецЕсли;

КонецПроцедуры

&НаСервере

Процедура УдалитьПользователяИБ(ИдентификаторПользователяИБ, Отказ)

ОписаниеОшибки = "";

Если НЕ Пользователи.УдалитьПользователяИБ(ИдентификаторПользователяИБ, ОписаниеОшибки) Тогда

ОбщегоНазначенияКлиентСервер.СообщитьПользователю(ОписаниеОшибки, , , , Отказ);

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура ОткрытьПользователяПоСсылке()

ТекущиеДанные = Элементы.ПользователиИнфБазы.ТекущиеДанные;

Если ТекущиеДанные = Неопределено Тогда

Возврат;

КонецЕсли;

Если ЗначениеЗаполнено(ТекущиеДанные.Ссылка) Тогда

ОткрытьФорму(

?(ТекущиеДанные.СопоставленСВнешнимПользователем,

"Справочник.ВнешниеПользователи.ФормаОбъекта",

"Справочник.Пользователи.ФормаОбъекта"),

Новый Структура("Ключ", ТекущиеДанные.Ссылка));

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура УдалитьТекущегоПользователяИБ(УдалитьСтроку = Ложь)

ПоказатьВопрос(

Новый ОписаниеОповещения("УдалитьТекущегоПользователяИБЗавершение", ЭтотОбъект, УдалитьСтроку),

НСтр("ru = 'Удалить пользователя информационной базы?'"),

РежимДиалогаВопрос.ДаНет);

КонецПроцедуры

&НаКлиенте

Процедура УдалитьТекущегоПользователяИБЗавершение(Ответ, УдалитьСтроку) Экспорт

Если Ответ = КодВозвратаДиалога.Да Тогда

Отказ = Ложь;

УдалитьПользователяИБ(

Элементы.ПользователиИнфБазы.ТекущиеДанные.ИдентификаторПользователяИБ, Отказ);

Если Не Отказ И УдалитьСтроку Тогда

ПользователиИнфБазы.Удалить(Элементы.ПользователиИнфБазы.ТекущиеДанные);

КонецЕсли;

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура СопоставитьПользователяИБ(СНовым = Ложь)

Если ТипыПользователей.Количество() > 1 Тогда

ТипыПользователей.ПоказатьВыборЭлемента(

Новый ОписаниеОповещения("СопоставитьПользователяИБДляТипаЭлемента", ЭтотОбъект, СНовым),

НСтр("ru = 'Выбор типа данных'"),

ТипыПользователей[0]);

Иначе

СопоставитьПользователяИБДляТипаЭлемента(ТипыПользователей[0], СНовым);

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура СопоставитьПользователяИБДляТипаЭлемента(ЭлементСписка, СНовым) Экспорт

Если ЭлементСписка = Неопределено Тогда

Возврат;

КонецЕсли;

ИмяСправочника = ?(ЭлементСписка.Значение = Тип("СправочникСсылка.Пользователи"), "Пользователи", "ВнешниеПользователи");

Если Не СНовым Тогда

ПараметрыФормы = Новый Структура;

ПараметрыФормы.Вставить("РежимВыбора", Истина);

ПараметрыФормы.Вставить("ИдентификаторыНесуществующихПользователейИБ", ИдентификаторыНесуществующихПользователейИБ);

ОткрытьФорму("Справочник." + ИмяСправочника + ".ФормаВыбора", ПараметрыФормы,,,,,

Новый ОписаниеОповещения("СопоставитьПользователяИБСЭлементом", ЭтотОбъект, ИмяСправочника));

Иначе

СопоставитьПользователяИБСЭлементом("Новый", ИмяСправочника);

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура СопоставитьПользователяИБСЭлементом(Элемент, ИмяСправочника) Экспорт

Если Не ЗначениеЗаполнено(Элемент) И Элемент <> "Новый" Тогда

Возврат;

КонецЕсли;

ПараметрыФормы = Новый Структура;

Если Элемент <> "Новый" Тогда

ПараметрыФормы.Вставить("Ключ", Элемент);

КонецЕсли;

ПараметрыФормы.Вставить("ИдентификаторПользователяИБ",

Элементы.ПользователиИнфБазы.ТекущиеДанные.ИдентификаторПользователяИБ);

ОткрытьФорму("Справочник." + ИмяСправочника + ".ФормаОбъекта", ПараметрыФормы, ЭтотОбъект);

КонецПроцедуры

&НаКлиенте

Процедура ОтменитьСопоставлениеПродолжение(Ответ, Неопределен) Экспорт

Если Ответ = "ОтменитьСопоставление" Тогда

ОтменитьСопоставлениеНаСервере();

КонецЕсли;

КонецПроцедуры

&НаСервере

Процедура ОтменитьСопоставлениеНаСервере()

ТекущаяСтрока = ПользователиИнфБазы.НайтиПоИдентификатору(

Элементы.ПользователиИнфБазы.ТекущаяСтрока);

Объект = ТекущаяСтрока.Ссылка.ПолучитьОбъект();

Объект.ИдентификаторПользователяИБ = Неопределено;

Объект.ОбменДанными.Загрузка = Истина;

Объект.Записать();

ЗаполнитьПользователейИБ();

КонецПроцедуры

#КонецОбласти

Форма документа заявка на операцию

&НаСервере

Процедура ОбъектПриИзмененииНаСервере()

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Объекты.Операция КАК Операция,

| Объекты.Статус КАК Статус,

| Объекты.Агент КАК Агент,

| Объекты.Стоимость КАК Стоимость,

| Объекты.Процент КАК Процент,

| Объекты.ФактическийАдрес КАК ФактическийАдрес

|ИЗ

| Справочник.Объекты КАК Объекты";

Выборка = Запрос.Выполнить().Выбрать();

Пока Выборка.Следующий() Цикл

Объект.Операция = Выборка.Операция;

Объект.Статус = Выборка.Статус;

Объект.Агент = Выборка.Агент;

Объект.Процент = Выборка.Процент;

Объект.Цена = Выборка.Стоимость;

Объект.Адрес = Выборка.ФактическийАдрес;

КонецЦикла;

КонецПроцедуры

&НаКлиенте

Процедура ОбъектПриИзменении(Элемент)

ОбъектПриИзмененииНаСервере();

КонецПроцедуры

## **Приложение 3**

Компакт-диск с материалами проекта

На диске располагается:

* Файлы конфигурации (SF, DT)
* Проект программы
* Файл курсового проекта в формате MS Word
* Файл с презентацией курсового проекта