

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

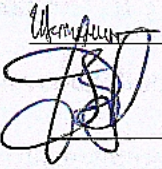
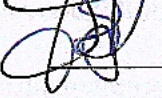
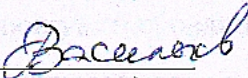
вычислительной техники

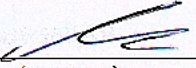
Кафедра

«Вычислительная техника»

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»
Магистерская программа: «Вычислительные машины, комплексы, системы и сети»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ
на тему
ПРОГРАММА ДЛЯ ОБРАБОТКИ,
ХРАНЕНИЯ И ВИЗУАЛИЗАЦИИ
ИНФОРМАЦИИ

Студент  31.05.2022 Шетинин Кирилл Александрович
(подпись, дата) (ФИО полностью)
Руководитель  31.05.2022 Федюнин Р.Н.
(подпись, дата) (фамилия, инициалы)
Нормоконтролёр  31.05.2022 Бычков А.С.
(подпись, дата) (фамилия, инициалы)
Рецензент  31.05.2022 Васильев А.В.
(должность, место работы) (подпись, дата) (фамилия, инициалы)
Работа допущена к защите (протокол зас. кафедры от 02.06.2022 г. №13

Заведующий кафедрой  М.А. Митрохин
(подпись) (фамилия, инициалы)

Работа защищена с отметкой  (протокол заседания ГЭК от

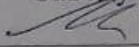
14.06.2022 г. № 8)
Секретарь ГЭК 
(подпись)

Е.И. Гурин
(фамилия, инициалы)

Пенза 2022

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
 М.А. Митрохин
« 24 » сентября 20 20 г.

ЗАДАНИЕ НА МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ

1. Студент Щетинин Кирилл Александрович гр. 20ВВм1 факультета ВТ

2. Тема работы «Программа для обработки, хранения и визуализации информации»

Тема утверждена приказом ПГУ № 656/0 от " 24 " сентября 20 20 г.

3. Руководитель магистерской диссертации к.т.н., доцент Р.Н. Федюнин

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Назначение разработки: требуется разработать программу как универсальную систему документооборота, удовлетворяющую требованиям всех отделов организации.

Программа должна использовать справочники или классификаторы для хранения постоянной или редко меняющейся информации, которая применяется различными отделами, а также персоналом организации.

5. Перечень подлежащих разработке вопросов:

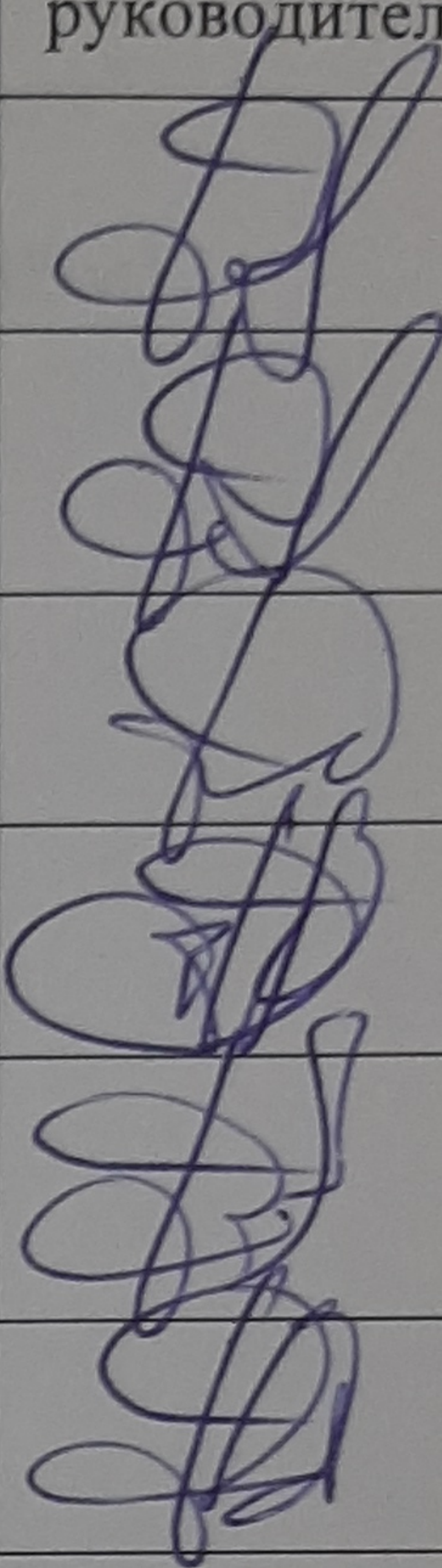
1. Анализ предметной области: описать существующие системы учета документов, выявить их недостатки.

2. Выявление методов, необходимых для разработки системы документооборота.

3. Разработка системы документооборота.

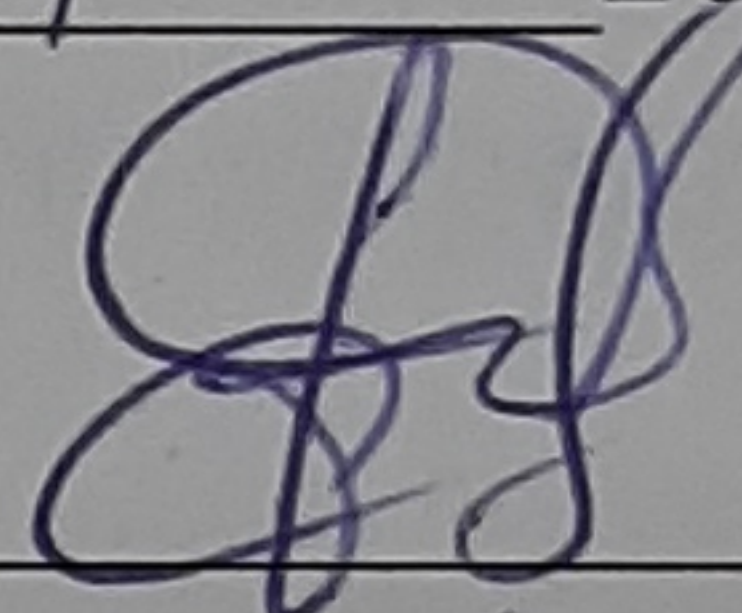
4. Анализ полученных результатов.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	16%	21.09.2020-21.11.2020	
2.	Анализ опыта построения систем учета	10%	22.11.2020-22.01.2021	
3.	Разработка системы учета	37%	23.01.2021-23.04.2021	
4.	Отладка рекомендательного модуля	10%	24.04.2021-29.01.2022	
5.	Анализ полученных результатов исследования	12%	30.01.2022-30.03.2022	
6.	Оформление выпускной работы	15%	31.03.2022-31.05.2022	

Дата выдачи задания " 21 " сентября 2020 г.

Руководитель магистерской диссертации

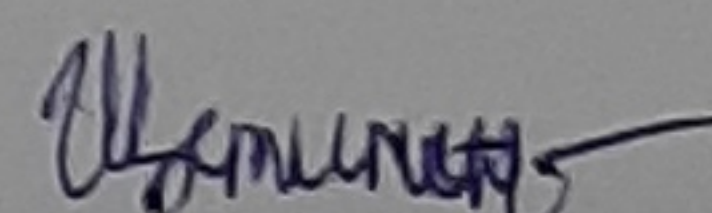


21.09.20

(подпись, дата)

Р.Н. Федюнин

Задание к исполнению принял студент

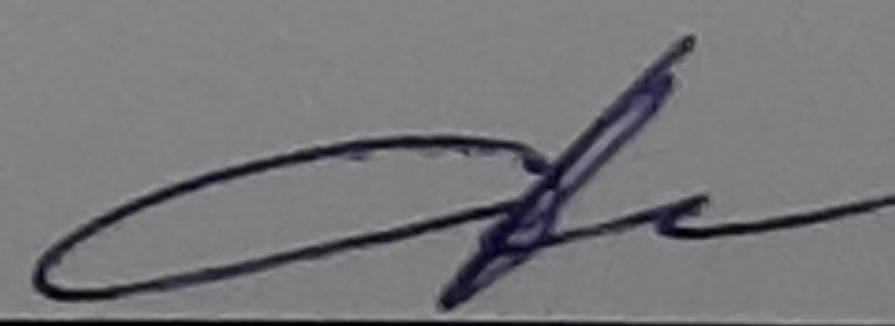


21.09.20

(подпись, дата)

Щетинин К.А.

Работу к защите допустить
Декан факультета



(подпись, дата)

09.06.2022

Фионова Л.Р.

РЕФЕРАТ

Пояснительная записка содержит 95 листов, 66 рисунков, 7 таблиц, 21 источник, 2 приложения.

ИНФОРМАЦИОННАЯ СИСТЕМА, СИСТЕМА ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА, СУБД, СИСТЕМА РАЗРАБОТКИ, ИНТЕРФЕЙС

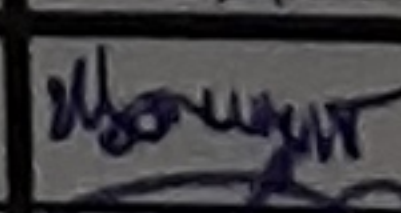
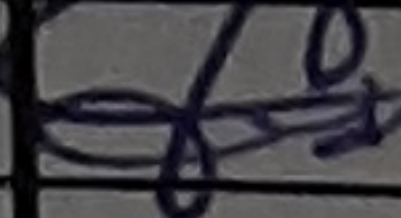
Объект исследования — это информационная система, а также основополагающие способы учета движения различной документации, как внутренней, так и внешней.

Предметом исследования являются существующие информационные системы для учета документооборота.

Цель исследования данной работы - разработка информационной системы управления электронным документооборотом предприятия.

Научная новизна исследования заключается в написании простейших информационных систем для небольших предприятий. Это позволяет значительно сократить время обучения персонала предприятия, сэкономить на техническом и программном обеспечении, а также ускорить развёртывание и внедрение информационных систем.

В результате разработанная информационная система подняла учет документации на более высокий уровень, предоставляя недоступные ранее функции работы с документами и отчетностью.

					ПГУ 1.090401.09.001 ПЗ			
Изм.	Лист	№ докум	Подпись	Дата	Программа для обработки, хранения и визуализации информации Пояснительная записка	Лит.	Лист	Листов
Разраб.		Щетинин К.А.						
Провер.		Федюнин Р.Н.					2	95
Н. Контр.		Бычков А.С.				ФВТ, каф. ВТ гр. 20ВВм1		
Утв.		Митрохин М.А.						

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	5
ВВЕДЕНИЕ	6
1 Теоретическая часть	7
1.1 Краткая характеристика предприятия	7
1.2 Описание существующей системы учета	8
1.3 Выявление недостатков существующей системы учета	9
1.4 Обоснование автоматизации документооборота	9
1.5 Анализ существующих вариантов решения задачи	10
1.6 Выбор и обоснование своего варианта решения задачи	22
1.7 Расширенная постановка задачи	22
2 Проектная часть	25
2.1 Выбор системы разработки	25
2.2 Функциональная структура разрабатываемой системы	27
2.3 Архитектура разрабатываемой системы	32
2.4 Разработка концептуальной (инфологической) модели данных	33
2.5 Выбор системы управления базами данных	34
2.6 Разработка физической модели базы данных	36
3 Реализация системы	44
3.1 Общая концепция разрабатываемой системы	44
3.2 Создание прикладных объектов	44
3.3 Разработка интерфейса системы	53
3.4 Разработка запросов	57
3.5 Руководство по установке и настройке программы	60
ЗАКЛЮЧЕНИЕ	71
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	72

ПРИЛОЖЕНИЕ А Листинг программы	74
ПРИЛОЖЕНИЕ Б Презентация	90

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящем отчете о НИР применяют следующие сокращения и обозначения.

БД – база данных

ГМ – главное меню

ГФ – главная форма

ИС – информационная система

ЛС – локальная сеть

ПП – программный продукт

СУБД – система управления базами данных

СЭД – система электронного документооборота

ФЛ – физическое лицо

ЯП – язык программирования

ВВЕДЕНИЕ

Работы с различными документами в любой организации предполагают создание структуры бизнес-процессов предприятия, определение способов хранения документов и их применения персоналом.

Процесс документооборота — это сложное множество различных взаимосвязанных процедур, совокупность которых обеспечивает движение документов на предприятии с начала их создания или поступления на предприятие, и до завершения работы с ними или отправки адресатам.

В целях правильной организации документооборота предприятия все документы делятся по документопотокам, количество которых определяется потребностями организации, например, регистрируемые документы и нерегистрируемые документы, входящие, внутренние и исходящие документы и т.д.

Система электронного документооборота (СЭД) — это «многопользовательская» программа, облегчающая управление предприятием в сфере учета различных документов, применяемых в жизнедеятельности организации.

Задачи диссертации:

- изучение предметной области задачи, определение задач автоматизации,
- разработка «ER-диаграммы» и проектирование структуры базы данных (БД) СЭД,
- разработка интерфейса и программных модулей ИС,
- написание полного руководства по установке, использованию ИС и ее тестирование.

Практическая значимость разработки и внедрения ИС учета документов заключается в сокращении времени, которое затрачивается на выполнение основных операций в отделе архива и уменьшении вероятности ошибочных решений из-за влияния человеческого фактора.

1 Теоретическая часть

1.1 Краткая характеристика предприятия

Акционерное общество “Пензенский научно-исследовательский электротехнический институт” (АО “ПНИЭИ”, бывшее ФГУП “ПНИЭИ”) образовано в 1958 году, на сегодняшний день является ведущим предприятием, решающим вопросы в сфере информационной безопасности.

За полувековой период научно-производственной деятельности в институте разработано, принято на вооружение, освоено в серийном производстве и эксплуатации более 200 типов аппаратуры, комплексов технических средств защиты и передачи информации для Вооруженных Сил РФ, систем и комплексов телекоммуникаций государственных органов Российской Федерации, получено 674 авторских свидетельств и патентов на изобретения в области создания техники связи, 5 разработок отмечены Государственными премиями СССР и премией Правительства Российской Федерации в области науки и техники.

АО “ПНИЭИ” имеет лицензии МО РФ, ФСБ России, ФСТЭК России на проведение полного цикла работ и оказание услуг в сфере информационной безопасности [1].

Структура организации представлена на рисунке 1.1.

Ядро команды организации составляют опытные профессионалы, обладающие уникальными компетенциями и многолетним опытом работы в организации.



Рисунок 1.1 — Структура компании

1.2 Описание существующей системы учета

В данной диссертации решается задача автоматизации работы документооборота организации. ИС требуется разрабатывать как универсальную систему документооборота, удовлетворяющую требованиям всех отделов организации.

ИС должна использовать справочники или классификаторы для хранения постоянной или редко меняющейся информации, которая применяется различными отделами, а также персоналом организации.

Например, справочник организаций является единым справочником системы, используемым всеми ее пользователями.

Документы ИС также применяются сообща, всеми подразделениями предприятия.

Большой частью СЭД является подсистема отчетности. Данная подсистема также объединяет все отделы организации.

Управление организацией основывается на информации о клиентах компании, её поставщиках.

В существующей системе учета все эти данные хранятся разрозненно, часть в виде бумажной документации, часть в виде отдельных электронных документов.

1.3 Выявление недостатков существующей системы учета

До текущего момента времени, предприятие не использует никакой СЭД, все данные хранятся либо в «бумажном» виде, либо в виде электронных документов: файлов «Office», «Excel», «JPEG», «AVI» и т.д.

Составление отчетов также выполняется «вручную» с применением программ «Microsoft Excel» и «Microsoft Word», что отнимает много времени и требует занятости большого количества персонала.

Информация о контрагентах берется «вручную» из полученных «бумажных» документов.

Оформление договоров, карточек клиентов и т.п. ведется с применением программ «Microsoft Office», которые дают возможность создавать и изменять формы различных документов.

Главный недостаток текущего способа документооборота связан с отсутствием правильной организации хранения и использования данных, разнесенных по различным ПК объединенных в локальную сеть (ЛС). Все документы, касающиеся данных контрагентов, различных договоров, внутренних и внешних документов, являются множеством отдельных файлов формата «MS Word» и «MS Excel», хранящимся на различных ПК в ЛС.

Основные «минусы» применяемого способа документооборота:

- нет общего архива данных,
- нет стандарта хранимых данных,
- формирование документов «вручную»,
- дублирование данных в файлах.

1.4 Обоснование автоматизации документооборота

Актуальность организации СЭД обусловлена большим объемом документооборота современных предприятий и нормальное функционирование предприятия без использования средств вычислительной

техники (ВТ) и программного обеспечения (ПО) невозможно. Основным преимуществом является то, что удобство взаимодействия с ИС в разы больше, чем при работе без неё. Применение «многопользовательского» способа функционирования тоже является большим преимуществом применения ИС. Использование ИС, выполняющей автоматизацию ведения документоучета повышает функционирование организации на новый уровень.

Ещё одним плюсом применения ИС является информационная безопасность. В случае хранения данных в виде бумажных документов, утрата подобного документа была бы невосполнимой. Хранение данных в «электронном» виде позволяет организовывать различные способы безопасности информации, а также её архивирование для большей сохранности данных.

Из выше сказанного можно назвать главные плюсы применения ИС для решения поставленной задачи организации СЭД:

- удобство нахождения необходимой информации,
- ускорение различных операций, как поиска данных, так и внесения данных,
- мгновенное получение аналитической отчетности,
- информационная безопасность,
- наличие «многопользовательского» режима функционирования, что позволяет группе сотрудников совместно работать с документами.

1.5 Анализ существующих вариантов решения задачи

Рассмотрим самые применяемые СЭД в России: «Directum», «PayDox», «DocsVision», «ЕВФРАТ», «1С: Документооборот», «Босс-референт», «ДЕЛО», «Globus Professional», «МОТИВ». Для подготовки обзора брались данные из сети «Интернет». Все возможности СЭД поделены на 6 подзадач:

- 1) внесение новых документов,
- 2) использование документов,
- 3) контроль над потоками заданий,
- 4) отбор и анализ данных,

- 5) информационная безопасность,
- 6) настройки СЭД.

В общем, рабочие возможности СЭД похожи, и только подробная детализация задач документооборота и особенности их решения дают сравнить различные программные продукты. Помимо рассмотренных вариантов, существуют еще более 50 ПП, получивших меньшее распространение [2].

«DIRECTUM» - российская масштабируемая ЕСМ-система с готовым межкорпоративным документооборотом и вовлечением всех сотрудников от топ-менеджера до конечного исполнителя (рисунок 1.2) Позволяет организовать сквозные процессы внутри холдингов и межкорпоративный документооборот, стыкуя между собой учетные системы, СЭД и сервисы обмена. Единая точка входа исключает необходимость работы в нескольких системах и обеспечивает полностью безбумажное юридически значимое взаимодействие [3].

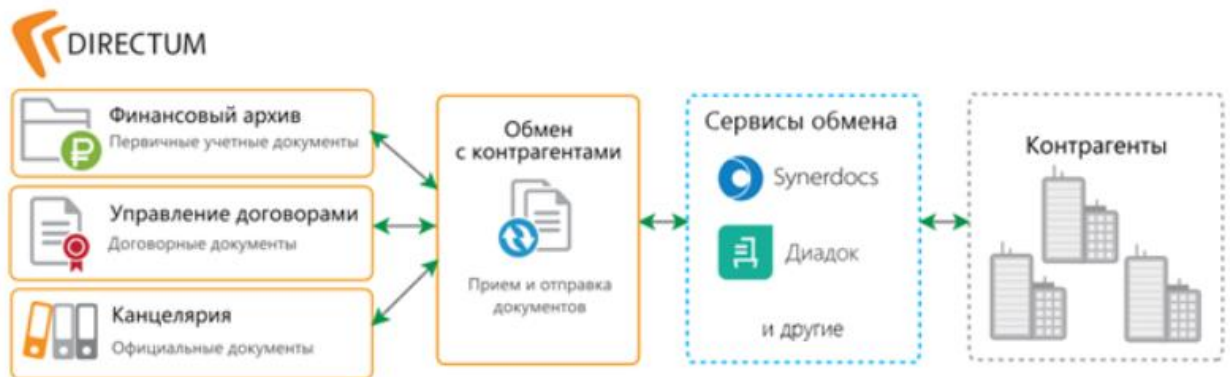
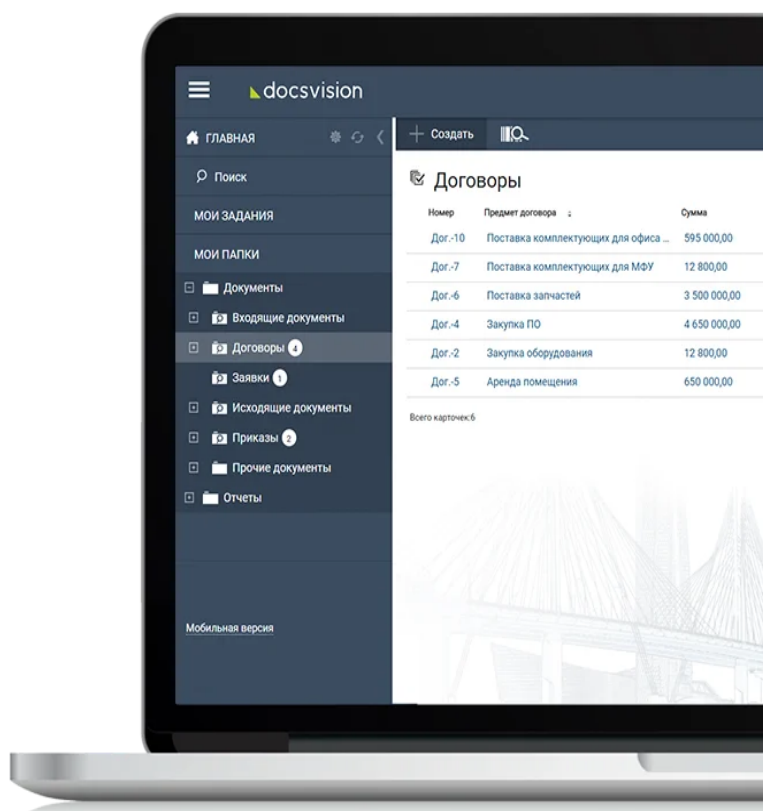


Рисунок 1.2 — «DIRECTUM»

«Docsvision» - полнофункциональная СЭД/ЕСМ-платформа российской разработки. Позволяет автоматизировать широкий круг бизнес-процессов и задач по обработке документов (рисунок 1.3). Входит в ТОП-3 российских СЭД/ЕСМ (по данным обзора аналитического портала TAdviser, декабрь 2019 г.) [4].



Электронный архив документов
Удобный ввод и хранение, быстрое распознавание, поиск и выгрузка данных



Договорной документооборот с контрагентами
Быстрое согласование и обмен документами, архив и контроль исполнения



Кадровый электронный документооборот
Автоматизация кадровых и HR-процессов, быстрая обработка документов сотрудников



Делопроизводство
Лёгкое управление документами: регистрация, согласование и контроль



Цифровое управление
Удобная работа с заданиями и поручениями в едином интерфейсе, контроль исполнения



Специализированные решения
Найдите решение для своей бизнес-задачи, основанное на опыте наших клиентов

Рисунок 1.3 — «Docsvision»

Корпоративная система «Globus Professional» позволяет полностью автоматизировать деятельность компании (рисунок 1.4). Автоматизация охватывает как работу с документами, так и все ключевые направления, которые влияют на эффективность бизнес-процессов в организации. Использование Globus Professional открывает для предприятия такие возможности, которые остались бы недоступными при использовании конкурирующих программных продуктов.

Функционал программы решает вопросы:

- перехода на электронный оборот документов,
- мотивации и контроля деятельности персонала,
- управления проектами,
- управления денежными ресурсами,
- бюджетирования,
- управления материальными ресурсами,
- управления загрузкой оборудования,
- управления рисками,

- защиты от «инсайдеров»,
- оперативной аналитики и отчётов [5].

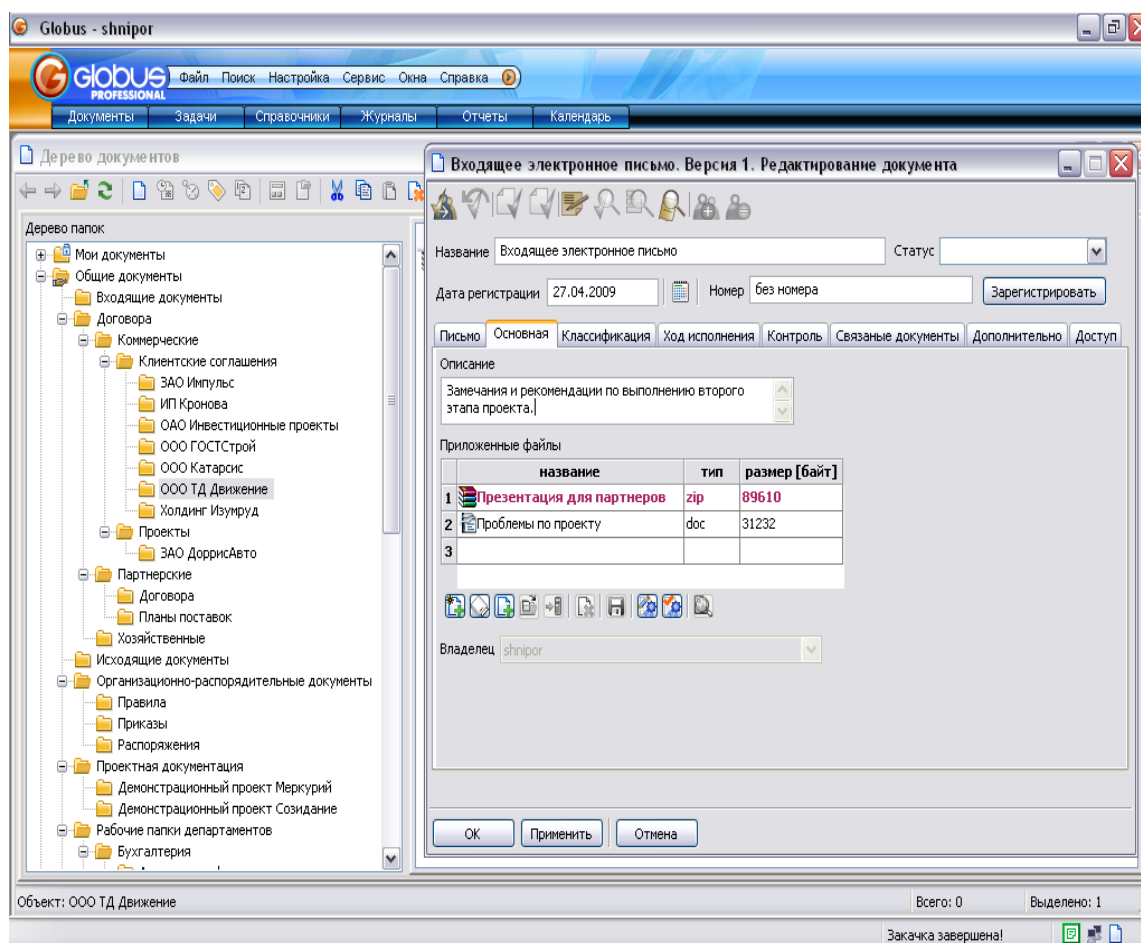


Рисунок 1.4 — Приложение «Globus Professional»

СЭД и поддержки совместной работы «PayDox» обеспечивает комплексное управление бизнес-процессами и документами предприятия и обладает неограниченными возможностями адаптации под требования заказчиков (рисунок 1.5) [6].



Рисунок 1.5 — Приложение «PayDox»

Приложение «1С: Документооборот» - это не просто система управления документами, но еще и современная система управления корпоративным контентом с широким набором функциональных возможностей для регулировки деловых процессов и совместной работы сотрудников (рисунок 1.6). Программа позволяет автоматизировать типовые процессы работы с документами, организовать электронный документооборот, обеспечить контроль исполнения задач, регламентировать управленческую деятельность [7].

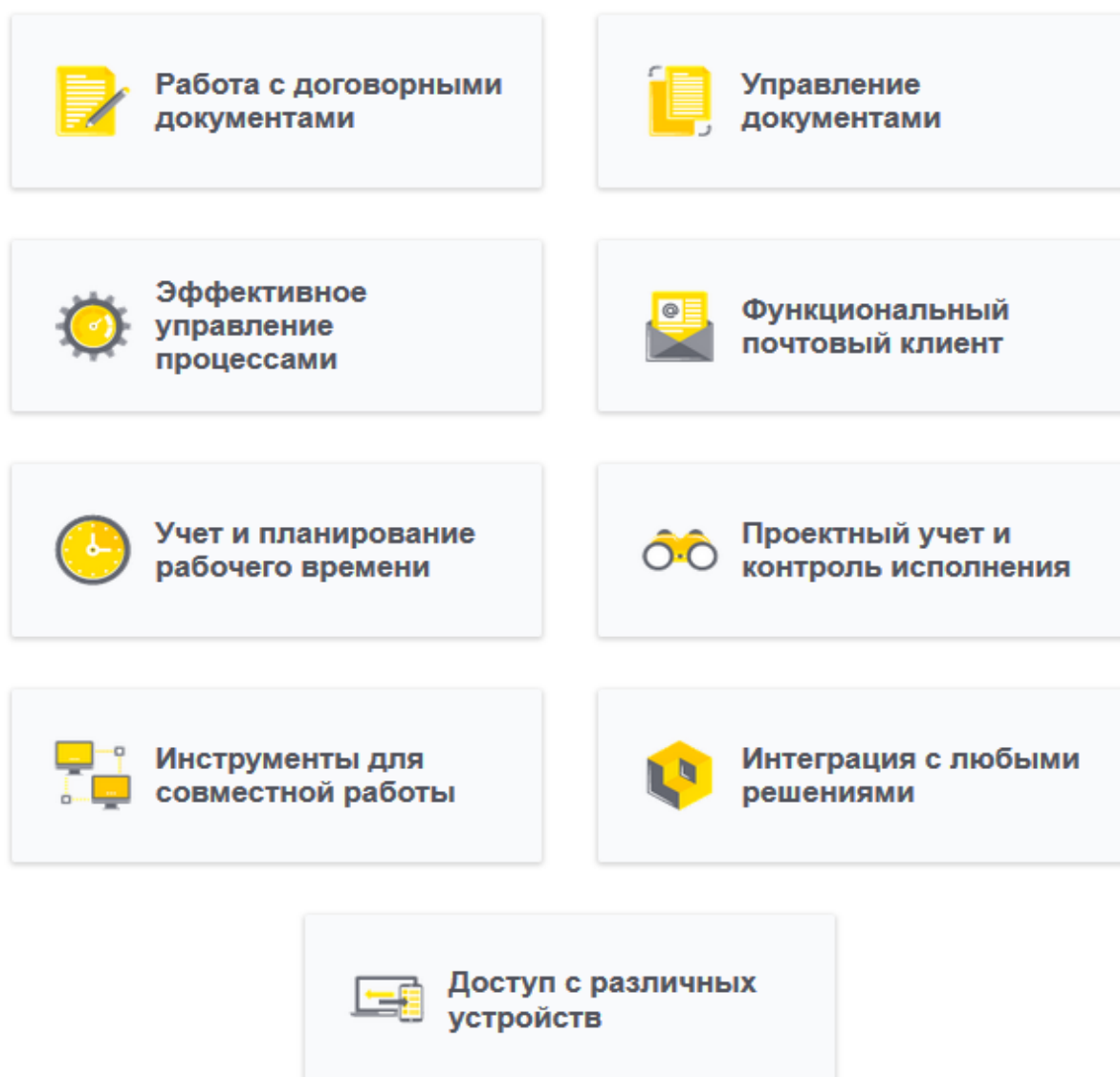


Рисунок 1.6 — Приложение «1С:Документооборот»

Система «БОСС-Референт» разработана для автоматизации управленческого документооборота и делопроизводства (рисунок 1.7). Ее потребителями являются коммерческие компании, федеральные государственные унитарные предприятия, государственные органы власти [8].

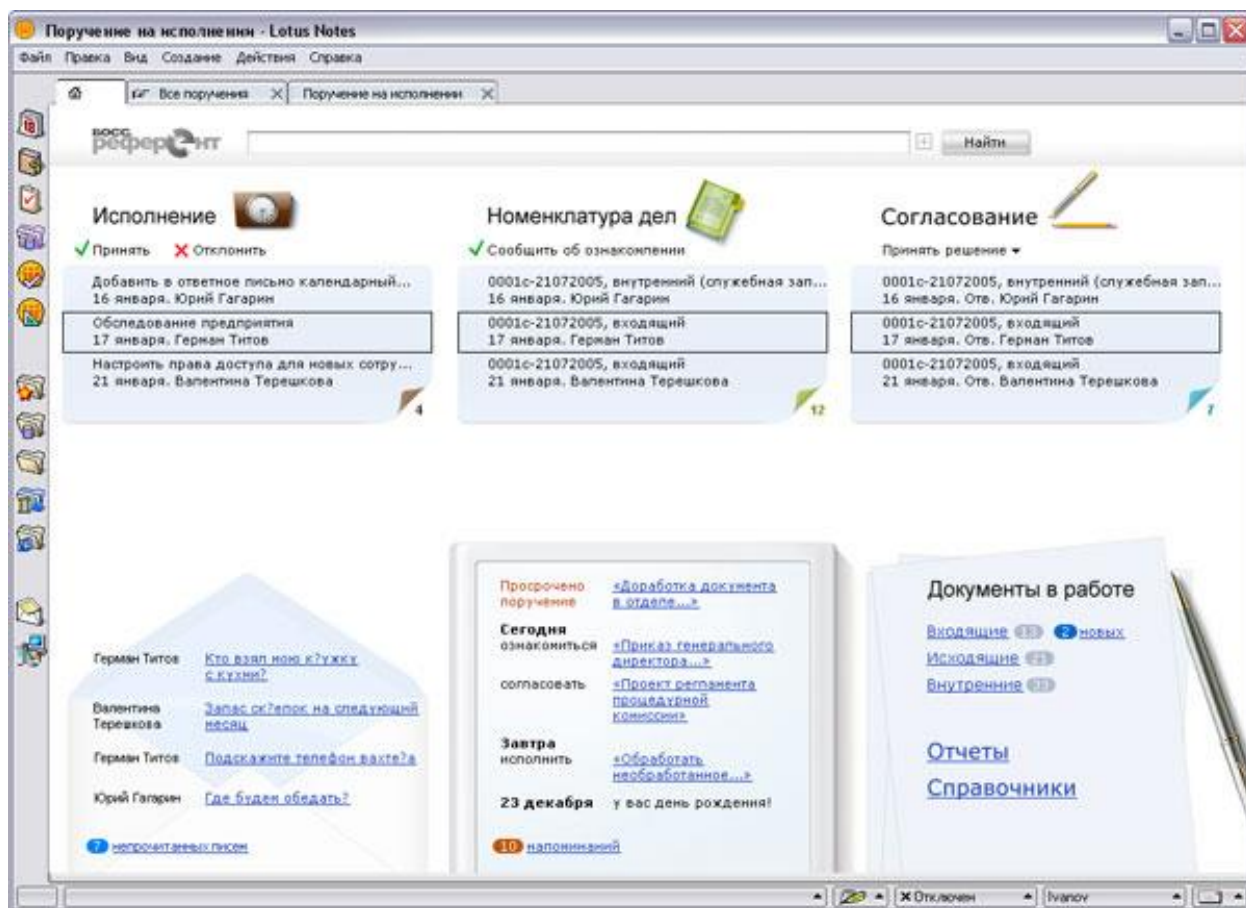


Рисунок 1.7 — Приложение «БОСС-Референт»

««ДЕЛО» - система с полным набором инструментов для управления документооборотом и делопроизводством, рассчитанная на максимальные нагрузки. Надежное и безотказное решение» (рисунок 1.8) [9].

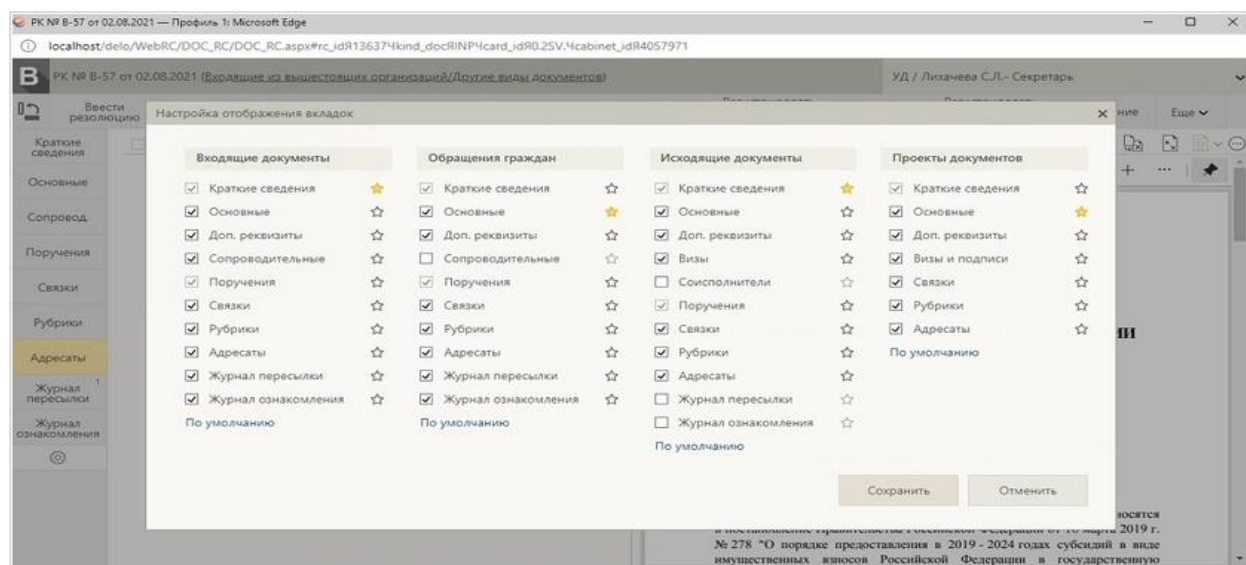


Рисунок 1.8 — Приложение «ДЕЛО»

«ЕВФРАТ» — система электронного документооборота и автоматизации бизнес-процессов, разработанная российской компанией «Cognitive Technologies». Система предназначена для комплексной автоматизации процессов делопроизводства, создания электронных архивов документов, организации корпоративного документооборота, автоматизации типовых бизнес-процессов (рисунок 1.9) [10].

Рисунок 1.9 — Приложение «ЕВФРАТ»

«МОТИВ» — это система оперативного управления компанией, обеспечивающая ведение электронного документооборота (СЭД), контроль исполнения поручений, управление проектами («Project Management»), взаимодействие с клиентами, совместную работу сотрудников подразделений. В МОТИВ можно организовать автоматизированную обработку запросов («HelpDesk», «Service Desk»), а также настроить многие другие бизнес-процессы (рисунок 1.10) [11].

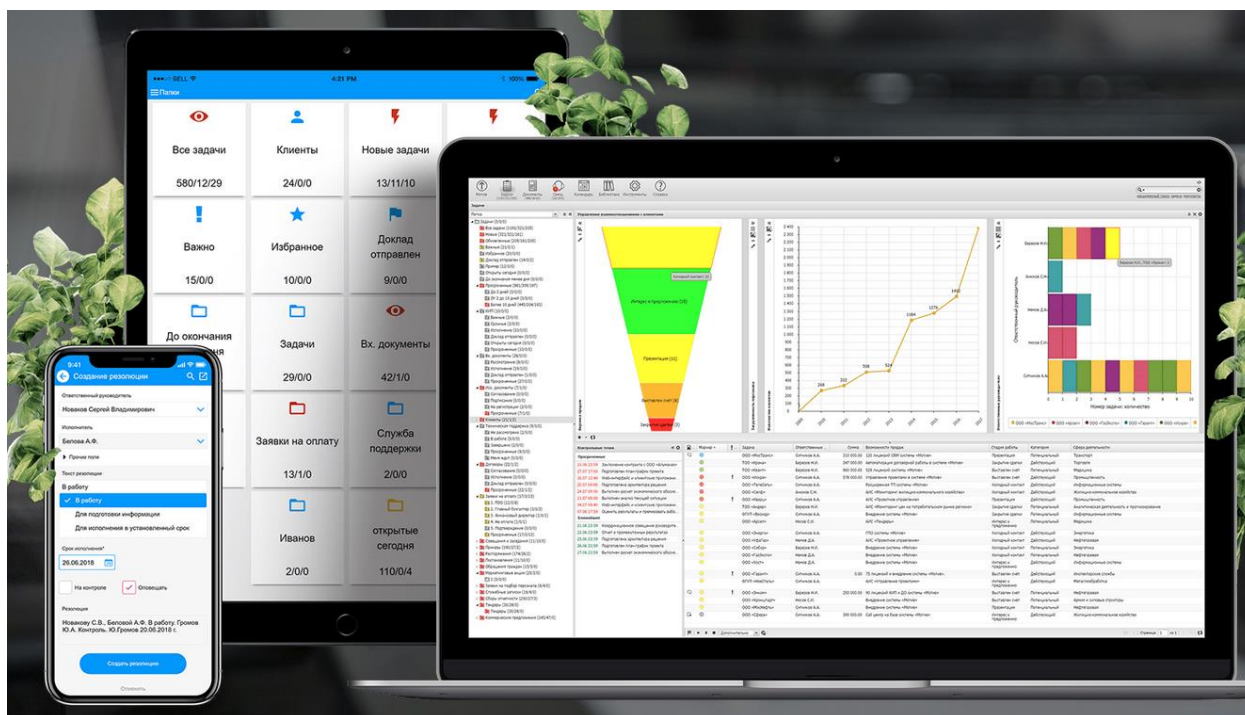


Рисунок 1.10 — «МОТИВ»

При анализе рабочих возможностей СЭД взяты следующие сокращения:

- «+» - возможность присутствует,
- «+-» - возможность ограничена,
- «-» - возможность отсутствует.

Результаты анализа приведены в таблицах 1.1 – 1.6

Таблица 1.1 — Внесение информации

Внесение различной документации в СЭД	«Directum»	«DocsVision»	«Globus Professional»	«PayDox»	«1С:Документооборот»	«Босс-референт»	«ДЕЛО»	«ЕВФРАТ»	«МОТИВ»
Внесение различной документации	+	+	+	+	+	+	+	+	+
Возможность прикрепления файлов разных форматов	+	+	+	+	+	+	+	+	+
Новые документы по заранее созданному шаблону	+	+	+	+	+	-	+	+	+

Таблица 1.1 — продолжение

Внесение различной документации в СЭД	«Directum»	«DocsVision»	«Globus Professional»	«PayDox»	«1С:Документооборот»	«Босс-референт»	«ДЕЛО»	«ЕВФРАТ»	«МОТИВ»
Внесение из «Email»	+	+	-	-	+	+	+	+	+
Внесение из «MS Office / OpenOffice»	+-	+-	-	-	-	-	+-	+	-
Потоковый ввод информации	+-	+-	-	-	-	+-	+-	+-	-

Таблица 1.2 — Использование документации

Работа с различной документации в СЭД	«Directum»	«DocsVision»	«Globus Professional»	«PayDox»	«1С:Документооборот»	«Босс-референт»	«ДЕЛО»	«ЕВФРАТ»	«МОТИВ»
Встроенный просмотр присоединенных файлов различных форматов	-	+	-	-	-	+	-	-	-
Создание списков рассылок	+	+	+	+	+	+	+	+	+
Работа с проектами	+	+	+-	+-	+-	+	+	+	+
Ведение личных папок пользователя	+	+	+-	+-	+-	+	+-	+	+
Работа с версиями документов	+	+	+	+	+	+	+	+	+
Ведение листа согласования	+	+	-	+	-	-	+	+	+

Таблица 1.3 — Контроль над потоками задач

Управление потоками работ	«Directum»	«DocsVision»	«Globus Professional»	«PayDox»	«1С:Документооборот»	«Босс-референт»	«ДЕЛО»	«ЕВФРАТ»	«МОТИВ»
Формирование типовых маршрутов движения документации	+	+	+	+	+-	+	+	+	+
Свободная маршрутизация документации	+	+	+-	+	-	+	-	+	+
Отсроченное исполнение	+-	+	-	+	-	-	+	-	+
Контроль за исполнением	+	+	+	+	+	+	+	+	+
Рассылка уведомлений	+	+	+	+	+	+	+	+	+
Рассылка документации для прочтения	+	+	+	+	+	+	+	+-	+

Таблица 1.4 — Отбор и анализ информации

Поиск и анализ	«Directum»	«DocsVision»	«Globus Professional»	«PayDox»	«1С:Документооборот»	«Босс-референт»	«ДЕЛО»	«ЕВФРАТ»	«МОТИВ»
Поиск по реквизитам	+	+	+	+	+	+	+	+	+
Полнотекстовый поиск данных	+-	+-	+-	+-	+	+	+-	+	+
Различные способы поиска в едином запросе	+	+	+-	+	+	+	+	+	+
Применение шаблонов при поиске	+-	+	+	-	-	+	+	+-	+

Таблица 1.5 — Безопасность хранения данных

Безопасность данных	«Directum»	«DocsVision»	«Globus Professional»	«PayDox»	«1С:Документооборот»	«Босс-референт»	«ДЕЛО»	«ЕВФРАТ»	«МОТИВ»
Различные способы аутентификации	+-	+	+-	+-	+-	+	+-	+-	+-
Назначение прав пользователям	+	+	+	+	+	+	+	+	+
Разграничение прав доступа по объектам системы	+	+-	+	+	+-	+	+-	+-	+
Использование ЭЦП	+	+	+	+	+	+	+	+	+

Таблица 1.6 — Результирующая оценка СЭД

Результирующая оценка	«Directum»	«DocsVision»	«Globus Professional»	«PayDox»	«1С:Документооборот»	«Босс-референт»	«ДЕЛО»	«ЕВФРАТ»	«МОТИВ»
Регистрация и ввод документации	5	5	3	3	4	3.5	5	5.5	4
Работа с документацией	5	6	3	4	3	5	4.5	5	5
Управление потоками работ	5.5	6	4.5	6	3.5	5	5	4.5	6
Поиск и анализ данных	3	3.5	3	2.5	3	4	3.5	3.5	4
Безопасность информации	3.5	3.5	3.5	3.5	3	4	3	3	3.5
Итого	22	24	17	19	16,5	21,5	21	21,5	22,5

Выполнив анализ наиболее известных в России СЭД, делаем вывод о необходимости написания своей СЭД. Данный вывод определяется большой стоимостью, сложностью и длительным временем освоения существующих СЭД. Также написание своей ИС вытекает из темы диссертации.

1.6 Выбор и обоснование своего варианта решения задачи

Задание диссертации подразумевает разработку собственной ИС. Эта цель ясна, так как в своей ИС можно выполнить свои, требуемые только в этой организации функции. Бывает так, что техническое задание к таким возможностям формируется после создания и внедрения ИС, на этапе функционирования, если у организации формируются новые требования на нужный функционал ИС, форматы отчетов и другие функциональные возможности ИС.

Очень часто в процессе работы с разработанной ИС возникают пожелания о создании новых или измененных отчетов, внесении дополнительных реквизитов и атрибутов элементам различных справочников и документов. Выполнить такие пожелания заказчика можно только имея ИС своей разработки, т.к. «чужие» ИС поставляются, обычно, без «исходных кодов» и не могут быть изменены.

Создание ИС необходимо выполнить с помощью какой-либо удобной и известной системы разработки ПП. Распространенность дает возможность поддержки и расширения функционала ИС наиболее распространенными программистами. Также универсальность позволяет дополнять ИС новыми функциями.

1.7 Расширенная постановка задачи

Целью диссертации является организация СЭД в организации.

Разрабатываемая ИС учета документов должна применять единый архив - базу данных.

ИС должна максимально исключить «ручной» ввод информации в текстовые поля. Желательно создать отдельные таблицы БД для всех сущностей предметной области, даже если они не играют решающего значения в процедуре оформления и учета документов. Сущности, зависящие друг от друга, необходимо реализовать в виде логически связанных таблиц БД.

ИС должна обладать интуитивно понятным интерфейсом работы. Интерфейс должен увеличивать скорость работы оператора и уменьшать вероятность совершения ошибок ввода данных.

Оператор ИС должен иметь возможность управлять программой с помощью клавиатуры для ускорения работы, так и применяя манипулятор типа «мышь».

Функциональность ИС включает в себя:

- ведение дополнительных или «простейших» справочников,
- ведение организаций,
- ведение физических лиц (ФЛ),
- ведение различных параметров вводимой документации,
- возможность прикрепления файлов различных форматов,
- ведение журнала документов,
- внесение пользователей и их авторизация,
- введения параметров различной документаци**и**.

«Простейшие» справочники это:

- подразделения,
- проекты,
- статусы документа,
- типы контакта,
- виды документа,
- и т.д.

Ведение списка документов – одна из основных функций ИС. Документ характеризуется следующими реквизитами: название, номер, дата, ответственный, подразделение, проект, вид, статус. К любому элементу списка можно прикрепить большое количество «электронных» объектов: картинки, файлы «MS Office» и т.д.

Журнал документов представляет собой «дела», объединяющие группу связанных документов. Журнал документов характеризуется следующими

реквизитами: тип, номер, дата, контрагент, адрес, ответственный, подразделение, проект, вид, статус.

В табличной части можно прикрепить отдельные документы.

ИС учета документов должна позволять следующие действия с журналом документов:

- ввод нового документа,
- редактирование документа,
- удаление,
- применение фильтра по ячейке,
- поиск,
- печать списка,
- сортировка.

ИС должна осуществлять выгрузку журнала документов, а также списков любых справочников в ПП «MS Excel 2013». Список должен выводиться в соответствии с установленными настройками фильтра, сортировки и настройки полей.

2 Проектная часть

2.1 Выбор системы разработки

Одним из самых «старых» и используемых языков программирования (ЯП) является «C++». В данном языке объекты представляются «классами». «Класс» расписывает, как именно функционируют объекты определенного типа, их «жизнедеятельность» от создания до удаления. Классы позволяют создавать программные модули более понятными [12].

«IDE «C++ Builder» — программный продукт, инструмент быстрой разработки приложений (RAD), интегрированная среда программирования (IDE), система, используемая программистами для разработки программного обеспечения на языках программирования Си и C++» (рисунок 2.1) [13].

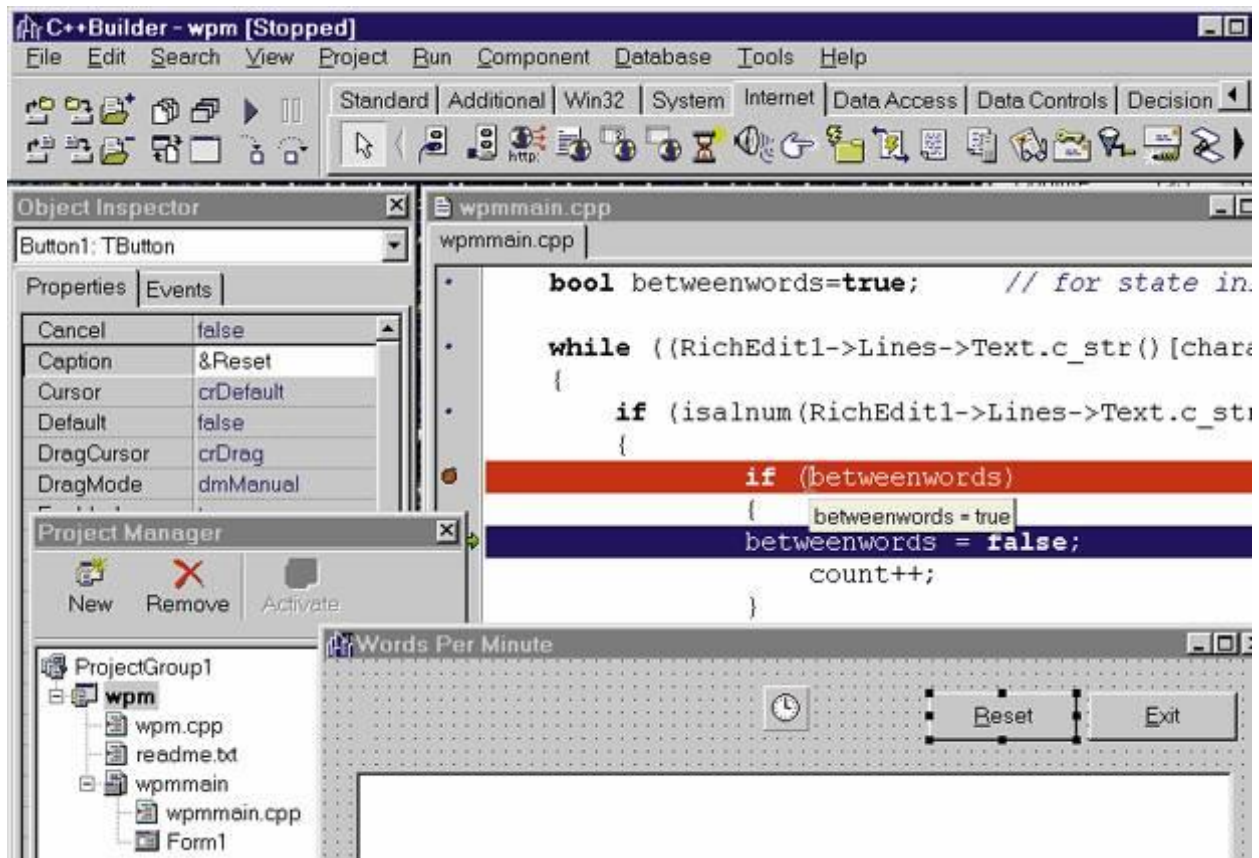


Рисунок 2.1 — «C++ Builder»

«Eclipse» — свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается «Eclipse Foundation»» (рисунок 2.2) [14].

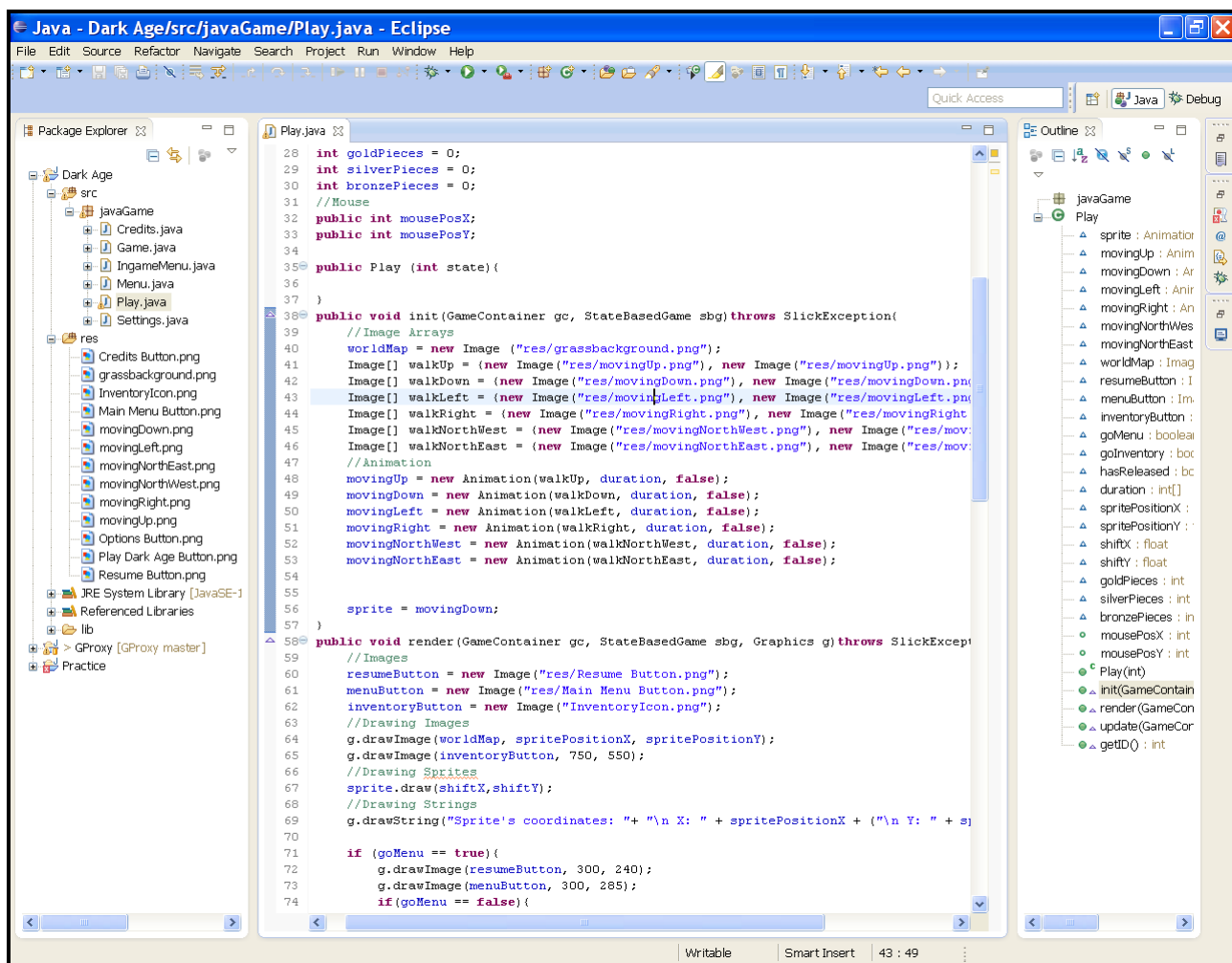


Рисунок 2.2 — «Eclipse»

Более новым языком, наследующимся от C++ является продукт фирмы Microsoft «C#» [15]. «C#» применяется для работы с библиотеками «.NET Framework» [16]. Применение данного языка позволяет значительно ускорить разработку в среде программ Microsoft (рисунок 2.3) [17].

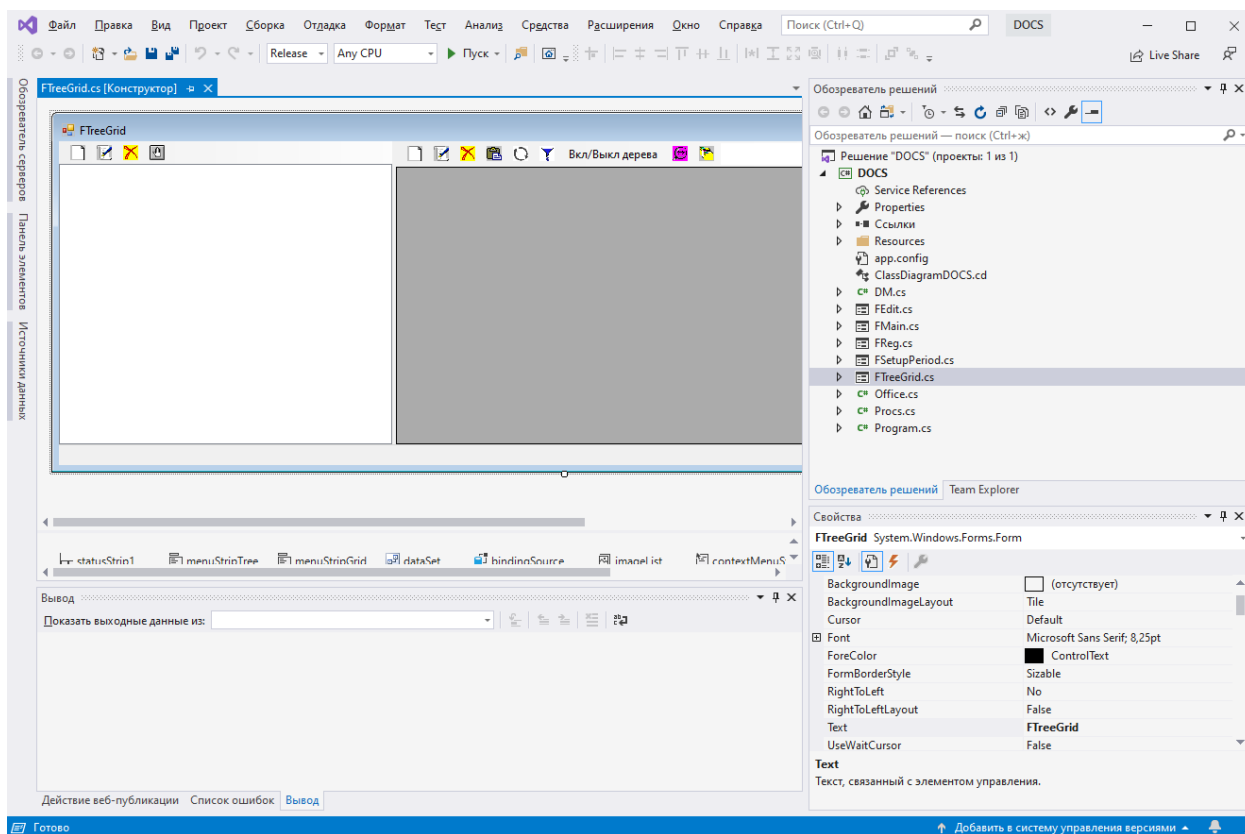


Рисунок 2.3 — IDE «Visual Studio» и язык «C#»

Результат анализа приведен в таблице 2.1.

Таблица 2.1 — Сравнение IDE и языков программирования

IDE	Лицензирование	Кроссплатформенный	Применение отладчика	Разработка графического интерфейса	Автодополнение	Анализатор кода	Дерево классов
«RAD C++ Builder»	«проприетарная»	-	+	+	+	+	?
«Microsoft Visual Studio C#»	«проприетарная»	+	+	+	+	+	+
«Eclipse Java»	«EPL»	+	-	-	-	-	-

В связи с явными преимуществами средой разработки выбрана «Microsoft Visual Studio», а языком программирования – «C#».

2.2 Функциональная структура разрабатываемой системы

На рисунке 2.4 представлена структурная схема создаваемой ИС.

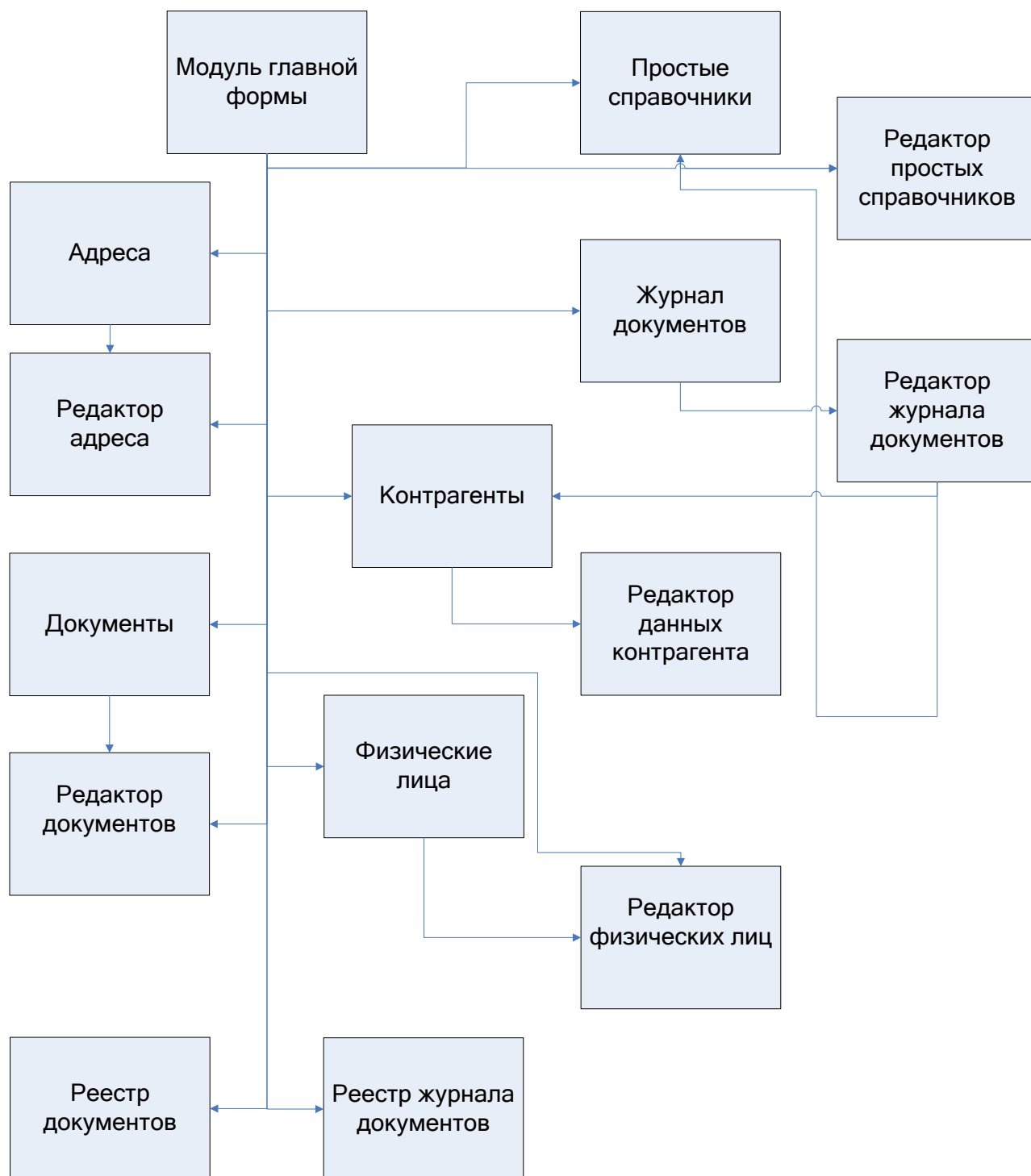


Рисунок 2.4 — Структурная схема ИС

Все модули ИС вызываются из главной формы, редактирование данных происходит в специальной форме «редакторе».

После запуска программы и вывода на экран главной формы приложения пользователь должен выбирать дальнейшие действия: работа со справочниками, документами, отчетами или завершение работы (рисунки 2.5, 2.6) [18].

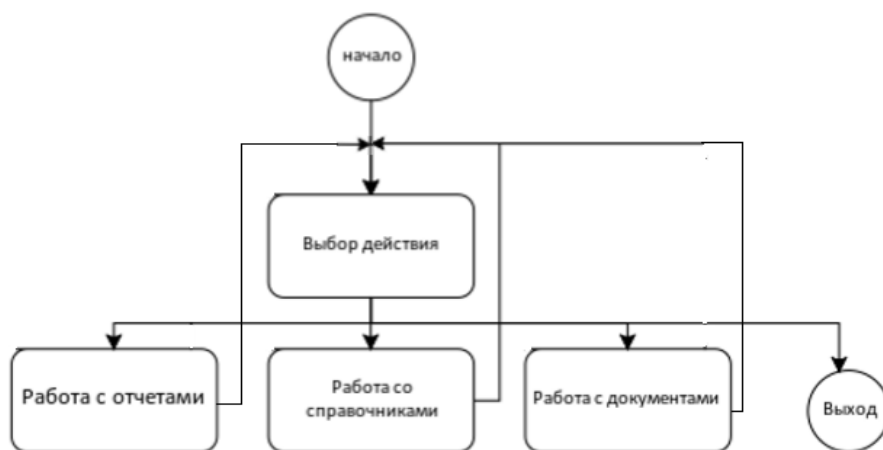


Рисунок 2.5 — Диаграмма прецедентов пользователя (часть 1)

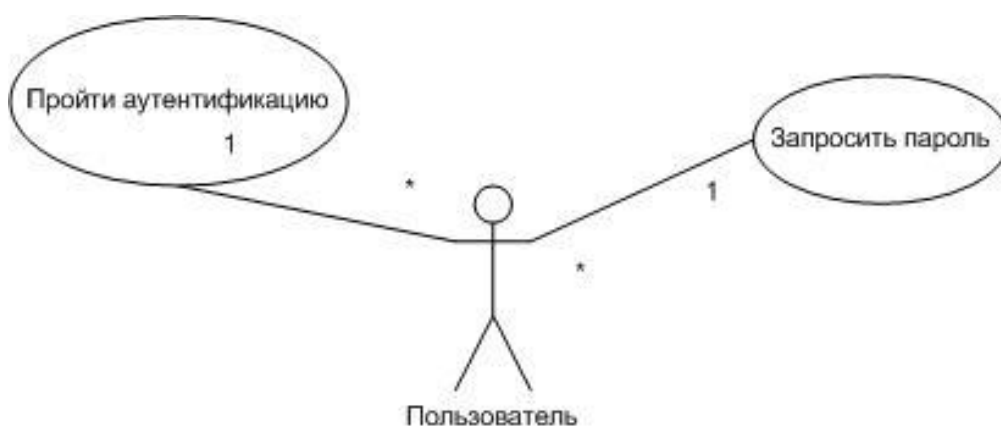


Рисунок 2.6 — Диаграмма прецедентов пользователя (часть 2)

Для более наглядного представления были разработаны «IDEF0 - диаграммы» функций разрабатываемой ИС (рисунок 2.7).

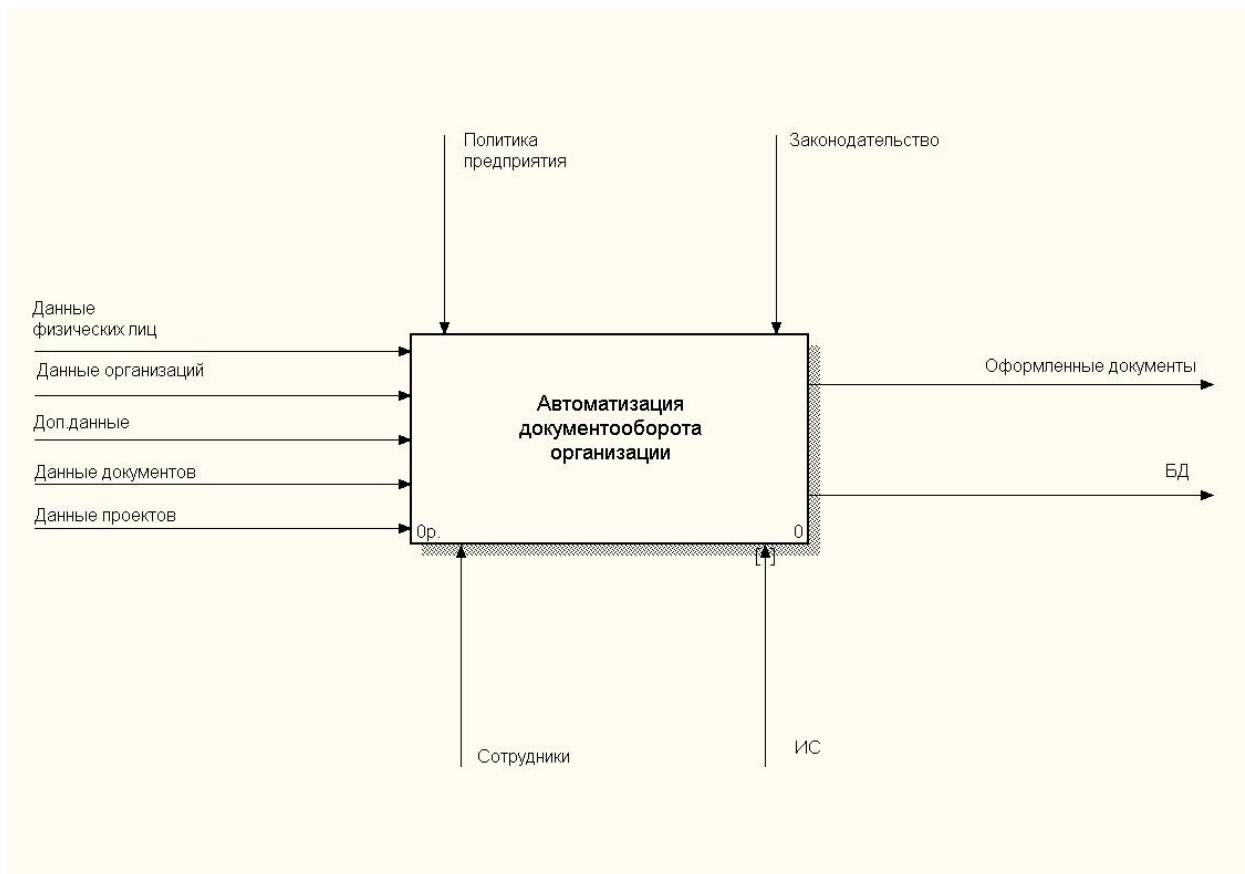


Рисунок 2.7 — Диаграмма организации СЭД «нулевого» уровня

Как видно из рисунка входящими данными для ИС документооборота организации являются:

- данные физических лиц,
- данные организаций,
- дополнительные данные,
- данные документов,
- данные проектов.

Оформление документов ведется посредством ИС. Оформление документов происходит согласно действующему законодательству РФ и политики организации.

Результатом работы являются:

- заполненная БД ИС,
- оформленные документы.

Более подробная детализация СЭД представлена на рисунке 2.8.

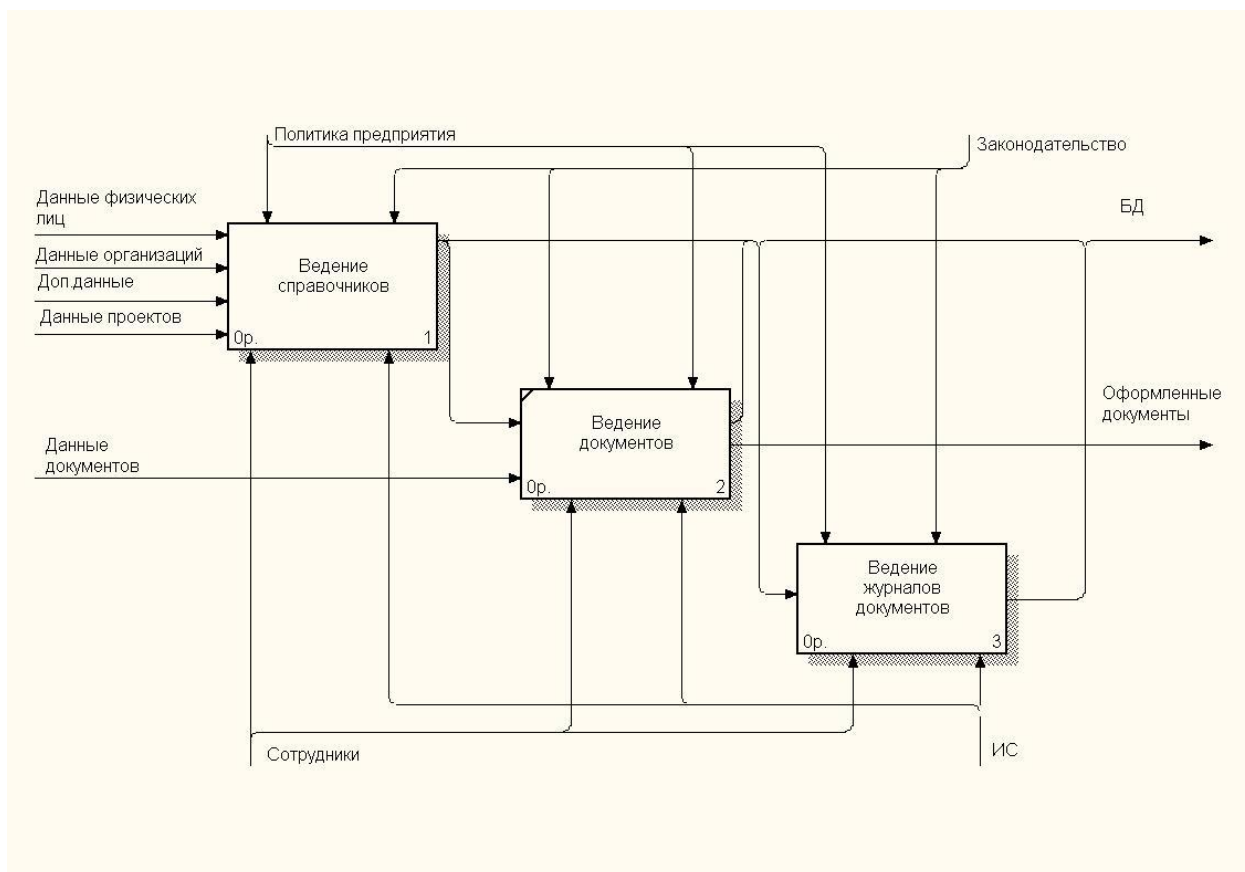


Рисунок 2.8 — Детализация СЭД

Главная задача делится на задачи: «Ведение справочников», «Ведение документов» и «Ведение журнала документов». Все эти задачи выполняются сотрудниками организации в ИС согласно политики предприятия и законов РФ. Результатами задач являются заполненная БД и оформленные документы. В задачах «Ведение документов» и «Ведение журнала документов» используются данные, полученные из задачи «Ведение справочников».

Декомпозиция блока «Ведение справочников» представлена на рисунке 2.9.

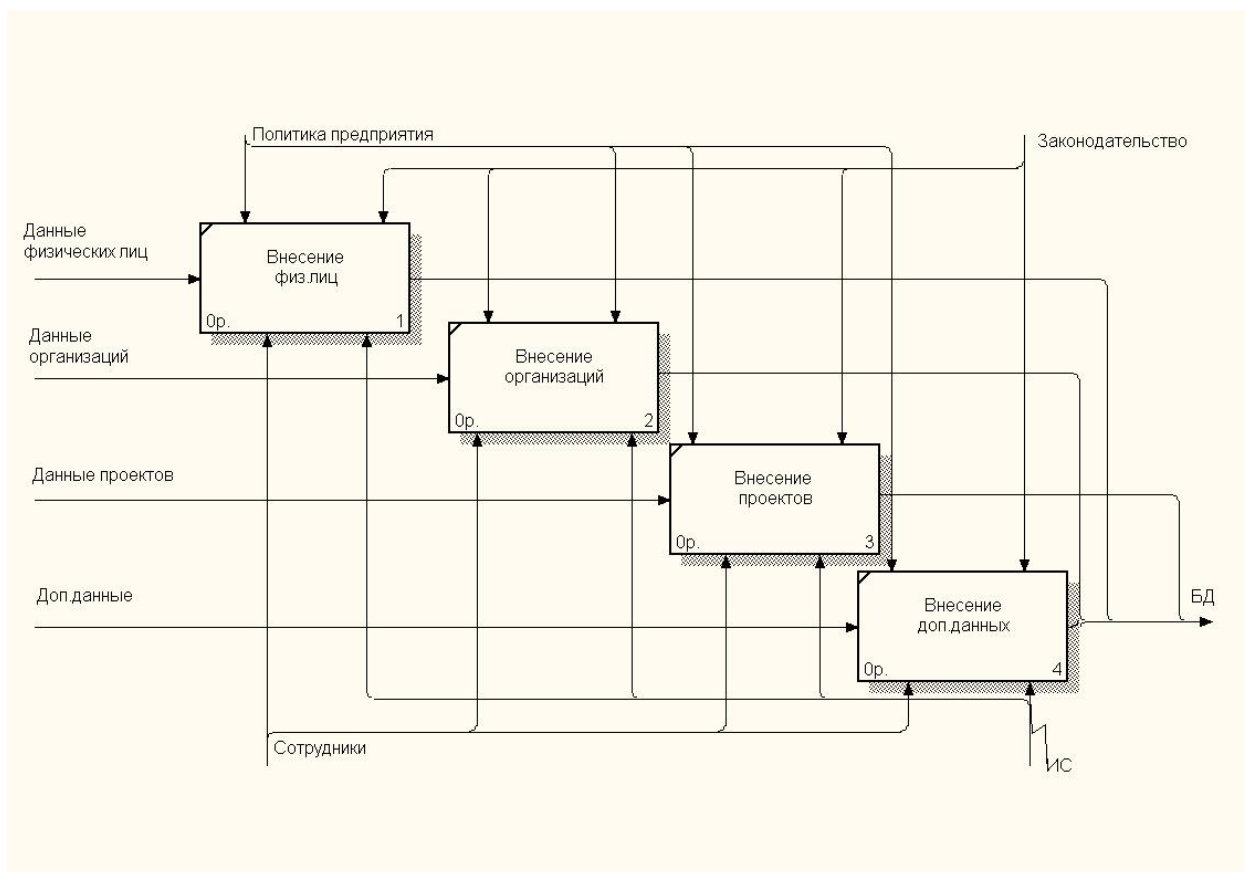


Рисунок 2.9 — Детализация блока «Ведение справочников»

Подзадачами блока «Ведение справочников» являются:

- внесение ФЛ,
- внесение организаций,
- внесение проектов,
- внесение данных дополнительных данных документов.

Все данные вносятся сотрудниками в ИС согласно действующему законодательству РФ и политики организации. Все внесенные данные записываются в БД ИС.

2.3 Архитектура разрабатываемой системы

ИС документоучета должна функционировать на следующем ПК:

- процессор – «Intel i3»,
- DDR4 4 Гб,
- HDD,
- сетевая карта,
- принтер.

Для «многопользовательской» работы требуется применение СУБД. Сервера для СУБД может располагаться на ПК:

- процессор – «Intel i5»,
- DDR4 8 Гб,
- HDD,
- сетевая карта.

Архитектура разрабатываемой системы приведена на рисунке 2.10.

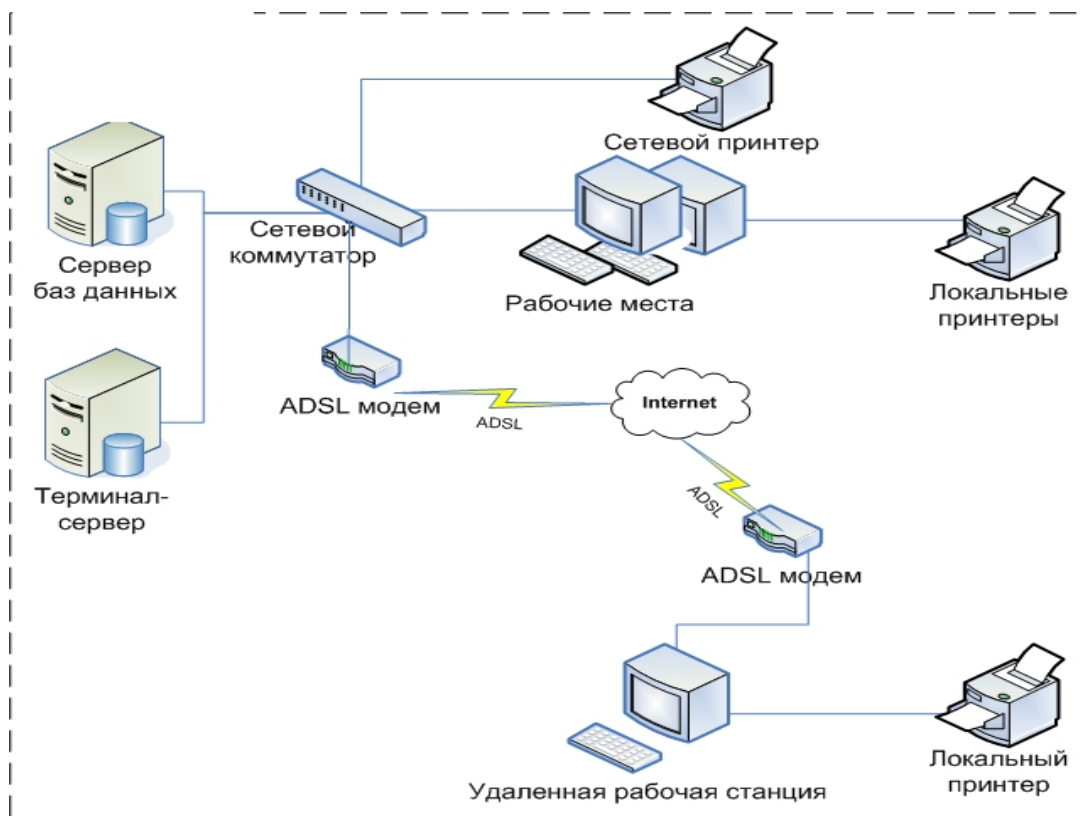


Рисунок 2.10 — Архитектура разрабатываемой системы

2.4 Разработка концептуальной (инфологической) модели данных

Была разработана информационная модель ИС. Информационная модель включает в себя несколько блоков:

- первый показывает процесс редактирования справочников,
- второй отображает процесс внесения документов,
- третий описывает подсистему отчетности.

Разработанная модель показана на рисунке 2.11.

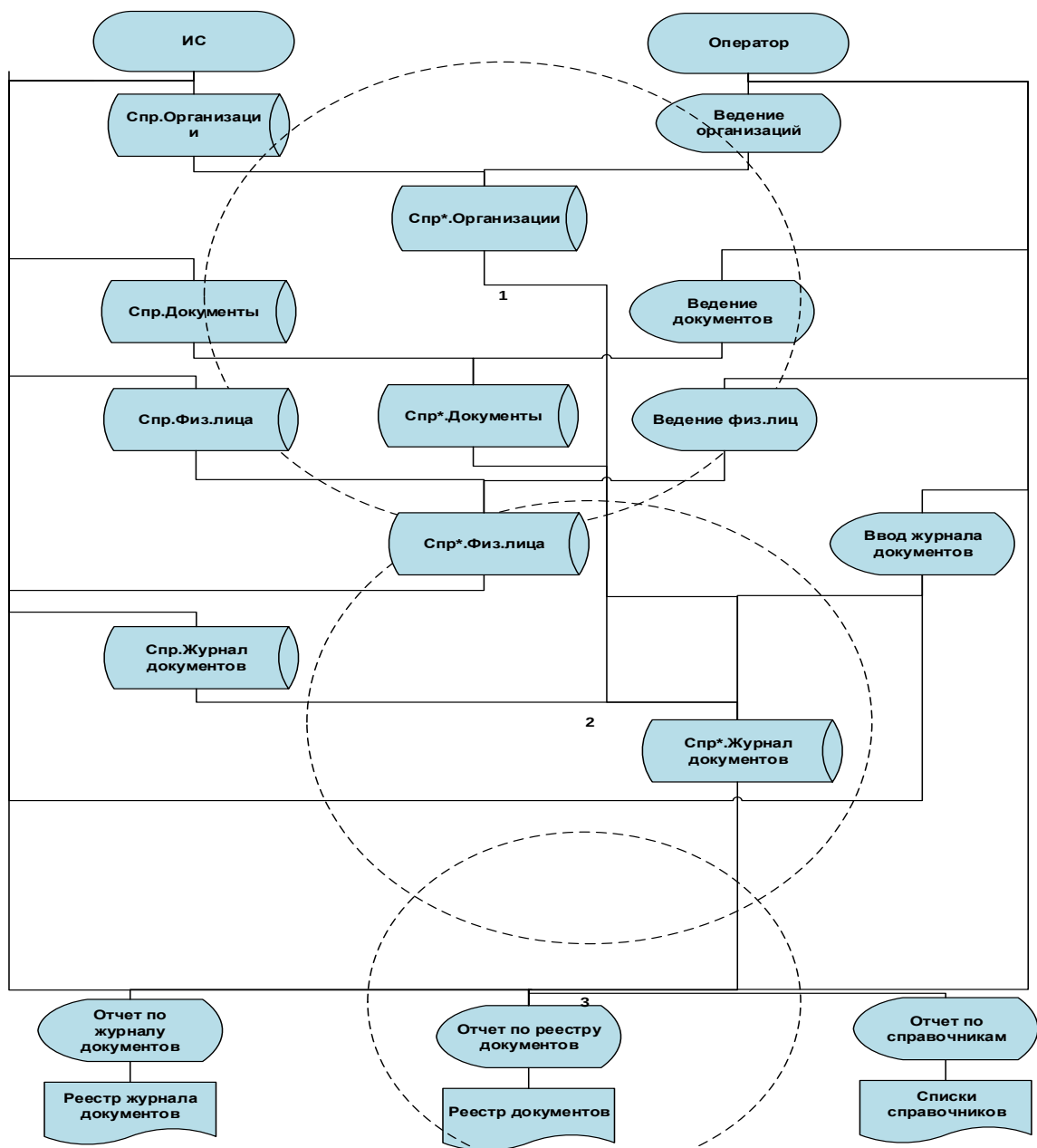


Рисунок 2.11 — Информационная модель

2.5 Выбор системы управления базами данных

Задача СЭД не подразумевает сохранения гигантских объемов информации, так как организация не применяет большого объема используемой информации.

СУБД является объединенным множеством различных программ для организации возможности использования информации в различных БД множеством пользователей. С возможностью обеспечения целостности данных и их безопасного хранения.

Наиболее известными СУБД в мире являются «MySQL» и «MS SQL Server» (рисунок 2.12) [19].

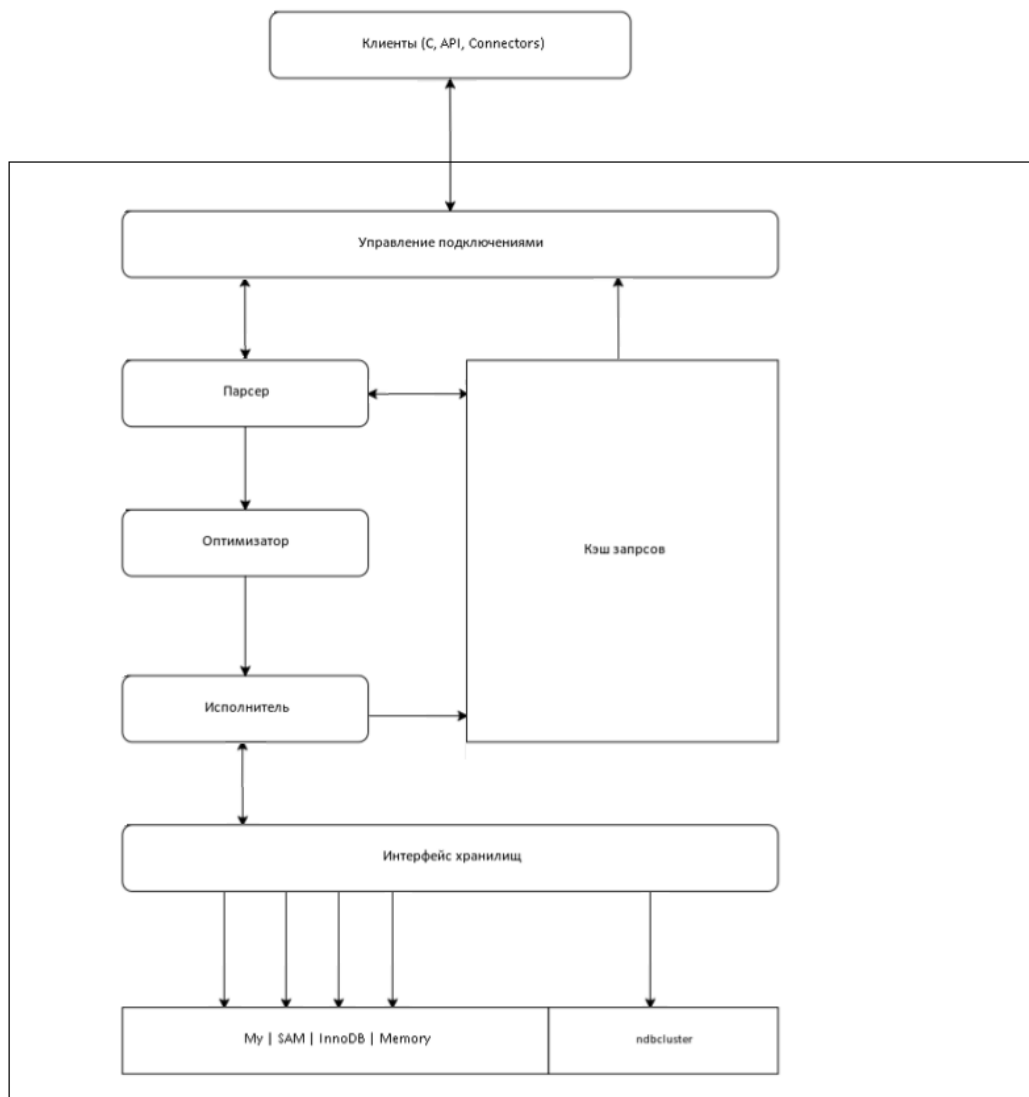


Рисунок 2.12 — СУБД «MySQL»

Для разработки ИС учета документов выбрана СУБД «Microsoft SQL Server» [20]. Единство разработчика СУБД и среды разработки значительно облегчает совмещение БД и интерфейса работы с ней (рисунок 2.13).

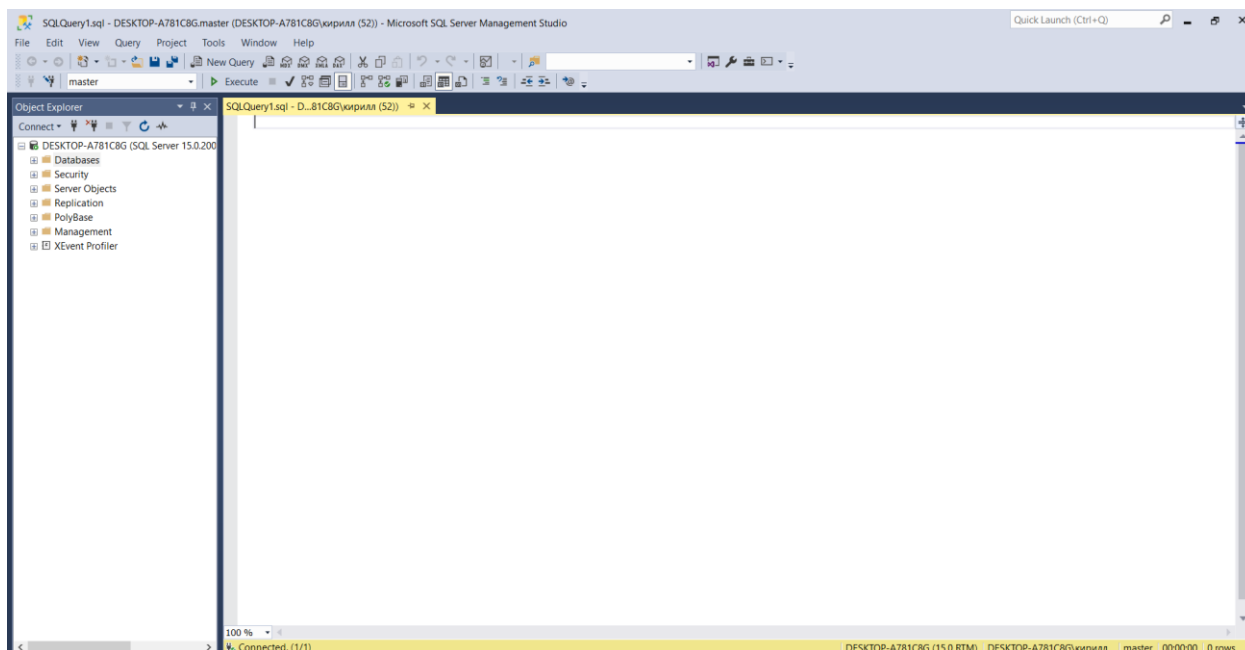


Рисунок 2.13 — СУБД «MS SQL Server»

2.6 Разработка физической модели базы данных

БД программы документооборота является реляционной и состоит из таблиц и хранимых процедур. Состав таблиц и их полей указан на рисунках 2.14 – 2.17.

	Таблица	Столбец	Тип	Макс. Длина байт	М.б. NULL	ИД
1	Address	ID	int	4	0	1
2	Address	ClientsID	int	4	1	0
3	Address	Name	varchar	254	1	0
4	Address	FullName	varchar	254	1	0
5	Class	ID	int	4	0	1
6	Class	Treeld	int	4	1	0
7	Class	Object	varchar	50	1	0
8	Class	Name	varchar	254	1	0
9	Class	FullName	varchar	254	1	0
10	ClassArr	ID	int	4	0	1
11	ClassArr	ClassId	int	4	1	0
12	ClassArr	ElementName	varchar	50	1	0
13	ClassArr	ElementId	int	4	1	0
14	ClassArr	Name	varchar	254	1	0
15	ClassArr	FullName	varchar	254	1	0
16	ClassName	ID	int	4	0	1
17	ClassName	Object	varchar	50	1	0
18	ClassName	Name	varchar	254	1	0
19	ClassName	FullName	varchar	254	1	0
20	ClassName	Tree	bit	1	1	0
21	ClassName	NotDel	bit	1	1	0
22	Clients	ID	int	4	0	1
23	Clients	Treeld	int	4	1	0
24	Clients	Name	varchar	100	1	0
25	Clients	FullName	varchar	254	1	0
26	Clients	INN	varchar	15	1	0
27	Clients	KPP	varchar	15	1	0
28	Clients	DateReg	datetime	8	1	0
29	Clients	UrFiz	bit	1	1	0
30	Clients	Telephon	varchar	100	1	0
31	Contacts	ID	int	4	0	1
32	Contacts	TypeContactId	int	4	1	0
33	Contacts	TypePeopleId	int	4	1	0
34	Contacts	PeoplesId	int	4	1	0
35	Contacts	ElementName	varchar	50	1	0
36	Contacts	ElementId	int	4	1	0

Рисунок 2.14 — Состав таблиц (часть 1)

	Таблица	Столбец	Тип	Макс. Длина байт	М.б. NULL	ИД
37	Contacts	Name	varchar	254	1	0
38	Contacts	FullName	varchar	254	1	0
39	Documents	ID	int	4	0	1
40	Documents	Treeld	int	4	1	0
41	Documents	PeoplesId	int	4	1	0
42	Documents	VidDocsId	int	4	1	0
43	Documents	StatusDocsId	int	4	1	0
44	Documents	PodrId	int	4	1	0
45	Documents	ProjectId	int	4	1	0
46	Documents	NumDoc	varchar	10	1	0
47	Documents	DateDoc	datetime	8	1	0
48	Documents	Name	varchar	254	1	0
49	Documents	FullName	varchar	254	1	0
50	FieldsJomal	ID	int	4	0	1
51	FieldsJomal	TypeDocsId	int	4	0	0
52	FieldsJomal	Name	varchar	254	1	0
53	FieldsJomal	FieldName	varchar	100	1	0
54	Jomal	ID	int	4	0	1
55	Jomal	ClientsId	int	4	1	0
56	Jomal	AddressId	int	4	1	0
57	Jomal	PeoplesId	int	4	1	0
58	Jomal	PeoplesIstId	int	4	1	0
59	Jomal	PodrId	int	4	1	0
60	Jomal	ProjectId	int	4	1	0
61	Jomal	VidDocsId	int	4	1	0
62	Jomal	StatusDocsId	int	4	1	0
63	Jomal	TypeDocsId	int	4	0	0
64	Jomal	NumDoc	varchar	10	1	0
65	Jomal	DateDoc	datetime	8	1	0
66	Jomal	Status	varchar	10	1	0
67	Jomal	Name	varchar	254	1	0
68	Jomal	FullName	varchar	254	1	0
69	JomalTabl	ID	int	4	0	1
70	JomalTabl	JomalId	int	4	1	0
71	JomalTabl	DocumentsId	int	4	1	0
72	JomalTabl	PeoplesId	int	4	1	0

Рисунок 2.15 — Состав таблиц (часть 2)

	Таблица	Столбец	Тип	Макс. Длина байт	М.б. NULL	ИД
73	JomalTabl	Name	varchar	254	1	0
74	JomalTabl	FullName	varchar	254	1	0
75	JomalTabl	FIRez	bit	1	1	0
76	Passport	ID	int	4	0	1
77	Passport	PeoplesId	int	4	1	0
78	Passport	VidPassId	int	4	1	0
79	Passport	Name	varchar	254	1	0
80	Passport	FullName	varchar	254	1	0
81	Passport	Num	varchar	10	1	0
82	Passport	Ser	varchar	10	1	0
83	Passport	Podr	varchar	10	1	0
84	Passport	Who	varchar	800	1	0
85	Passport	DatePass	datetime	8	1	0
86	Peoples	ID	int	4	0	1
87	Peoples	Treeld	int	4	1	0
88	Peoples	F	varchar	20	1	0
89	Peoples	I	varchar	20	1	0
90	Peoples	O	varchar	20	1	0
91	Peoples	Sex	varchar	1	1	0
92	Peoples	Name	varchar	100	1	0
93	Peoples	FullName	varchar	254	1	0
94	Peoples	INN	varchar	15	1	0
95	Peoples	DR	datetime	8	1	0
96	Peoples	Telephon	varchar	100	1	0
97	Peoples	AddressFact	varchar	100	1	0
98	Peoples	AddressUr	varchar	100	1	0
99	Prop	ID	int	4	0	1
100	Prop	PropValueId	int	4	1	0
101	Prop	ElementName	varchar	50	1	0
102	Prop	ElementId	int	4	1	0
103	Prop	Name	varchar	254	1	0
104	Prop	FullName	varchar	254	1	0
105	PropValue	ID	int	4	0	1
106	PropValue	ClassId	int	4	1	0
107	PropValue	Name	varchar	254	1	0
108	PropValue	FullName	varchar	254	1	0

Рисунок 2.16 — Состав таблиц (часть 3)

	Таблица	Столбец	Тип	Макс. Длина байт	М.б. NULL	ИД
109	SaveDocs	ID	int	4	0	1
110	SaveDocs	ElementName	varchar	50	1	0
111	SaveDocs	ElementId	int	4	1	0
112	SaveDocs	Name	varchar	254	1	0
113	SaveDocs	FullName	varchar	254	1	0
114	SaveDocs	FileName	varchar	254	1	0
115	SaveDocs	Doc	varbinary	-1	1	0
116	Tree	ID	int	4	0	1
117	Tree	OwnerId	int	4	1	0
118	Tree	Tree_Object	varchar	50	1	0
119	Tree	Name	varchar	100	1	0
120	Tree	FullName	varchar	254	1	0
121	Tree	ClassObject	varchar	50	1	0
122	TypeDocs	ID	int	4	0	1
123	TypeDocs	Name	varchar	100	1	0
124	TypeDocs	FullName	varchar	254	1	0
125	TypeDocs	WareSaldo	int	4	1	0
126	TypeDocs	MoneySaldo	int	4	1	0
127	Users	ID	int	4	0	1
128	Users	Name	varchar	254	1	0
129	Users	Login	varchar	50	1	0
130	Users	PASS	varchar	50	1	0
131	Users	Adm	bit	1	1	0

Рисунок 2.17 — Состав таблиц (часть 4)

На рисунке 2.18 изображен запрос для получения структуры БД.

```
SQLQuery_Структур...ker-ПК\Joker (55)) X
select Tab.Name as Таблица, Col.Name as Столбец, Typ.name as Тип,
       Col.max_length as МаксДлинаБайт, Col.is_nullable as МожетБытьNULL,
       Col.is_identity as ИД
from sys.columns Col
     join sys.tables Tab on Tab.Object_id = Col.Object_id
     join sys.types Typ on Typ.system_type_id = Col.system_type_id
order by Tab.Name
```

Рисунок 2.18 — «SQL»-запрос для получения структуры БД

Использование структурных элементов БД показано в таблице 2.1.

Таблица 2.1 — Применение таблиц ИС

Таблица	Назначение таблицы
«Address»	Адреса организаций. Таблица связана с таблицей «Clients»
«Class»	Простые таблицы
«ClassArr»	Характеристики для различных справочников
«ClassName»	Описание полей, созданных оператором и при создании БД простых таблиц
«Clients»	Различные контрагенты
«FieldsJornal»	Скрытые поля документов неотображаемые в шапке открытого в диалоговой форме документа
«Contacts»	Связь физического лица с организацией
«Documents»	Данные документов
«Jornal»	Журнал документов
«JornalTabl»	Табличные части журнала документов
«Passport»	Данные различных документов физических лиц
«Peoples»	Справочник «Физические лица»
«Prop»	Дополнительные свойства
«PropValue»	Значения дополнительных свойств справочников
«SaveDocs»	Хранилище документов
«TypeDocs»	Типы документов

Таблица 2.1 — продолжение

Таблица	Назначение таблицы
«Tree»	Структура папок для всех справочников, имеющих «иерархическую» структуру
«Users»	Операторы ИС и их права доступа на работу с программой

Итоговая архитектура разработанной системы, состав таблиц БД и их связи указаны на рисунке 2.19.

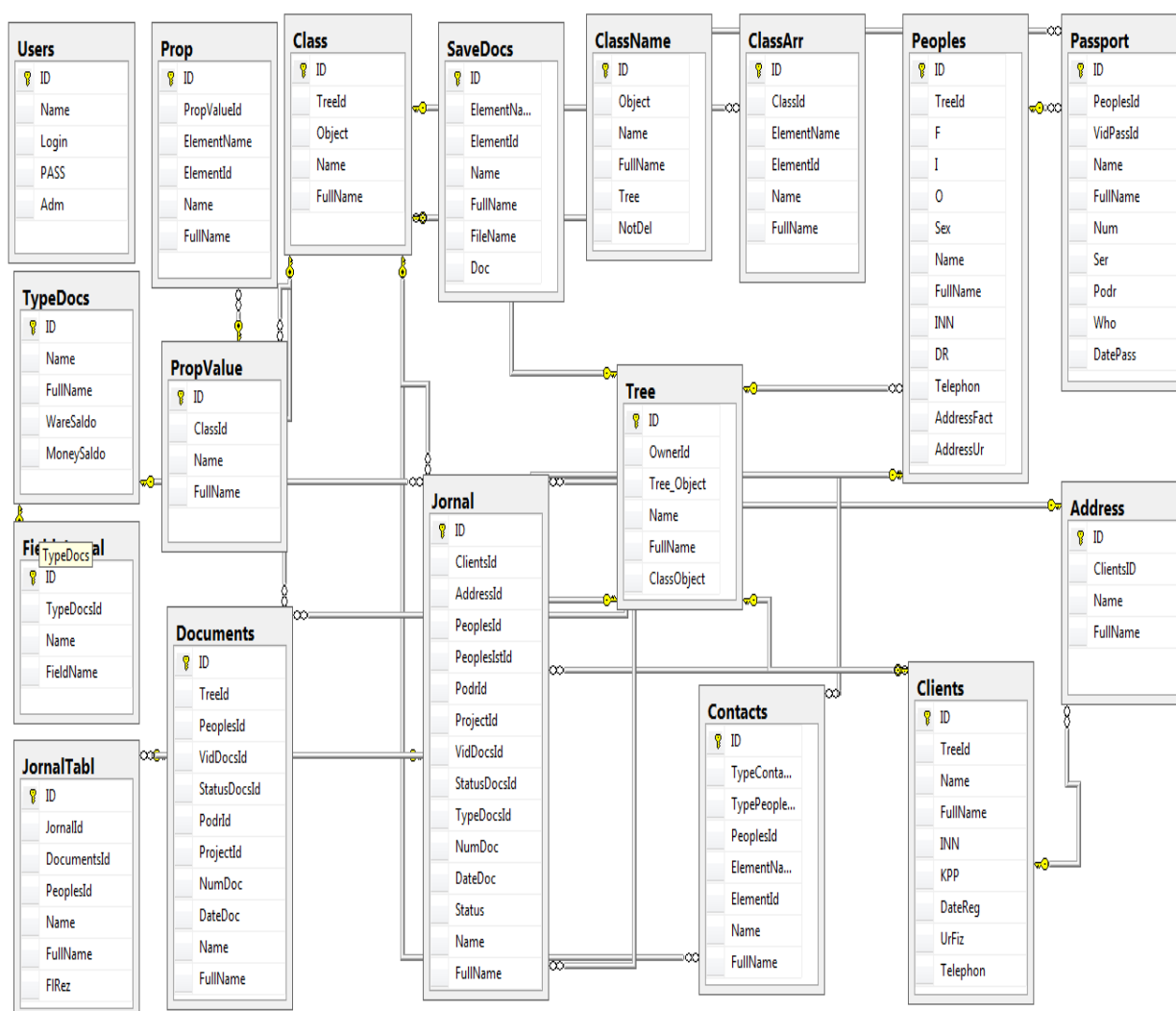


Рисунок 2.19 — Модель данных

ER-модель БД представлена на рисунке 2.20.

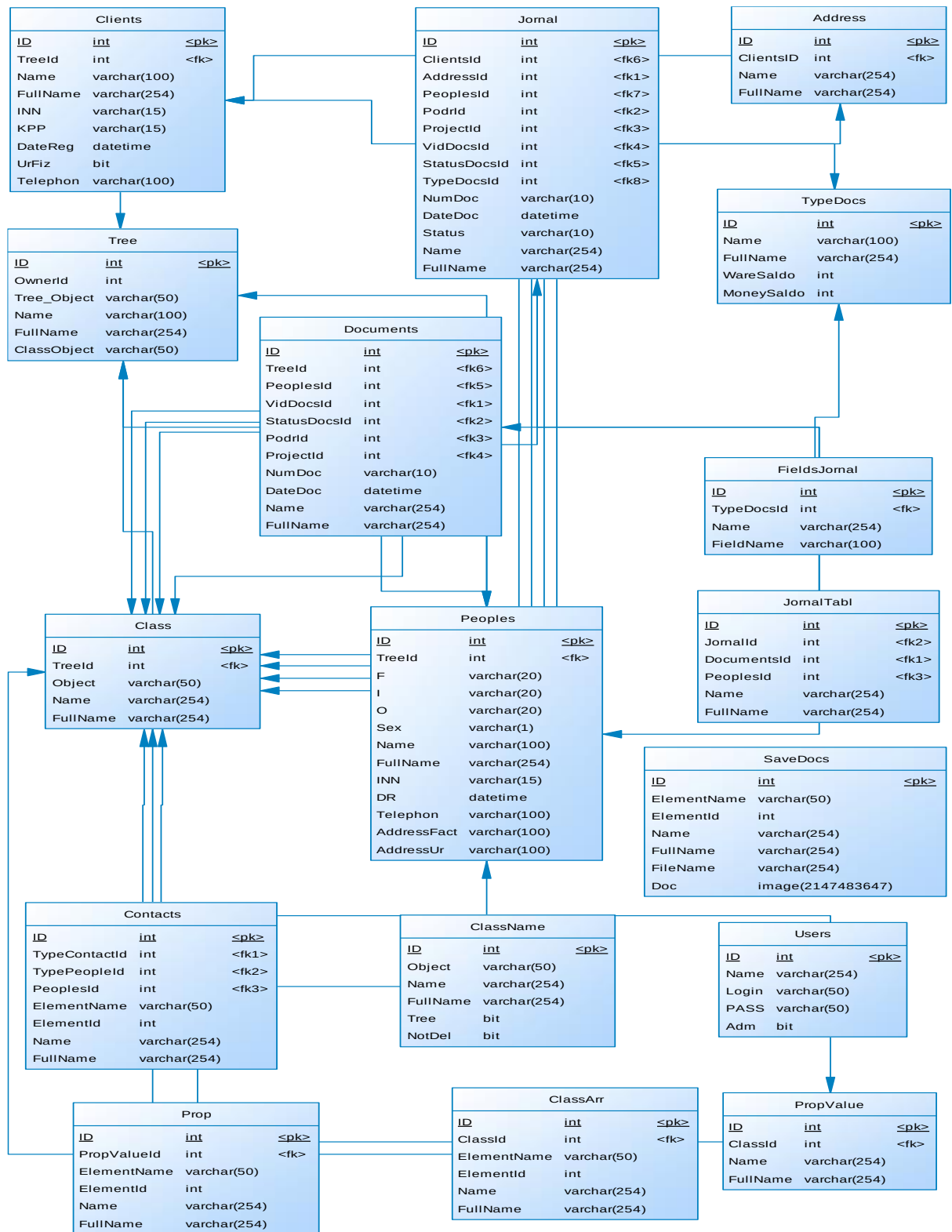


Рисунок 2.20 — ER-модель базы данных

3 Реализация системы

3.1 Общая концепция разрабатываемой системы

Функции ИС подразделяются на «основные» и «служебные».

На рисунке 3.1 показано дерево функций ИС учета документов.

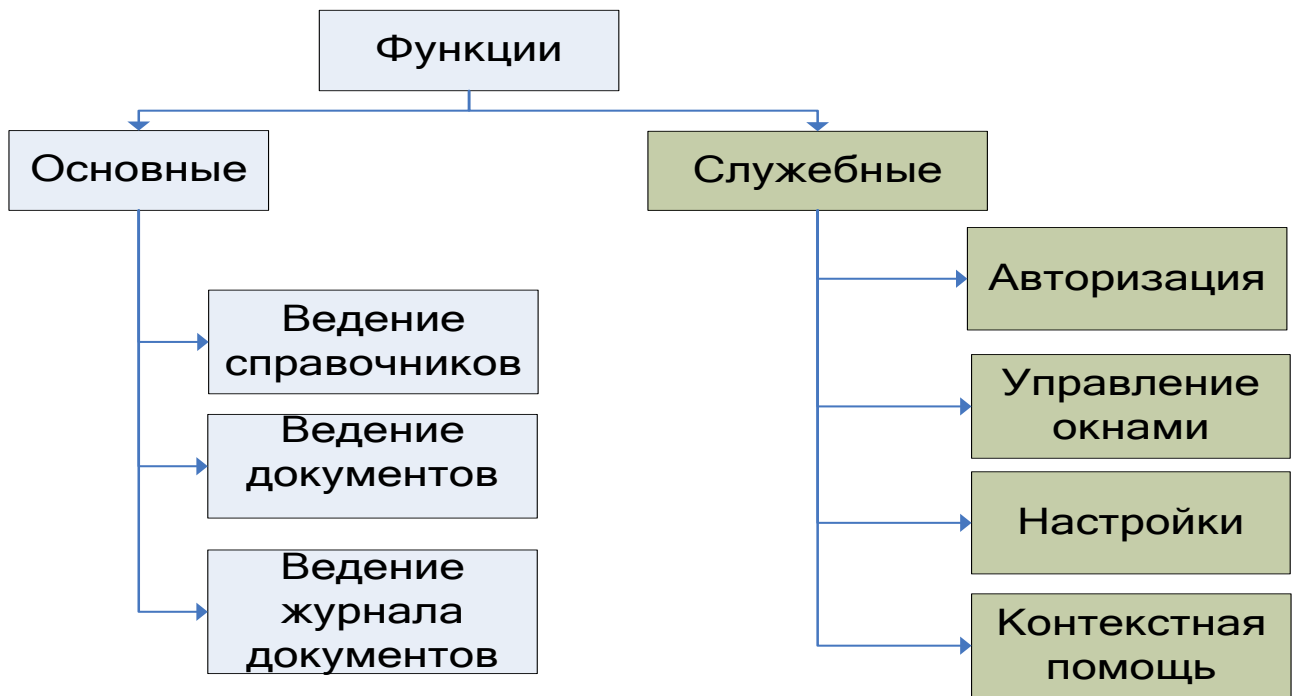


Рисунок 3.1 — Дерево функций

3.2 Создание прикладных объектов

В IDE «Visual Studio» для языка программирования C# основным визуальным объектом является «Форма» [21]. Форма хранит объекты интерфейса и программный код, относящийся к ним. В разработанном приложении присутствуют: главная форма (ГФ) «FMain» - это контейнер «MDI», форма для авторизации пользователей «FReg», форма для работы со всеми таблицами БД «FTreeGrid», форма настройки периода – «FSetupPeriod» и форма диалогового редактирования данных таблиц «FEdit» (рисунок 3.2).

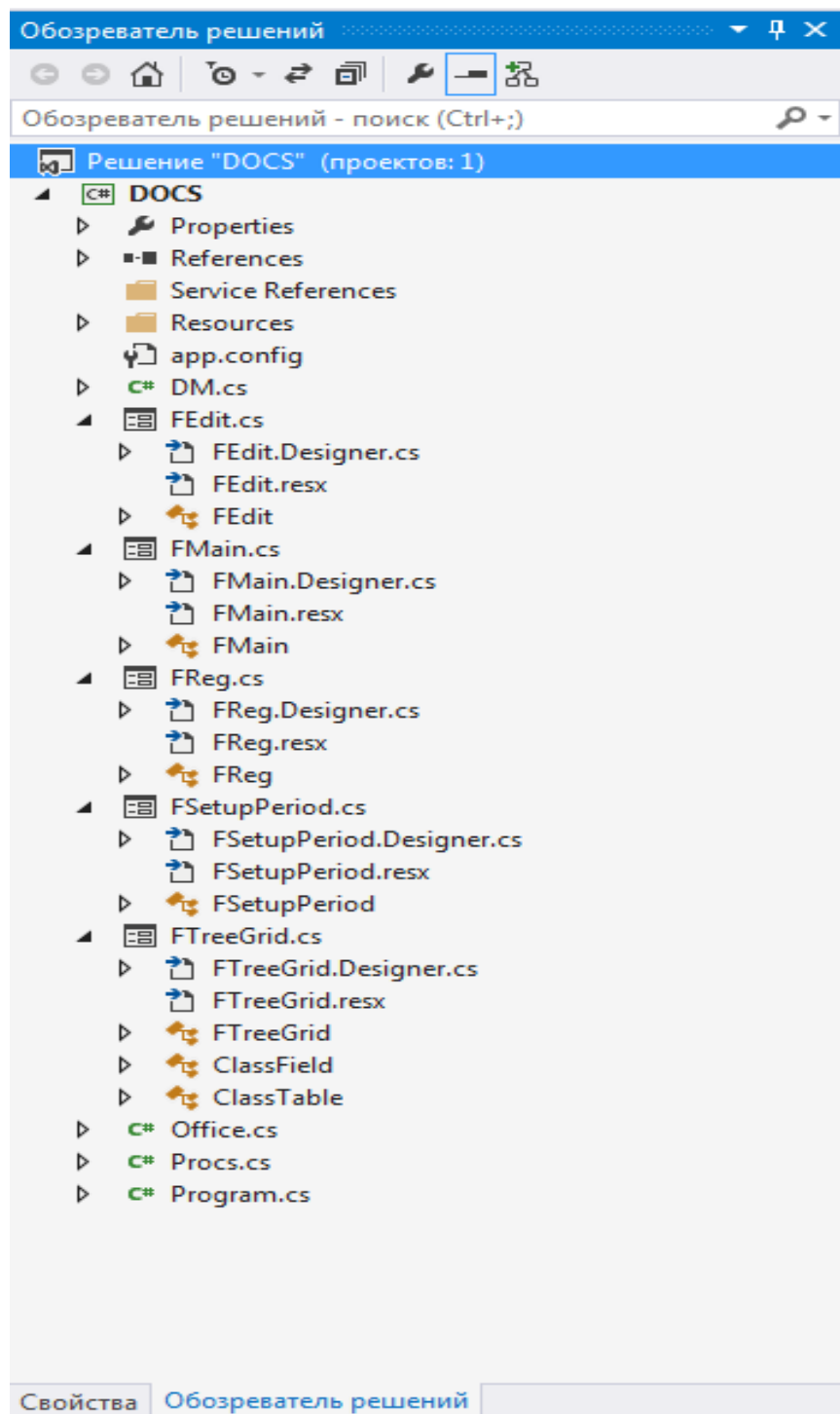


Рисунок 3.2 — Модули приложения

Вызов функций ИС выполняется через главное меню (ГМ) программы или кнопки панели инструментов ИС (рисунки 3.3, 3.4).

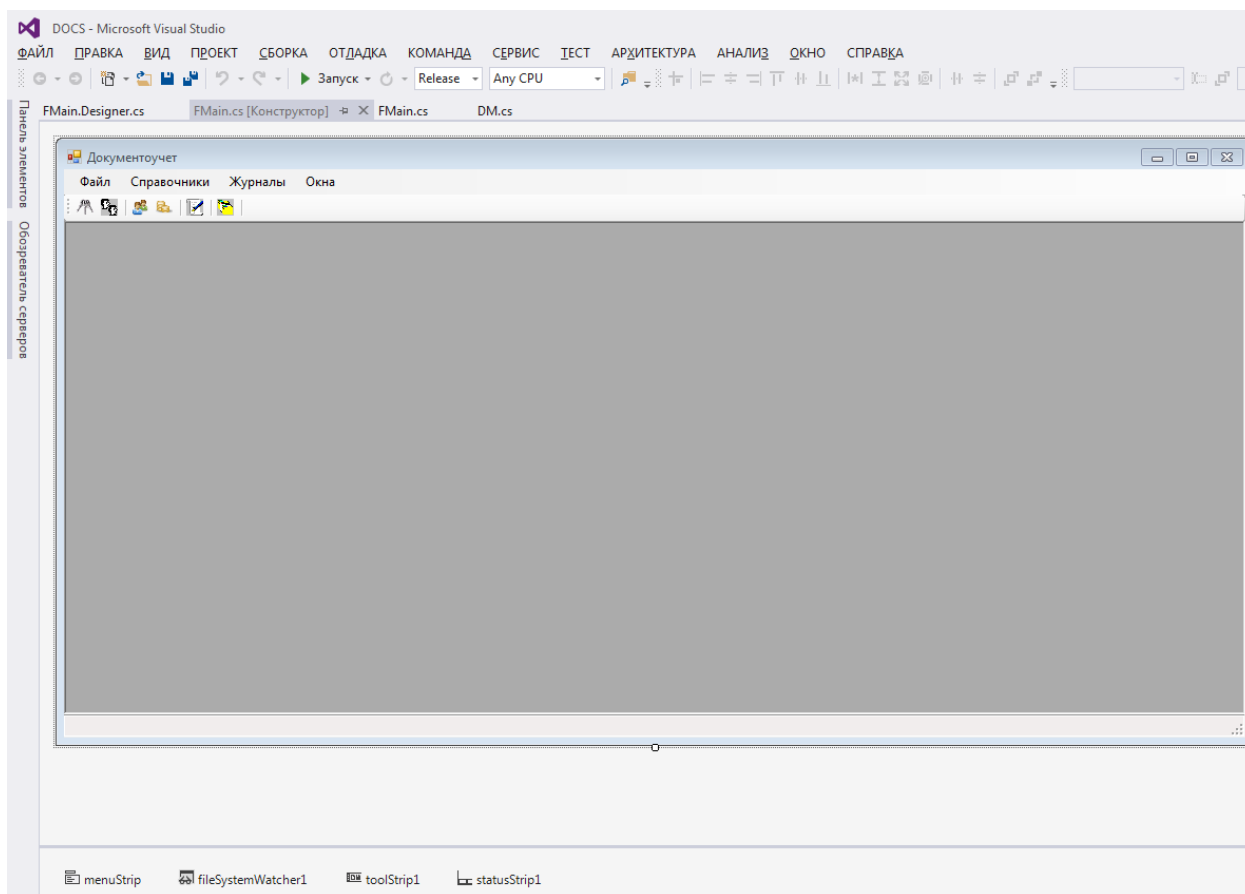


Рисунок 3.3 — Макет «FMain»

В форме авторизации пользователь вносит свои учетные данные для работы с ИС (по умолчанию 1, 1), модуль проверяет право доступа к ИС (рисунок 3.4).

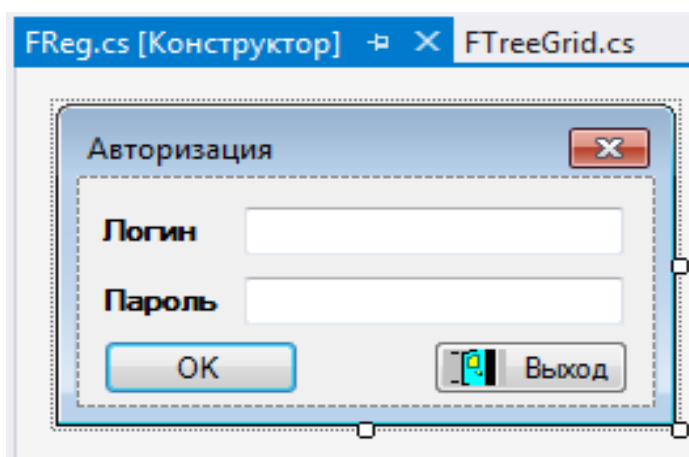


Рисунок 3.4 — Макет «FReg»

«FTreeGrid» - важная форма для редактирования и просмотра данных таблиц (рисунок 3.5). Основным реквизитом, получаемым формой, является ссылка на класс «ClassTable».

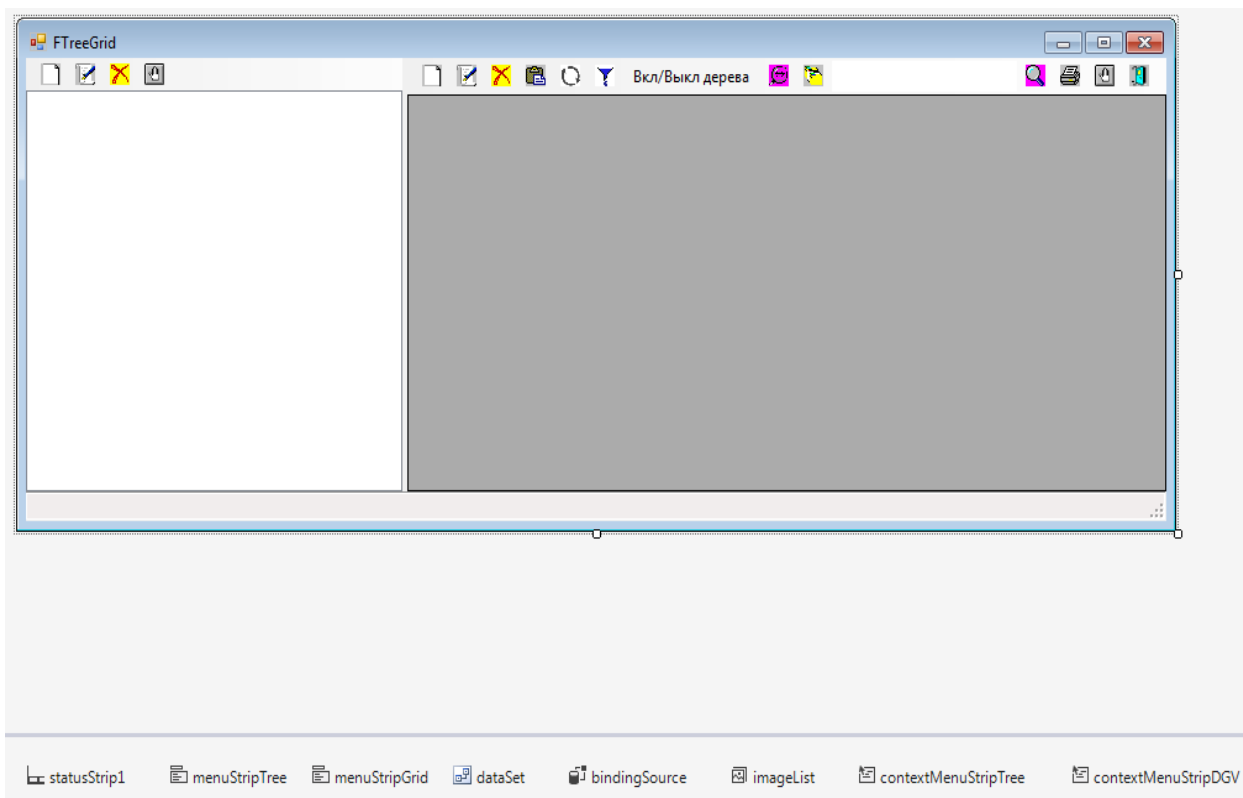


Рисунок 3.5 — Макет «FTreeGrid»

Класс дает описание всех таблиц БД и методов работы с ними (рисунки 3.6,3.7).

```

public class ClassTable
{
    public string ClassObject; //Для работы с таблицей Class если простая таблица имеет дерево

    public string BeginSQLText; //текст запроса выполняющегося при загрузки формы (LOAD)

    public bool FlReadOnly; // только для чтения
    public bool NotAllowUserToAddRows; // ЗАПРЕТИТЬ добавлять новые строки в ГРИДЕ
    public bool NotAllowUserToDeleteRows; // ЗАПРЕТИТЬ удалять строки в ГРИДЕ

    public bool FlTree; // наличие дерева (FALS ЕСТЬ ДЕРЕВО)
    public bool FlSelect; // true - форма для выбора
    public bool FlSelectTree; // true - форма для выбора дерева
    public bool MinusTree; // Настройка списка в ГРИДЕ весь или по группам (TRUE - весь)

    public string ConnectString; // подключение к БД

    public string TextForm; // заголовок формы
    public string TableName; // название основной таблицы
    public string SQLText; //текст запроса
    public string PrimaryKeyName; // имя ключа таблицы

    public int LocateID; // для позиционирования на элемент по ID
    public string LocateColumn; // активный столбец после позиционирования
    // ПОИСК
    public string FindStr; // строка для поиск
    public string FindColumn; // название столбца для поиска

    public string Pref; // префикс таблицы ( C.TreeId - префикс C.)

    public string StrWhereGlobal; // строка глобального условия (без where)
    public string StrWhereLocal; // строка локального условия (без where)
    public string StrWhereTree; // строка локального условия для вхождение в дерево
    public string StrWherePeriod; // строка условия по периоду (DateDoc)

    public ArrayList ArrFields; // массив полей, элементы типа ClassField

    // массив SQL запросов для удаления (или обнуления связи) из связанных таблиц ...
    public ArrayList ArrDelete;
    public ArrayList ArrCheckDelete; // Проверки перед удалением

    // поля которые добавляются к каждой записи таблицы необходимы для подчиненного справочника ...
    public ArrayList ArrInsertName;
    public ArrayList ArrInsertValue;

    public ArrayList ArrInsertType;

    // Массив с подчиненными, данной таблице, справочниками ...
    public ArrayList ArrTables;
    public ArrayList ArrTablesFK; //массив с именем внешнего ключа (ClientsID)
    ссылка 0
    public ClassTable()...
    ссылка 15
    public ClassTable(string sqltext)...
    ссылка 61
    public ClassTable(ClassTable Src)...

    ссылка 0
    public void Dispose()...
    public void SetField(string Name, string NameRus, int Width, bool flUpd, bool ReadOnly, bool Visible) ...
    // записать поле в массив полей таблицы
    ссылка 88
    public void SetField(ClassField cf)...
    // записать SQL DELETE (UPDATE) запрос в массив
    ссылка 0
    public void SetSQLDelete(string str)...
    // записать SQL проверки перед удалением
    ссылка 0
    public void SetSQLCheckDelete(string str)...
    // записать часть запроса SQL INSERT

```

Рисунок 3.6 — Класс «ClassTable» часть 1

```

ссылка 33
public void SetSQLInsert(string strName, string strValue, Type type)...
// очистить массив
ссылка 29
public void ClearSQLInsert()...

//удалить или обнулить ссылки на удаляемый элемент
ссылка 2
public bool SQLDeleteEx(int ID)...

//Проверка перед удалением, true можно удалять
ссылка 1
public bool SQLCheckDelete(int ID)...
// Запись массива подчиненных таблиц
ссылка 13
public void SetTables(ClassTable ct, string FN)...
// Поиск поля типа ClassField по наименованию
ссылка 22
public ClassField FindField(string Name)...
// Поиск поля типа ClassField по NameIDOwner ( поиск подчиненного справочника)
ссылка 4
public ClassField FindFieldOnNameIDOwner(string NameIDOwner)...

ссылка 27
public int FindFieldNumber(string Name)...

```

Рисунок 3.7 — Класс «ClassTable» часть 2

Класс «ClassField» хранит описание полей таблиц (рисунок 3.8).

```

#region КЛАСС ПОЛЯ ТАБЛИЦЫ
ссылка 139
public class ClassField
{
    public string Name;           //название поля как в SQL - запросе
    public string NameRus;        //название поля на русском
    public int Width;             //ширина столбца
    public bool flUpd;            //признак изменения поля в БД
    public bool ReadOnly;         //ReadOnly для столбца DGV
    public bool Visible;          //видимость столбца

    public string ToolTipText;    // подсказка для поля

    public int Butt;              // наличие кнопки в колонке 1 - кнопка, 2- CheckBox

    public string NameIDEx;       // Поле в основной таблице для связи с подчиненной

    public string NameIDOwner;    // Код владельца если в одной форме ред-я 2-а связанных справочника
                                // например шапка документа поля Организации и адреса ...
    public ClassTable ClTableEx;  // описание под-таблицы
    public Type type;
    public bool flInfField;       // true - информационное поле, отсутствует в таблице. не может изменять БД
                                // используется для вывода информации или связи с владельцем ...

    public bool flNotPref;        //ненужен префикс( таблица связана с основной через другую таблицу
ссылка 0
    public ClassField(string Name,
                        string NameRus,
                        int Width,
                        bool flUpd,
                        bool ReadOnly,
                        bool Visible)...

ссылка 88
    public ClassField()...
}
#endregion

```

Рисунок 3.8 — Класс «ClassField»

Для добавления или редактирования записей «FTreeGrid» может вызывать форму редактирования – «FEdit».

Если редактирование выполняется в «простых» таблицах, то соответствующая форма не используется, и весь процесс выполняется в форме-списке.

Форма «FSetupPeriod» требуется для задания периода, в рамках которого будут выводиться информация в журнале документов (рисунок 3.9). Форма «FSetupPeriod» требуется для задания периода, информация за который будут выводиться в журнале документов.

FMMain.Designer.cs FMMain.cs [Конструктор] FMMain.cs

Настройка периода

Установлен период:

Начало интервала

Окончание интервала

☐ Без ограничения

☐ Начало года

☐ Начало месяца

☐ Начало дня

☐ 01.01.2022

☐ Без ограничения

☐ Окончание года

☐ Окончание месяца

☐ Окончание дня

☐ 31.12.2022

OK Отмена

Рисунок 3.9 — Макет «FSetupPeriod»

«FEdit» применяется при диалоговом редактировании данных таблиц (рисунок 3.10).

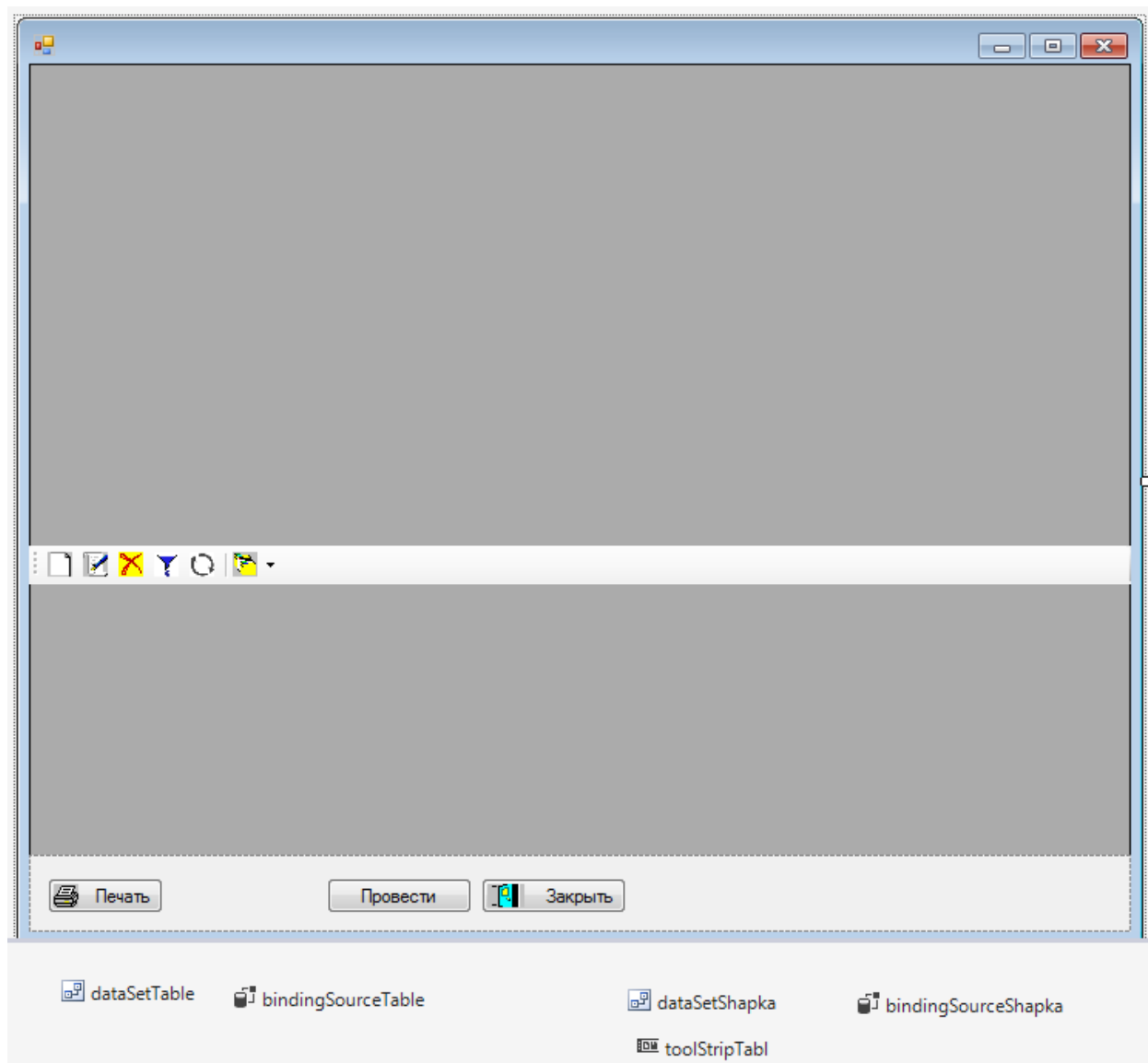


Рисунок 3.10 — Макет «FEdit»

«FEdit» применяется при диалоговом редактировании данных таблиц.

Модуль «DM.cs» применяется при работе с БД, они и содержит следующие классы: «iniSettings» и «DM». «iniSettings» хранит способ присоединения к СУБД (рисунок 3.11).

```

// КЛАСС с настройками
ссылка 5
} public class iniSettings
{
    public string ConnectionString; //строка подключения к SQL server (БЕЗ названия БД)
    public string Aliase;           // название БД
}

```

Рисунок 3.11 — Класс «iniSettings»

Класс «DM» организует работу с БД и всеми её таблицами и их полями (Рисунок 3.12).

```

#region Jornal
CTJornal = new ClassTable("select J.*, T.Name as TypeDocsName, C.Name as ClientsName, A.Name as AddressName, P.Name as PeoplesName, " +
    " PO.Name as PodrName, VD.Name as VidDocsName, SD.Name as StatusDocsName, " +
    " Pr.Name as ProjectName " +
    " from Jornal J " +
    " left join Clients C on C.ID = J.ClientsID " +
    " left join Address A on A.ID = J.AddressID " +
    " left join TypeDocs T on T.ID = J.TypeDocsID " +
    " left join Class PO on PO.ID = J.PodrId " +
    " left join Peoples P on P.ID = J.PeoplesID " +
    " left join Class Pr on Pr.ID = J.ProjectId " +
    " left join Class VD on VD.ID = J.VidDocsId " +
    " left join Class SD on SD.ID = J.StatusDocsId ");
CTJornal.TableName = "Jornal"; // название таблицы
CTJornal.TextForm = "Журнал документов"; // заголовок формы ( или название списка подчиненных справочников)

CTJornal.Pref = "J.";
CTJornal.FlTree = true;

Описание полей таблицы
#endregion

```

Рисунок 3.12 — Выборка данных для журнала документов

Данные для отчетов в ИС выбираются из БД благодаря SQL – запросам, а выводятся в пакет «MS Excel» с помощью класса «Office». Печать в любом списке выполняется по нажатию на кнопку «Печать», обработчик нажатия которой представлен на рисунке 3.13.

```

// ПЕЧАТЬ
ссылка 1
private void toolStripMenuItemPrint_Click(object sender, EventArgs e)
{
    Office off = new Office();
    off.ConnectionString = CT.ConnectionString;
    off.Connect(false);
    off.LoadDocument("");
    off.ReportFromDGV(dgv, 1, 1);
    off.Visible(true);

    off.Disconnect();
    off = null;
    GC.Collect();
}

```

Рисунок 3.13 — Обработчик нажатия кнопки «Печать»

В обработчике нажатия кнопки вызывается класс «Office» с необходимыми параметрами: строкой подключения к БД и заполненной данными таблицей типа «DataGridView». В классе «Office» происходит обработка данных и вывод их в пакет «MS Excel».

3.3 Разработка интерфейса системы

Общение оператора с ИС происходит через главное меню (ГМ), которое реализует диалог с оператором. Реализация представлена на рисунках 3.14, 3.15.

```

public class DM
{
    ПЕРЕМЕННЫЕ

    ссылка 1
    public DM()...
    ссылка 1
    public bool CreateClassTables()...
    // Загрузка настроек
    ссылка 1
    public void LoadXML(string FN)....
    //Сохранение настроек
    ссылка 1
    public void SaveXML(string FN)....
    ссылок 11
    public string GetConnectionString(bool FLAliasMaster = true)....
    ссылка 1
    public string FormConnectionString(string str)....

    // Создать БД
    ссылка 1
    public bool CreateBD()...//CreateBD

    //проверяет наличие таблицы в БД по имени
    ссылок 14
    public bool FindTable(string TableName) ...
    //проверяет наличие процедуры в БД по имени
    ссылок 0
    public bool FindProc(string TriggerName)....
    //Получить ID корневой записи
    ссылок 0
    public int GetRootTree(string Tree_Object)....
    //Получить ID всех подгрупп группы ID
    ссылок 0
    public string GetStrIDPodGroup(string ID)....

    // получить строку со всеми полями таблицы кроме ID (Примем "Name, NameFull")
    ссылка 1
    public string GetFieldTable(string TableName)....

    //поиск в таблице по полю
    ссылок 0
    public int GetIDpoField(string TableName, string FieldName, string FindStr, bool flLIKE)....
    //поиск в таблице Class по Name
    ссылок 0
    public int GetIDpoClass(string ObjectStr, string FindStr, bool flLIKE)....
    ссылок 0
    public int AddTree(string Tree_Object, string Name)....

    //***** Вспомогательные таблицы ...
    ссылка 1
    public bool CreateTableTree()... //CreateTableTree
    // создаетиет таблицы TypeDocs
    ссылка 1
    public bool CreateTableTypeDocs()... //CreateTableTypeDocs
    // созданиет таблицы ClassName ...
    ссылка 1
    public bool CreateTableClassName()... //CreateTableClassName
    // созданиет таблицы Class
    ссылка 1
    public bool CreateTableClass()... //CreateTableClass
    ссылка 1
    public bool CreateTableClassArr()... //CreateTableClassArr

```

Рисунок 3.14 — Класс «DM» (часть 1)

```

// создание таблицы контактные лица
ссылка 1
public bool CreateTableContacts()... //CreateTableContacts
//ЗНАЧЕНИЯ СВОЙСТВ
ссылка 1
public bool CreateTablePropValue()... //CreateTablePropValue
// СВОЙСТВА
ссылка 1
public bool CreateTableProp()... //CreateTableProp

ссылка 1
public bool CreateTableSaveDocs()... //CreateTableSaveDocs
// Настройка видимости элементов шапки в зависимости от типа документа
ссылка 1
public bool CreateTableFieldsJornal()... //CreateTableFieldsJornal

// созданиет таблицы Users
ссылка 1
public bool CreateTableUsers()... //CreateTableUsers
// созданиет таблицы Peoples
ссылка 1
public bool CreateTablePeoples()... //CreateTablePeoples

// созданиет таблицы Clients
ссылка 1
public bool CreateTableClients()... //CreateTableClients
// созданиет таблицы Address
ссылка 1
public bool CreateTableAddress()... //CreateTableAddress
// созданиет таблицы Documents
ссылка 1
public bool CreateTableDocuments()... //CreateTableDocuments

// созданиет таблицы Jornal
ссылка 1
public bool CreateTableJornal()... //CreateTableJornal
// созданиет таблицы JornalTabl
ссылка 1
public bool CreateTableJornalTabl()... //CreateTableJornalTabl
}

```

Рисунок 3.15 — Класс «DM» часть 3

Модуль «Office.cs» включает в себя класс «Office» для работы с приложением фирмы Microsoft «MS Excel». Реализация представлена на рисунках 3.16,3.17.

```

3 public class Office
4 {
5     public string ConnectionString;
6     public Excel.Application ExcelApp;
7     private Excel.Workbooks ExcelEppWorkBooks; //все книги
8     private Excel.Workbook ExcelEppWorkBook; // одна книга
9     private Excel.Sheets ExcelSheets; // листы
10    private Excel.Worksheet ExcelWorkSheet; // Рабочий лист
11    private Excel.Range ExcelCells; // ячейки
12
13    private string SeparatorFloat;
14    private string[]
15    vsS = { "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L",
16            "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z",
17            "AA", "AB", "AC", "AD", "AE" };
18
19    //ссылка 2
20    public Office(){}
21    //Подключится к офису
22    //ссылка 2
23    public bool Connect(bool visible){}
24    // Закреть офис
25    //ссылка 2
26    public bool Disconnect(){}
27    //ссылка 2
28    public bool LoadDocument(string FileName){}
29    //ссылка 2
30    public void Visible(bool visible){}
31    //ссылка 0
32    private string toText(int row,int col){}
33    //ссылка 0
34    public static string ParseColNum(int colNum){}
35    //ссылка 2
36    private static string GetLetterByNum(int colNum){}
37    //Вывести текст в ячейку
38    //ссылка 0
39    public void putTextToCell (int col, int row, string Text){}
40    //ссылка 0
41    public void putStyleToCell(int col, int row, int ColorIndex, bool Bold, bool Italic){}
42    //ссылка 0
43    public void putTextToCell(string cellAddress, string Text){}
44    // получить значение ячейки
45    //ссылка 0
46    public string getValueFromCell(int col, int row){}
47    //Вывести формулу в ячейку
48    //ссылка 0
49    public void putFormulaToCell(int col, int row, string Formula, string NumberFormat){}
50    //ссылка 0
51    public void iOutputExcelF(int col, int row, int FormulaCol1, int FormulaRow1, int FormulaCol2, int FormulaRow2, string sValue){}

```

Рисунок 3.16 — Класс «Office» (часть 1)

```

// Выделить область
    ссылка 0
    public void SelectCells (int col, int row, int col1, int row1)...
    // Задать ширину столбца в мм
    ссылка 0
    public void putColumnCells(int col, int ColumnWidth)...
    // Задать высоту столбца в мм
    ссылка 0
    public void putRowCells ( int row, int RowHeight)...
    // Задать ориентацию текста
    ссылка 0
    public void setOrientationForActiveRange(int Orientation)...
    // Группировать
    ссылка 0
    public void setGroupForActiveRange(string Field)...
    // Объединить активную область
    ссылка 0
    public void setActiveMerge()...
    ссылка 0
    public void setAlignmentForActiveRange (int HorizontalAlignment, int VerticalAlignment)...
    ссылка 0
    public void setFontForActiveRange (Font font, Color color)...
    ссылка 0
    public void setCellColorForActiveRange(Color color)...
    //Переносить по словам
    ссылка 0
    public void setCellPerenosForActiveRange(bool fl)...
    // ПЕЧАТЬ СПИСКА DataGridView
    ссылка 1
    public void ReportFromDGV(DataGridView DGV, int BegRow, int BegCol)...
    // ПЕЧАТЬ Шапки FEdit
    ссылка 1
    public void ReportFromShapkaFEdit(DataGridView DGV, int BegRow, int BegCol)...

    // ПЕЧАТЬ Шапки И таблицы FEdit
    ссылка 1
    public void ReportFromAllFEdit(DataGridView DGV, DataGridView DGVTabl, int BegRow, int BegCol)...
    // ПЕЧАТЬ
    ссылка 0
    public void ReportFromRD(string SQL, int BegRow, int BegCol)...

    //НАЧ ***** ШАБЛОНЫ *****
    ссылка 2
    public void PRNShablonFindLevel(out int LevBeg, out int LevEnd)...
    // Вывести шапку
    ссылка 0
    public void PRNShablon(SqlCommand cmd)...
    //вывести табличную часть
    ссылка 0
    public void PRNShablonSpisok(SqlCommand cmdTabl)...

```

Рисунок 3.17 — Класс «Office» (часть 2)

3.4 Разработка запросов

Все операции с БД выполняются с помощью SQL-запросов, в том числе создание самой БД и ее таблиц. Выполняется проверка на существование БД и в случае ее отсутствия происходит создание БД. Реализация представлена на рисунке 3.18.

```

public bool CreateBD()
{
    SqlConnection Conn = new SqlConnection(GetConnectionString(false));
    Conn.Open();
    #region Проверить существование БД
    SqlDataReader RD; ;
    SqlCommand cmd;
    cmd = new SqlCommand();
    cmd.Connection = Conn;
    cmd.CommandTimeout = 600;
    cmd.CommandText = "select 1 from sys.databases" +
        " where name = '" + iniSet.Aliase + "'";
    RD = cmd.ExecuteReader();
    if (RD.Read())// БД существует
    {
        RD.Close();
        RD.Dispose();
        cmd.Dispose();
        return true;
    }
    cmd.Dispose();
    RD.Close();
    RD.Dispose();
    #endregion
    MessageBox.Show("База данных отсутствует. Производится создание БД.", "Внимание.", MessageBoxButtons.OK);
}

```

Получить путь для создания БД

#region Создать БД

```

cmd = new SqlCommand();
cmd.Connection = Conn;
cmd.CommandTimeout = 600;
cmd.CommandText = "CREATE DATABASE " + iniSet.Aliase +
    " ON " +
    " ( NAME = " + iniSet.Aliase + "_dat, " +
    " FILENAME = '" + PathBD + "_dat.mdf', " +
    " SIZE = 10MB, " +
    " MAXSIZE = 500MB, " +
    " FILEGROWTH = 5MB) " +
    " LOG ON " +
    " ( NAME = " + iniSet.Aliase + "_log, " +
    " FILENAME = '" + PathBD + "_log.ldf', " +
    " SIZE = 2MB, " +
    " MAXSIZE = 5MB, " +
    " FILEGROWTH = 3MB ) ";

```

Рисунок 3.18 — Создание БД

Создание таблиц, если они отсутствуют в БД, выполняется также с помощью запроса (рисунок 3.19).

```

// создаст таблицу Peoples
ссылка 1
public bool CreateTablePeoples()
{
    if (FindTable("Peoples")) return true;

    SqlCommand cmd = new SqlCommand();
    cmd.Connection = Conn;
    cmd.CommandTimeout = 600;
    cmd.CommandText = " CREATE TABLE Peoples ( " +
        " ID int IDENTITY(1,1) NOT NULL PRIMARY KEY " +
        " ,TreeId      int REFERENCES Tree(ID) " +
        " ,F           varchar(20)      COLLATE Cyrillic_General_CI_AS NULL " +
        " ,I           varchar(20)      COLLATE Cyrillic_General_CI_AS NULL " +
        " ,O           varchar(20)      COLLATE Cyrillic_General_CI_AS NULL " +
        " ,Sex         varchar(1)        COLLATE Cyrillic_General_CI_AS NULL " +
        " ,Name        varchar(100)      COLLATE Cyrillic_General_CI_AS NULL " +
        " ,FullName    varchar(254)      COLLATE Cyrillic_General_CI_AS NULL " +
        " ,INN         varchar(15)       COLLATE Cyrillic_General_CI_AS NULL " +
        " ,DR          datetime " +
        " ,Telephon    varchar(100)      COLLATE Cyrillic_General_CI_AS NULL " +
        " ,AddressFact varchar(100)      COLLATE Cyrillic_General_CI_AS NULL " +
        " ,AddressUr   varchar(100)      COLLATE Cyrillic_General_CI_AS NULL " +
        ") ";

    try
    {
        cmd.ExecuteNonQuery();
    }
    catch (Exception e2)
    {
        MessageBox.Show("Ошибка при создании таблицы Peoples ! " + e2.Message);
        return false;
    }

    cmd.Dispose();
    return true;
} //CreateTablePeoples

```

Рисунок 3.19 — Создание таблицы «Peoples»

Другие таблицы создаются аналогично.

Выборка данных из таблиц также осуществляется с помощью запросов.

Сценарий диалога ИС представлен на рисунке 3.20.

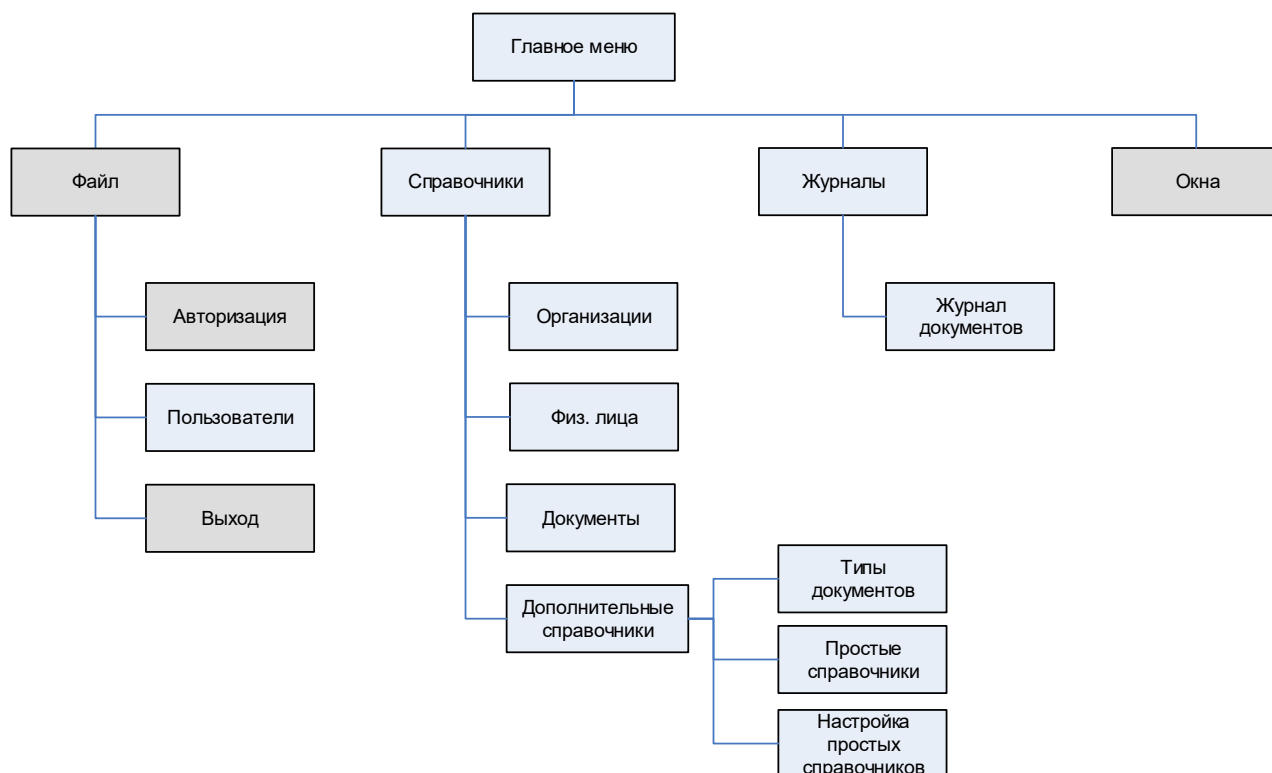


Рисунок 3.20 — Сценарий диалога

3.5 Руководство по установке и настройке программы

ИС хранит данные в СУБД «MS SQL Server». При первом запуске ИС выполняется создание БД. Информация для подключения к СУБД записывается в файл «DOCS.xml». На рисунках 3.21-3.23 показаны окна для настройки подключения к СУБД.

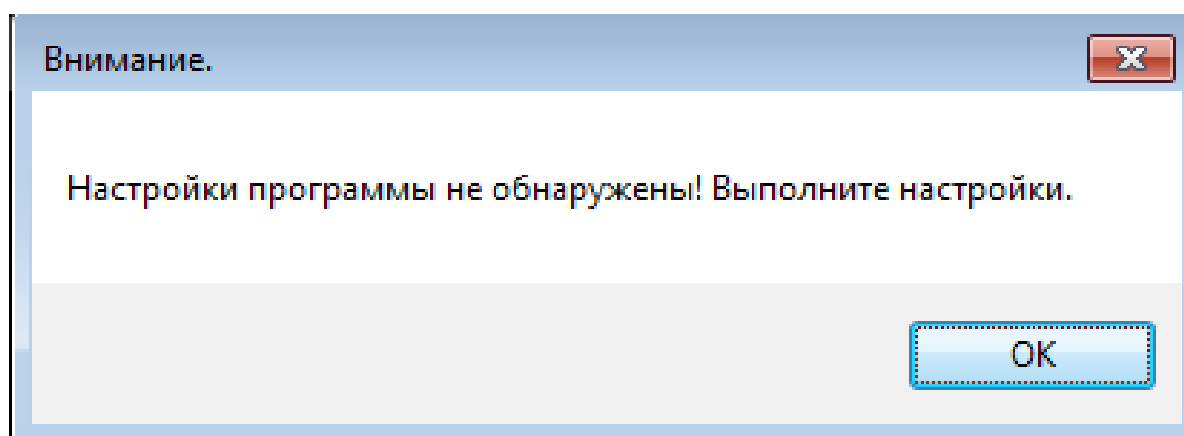


Рисунок 3.21 — Предупреждение о необходимости настройки системы

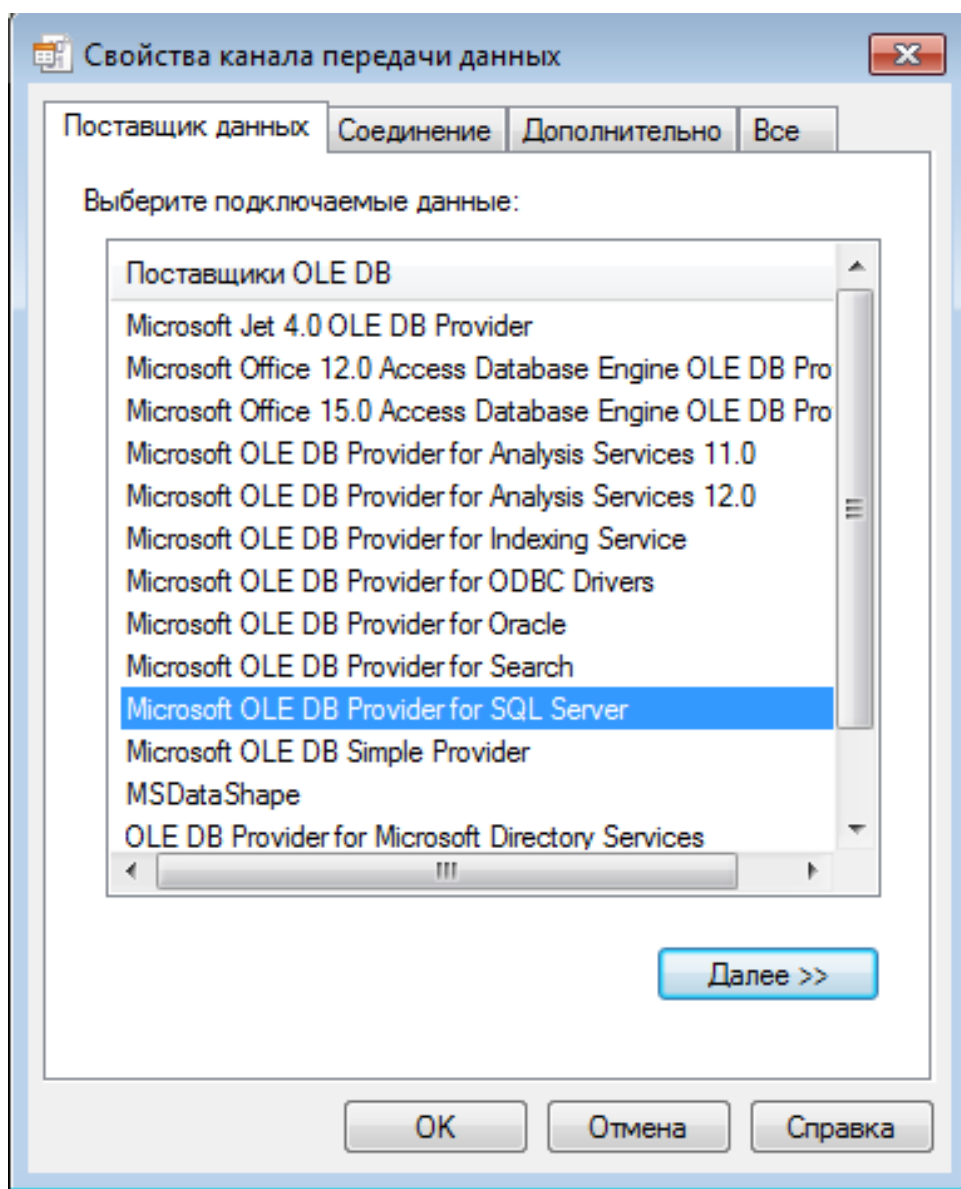


Рисунок 3.22 — Диалог «Свойства канала передачи данных»
Поставщиком данных для ИС является «MS SQL Server».

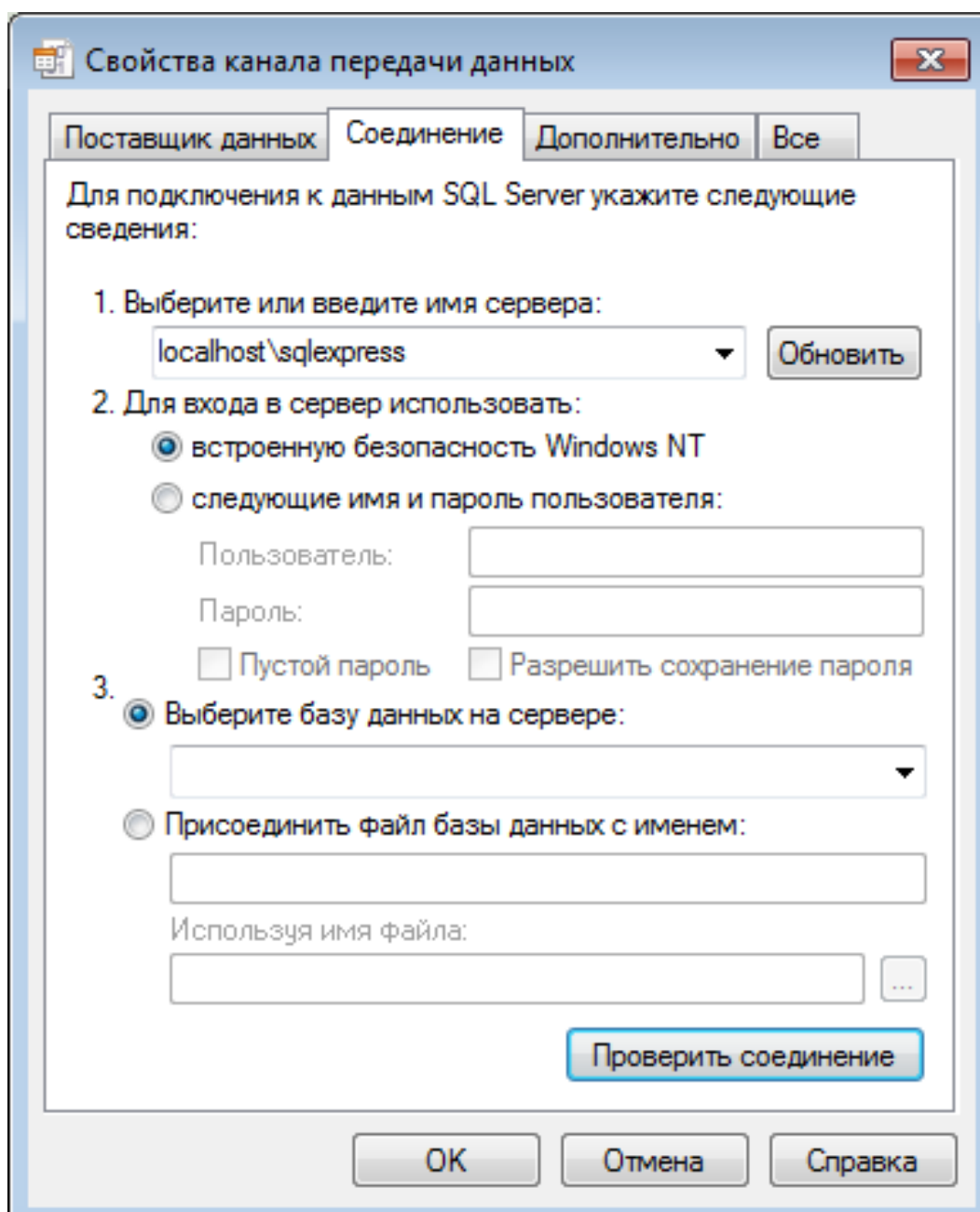


Рисунок 3.23 — Диалог «Свойства канала передачи данных»

У пользователя ИС должны быть «административные» права для СУБД и ОС. Заполнив настройки требуется выполнить проверку подключения.

3.6 Руководство пользователя

Выполнение ИС начинается с запуска «DOCS.exe». На экран выводится окно для авторизации оператора. Настроечные данные авторизации: «логин» «1», пароль «1» (рисунок 3.24).

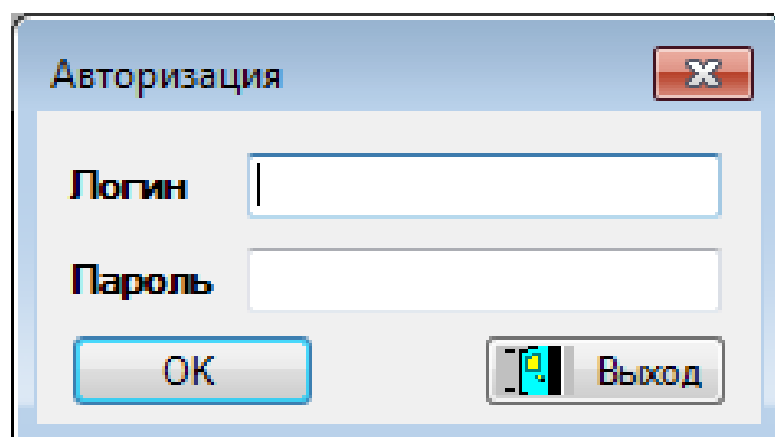


Рисунок 3.24— Окно авторизации

После проверки доступа открывается главное окно ИС.

Главное меню (ГМ) ИС позволяет пользователю использовать все функции ИС и включает в себя пункты: «Файл», «Справочники», «Журналы» и «Окна».

Одним из самых востребованных справочников ИС является «Организации». Открыть окно просмотра и изменения справочника можно через пункт ГМ «Справочники / Организации» (рисунок 3.25, 3.26).

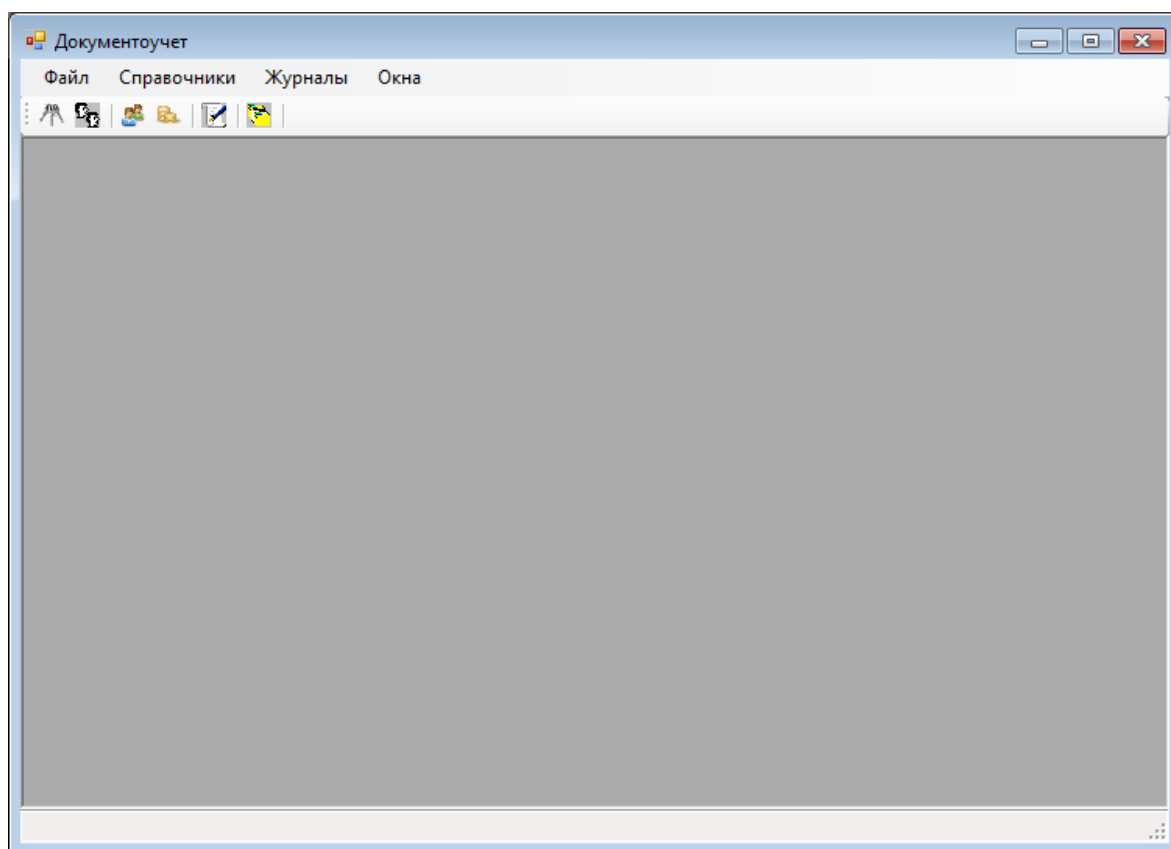


Рисунок 3.25 — Главное окно ИС

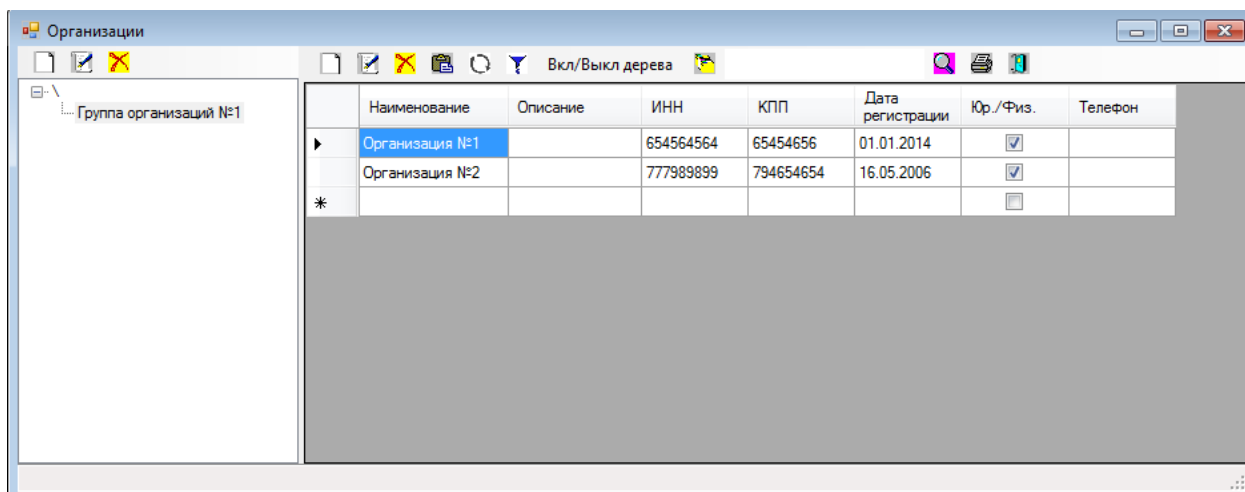


Рисунок 3.26 — Окно организаций

Данная форма универсальная для просмотра всех таблиц ИС. Опишем ее командные элементы, кнопки:

- «Новый» применяется при вводе нового элемента. Редактирование может производиться как в самом списке, так и в отдельном окне, если это предусмотрено для данного справочника,

- «Изменить» — это изменение элемента,

- «Удалить» применяется при удалении элементов,

- «Копировать» позволяет копировать текущий элемент,

- «Обновить» применяется для обновления данных путем запроса у СУБД,

- «Фильтр» — это фильтрация списка элементов по заданным пользователем параметрам,

- «Включение/Выключение дерева» отображение элементов согласно иерархии справочника,

- «Подчиненные таблицы» отображает подчиненные справочники для текущего элемента,

- «Поиск» применяется при необходимости найти элемент по определенной строке,

- «Печать» экспортирует текущий список элементов в приложение «MS Excel»,

- Сортировка элементов списка выполняется двойным «щелчком»

манипулятора «мышь» по заголовку требуемого столбца,

- «Выход» закрывает текущее окно.

Выполнить настройку «простейших» справочников можно выбрав: «Справочники / Дополнительные справочники / Настройки простых справочников» (рисунки 3.27, 3.28).

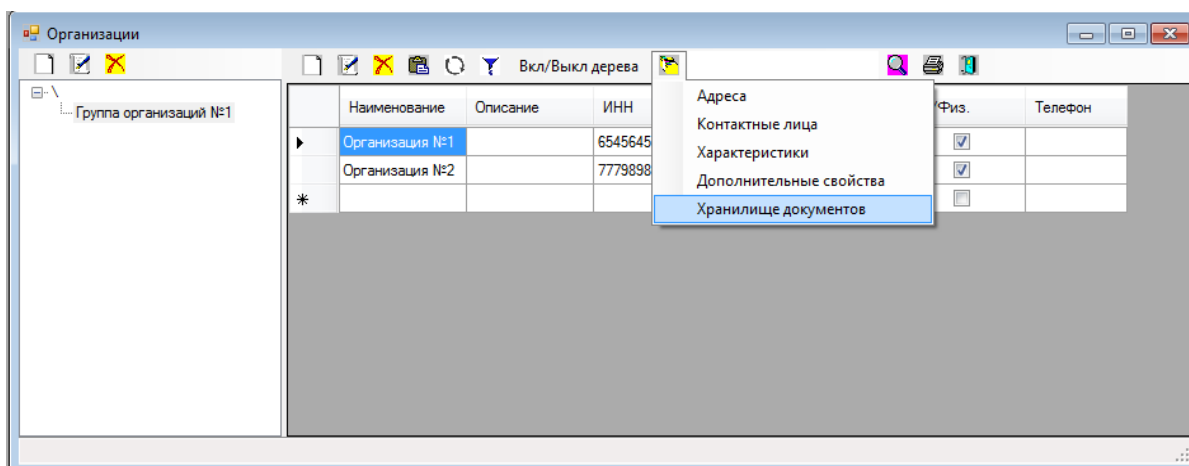


Рисунок 3.27 — Окно просмотра подчиненных справочников

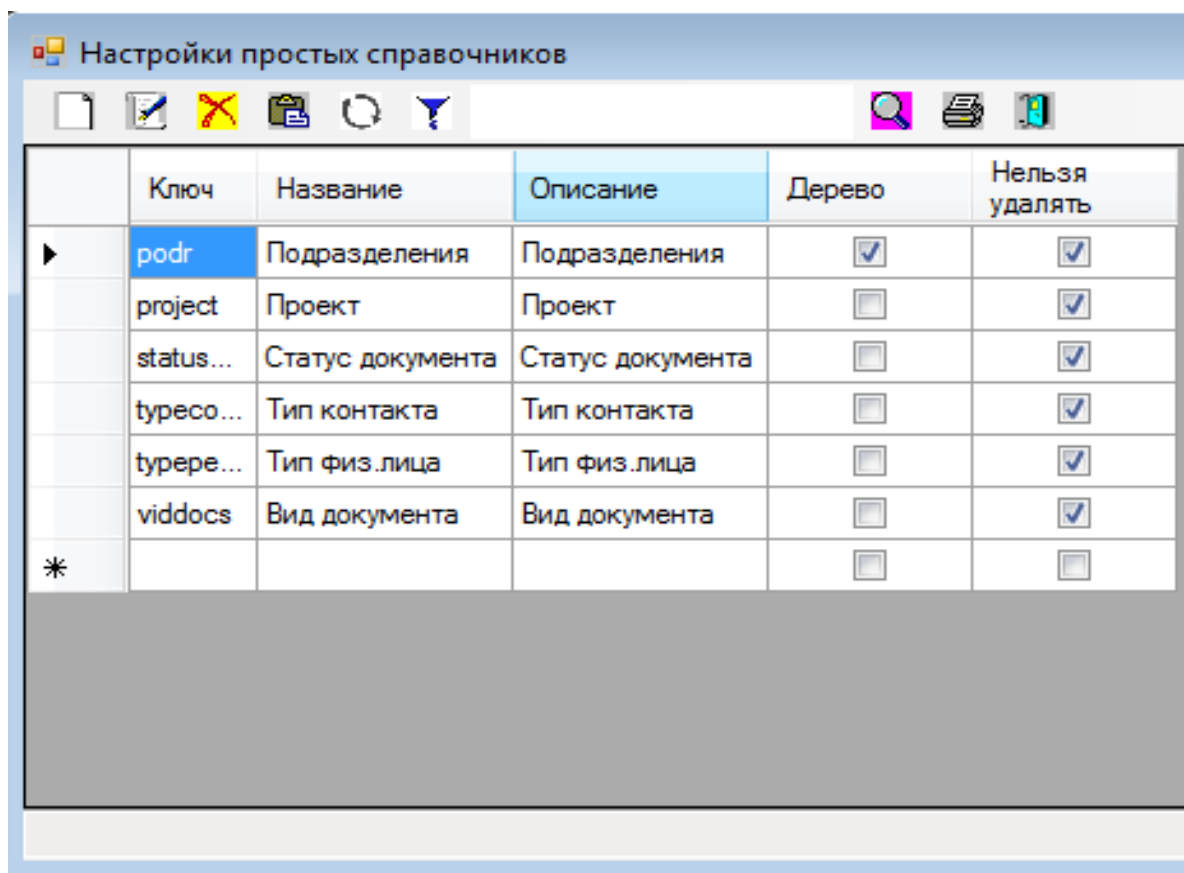


Рисунок 3.28 — Окно настройки «простейших» справочников

Элемент контекстного меню «Подчиненные таблицы» «Характеристики» позволяет открыть окно с характеристиками текущего объекта (рисунок 3.29).

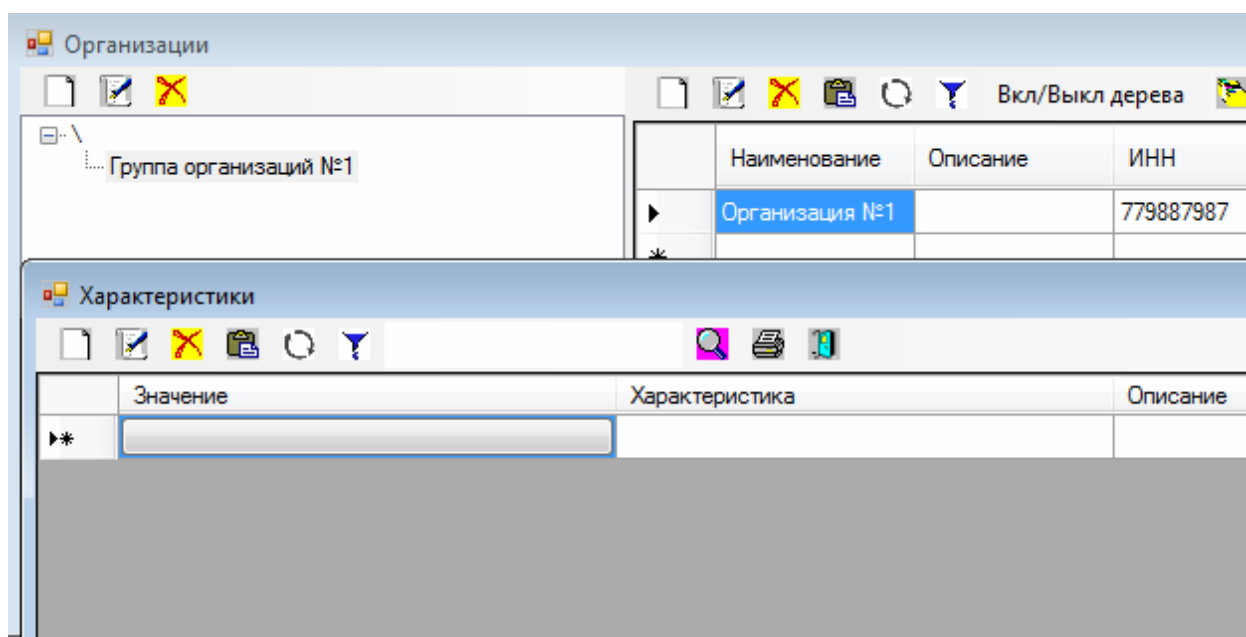


Рисунок 3.29 — Характеристики текущего объекта

В столбце «Значение» можно выбрать элемент из любого «простейшего» справочника (рисунки 3.30,3.31).

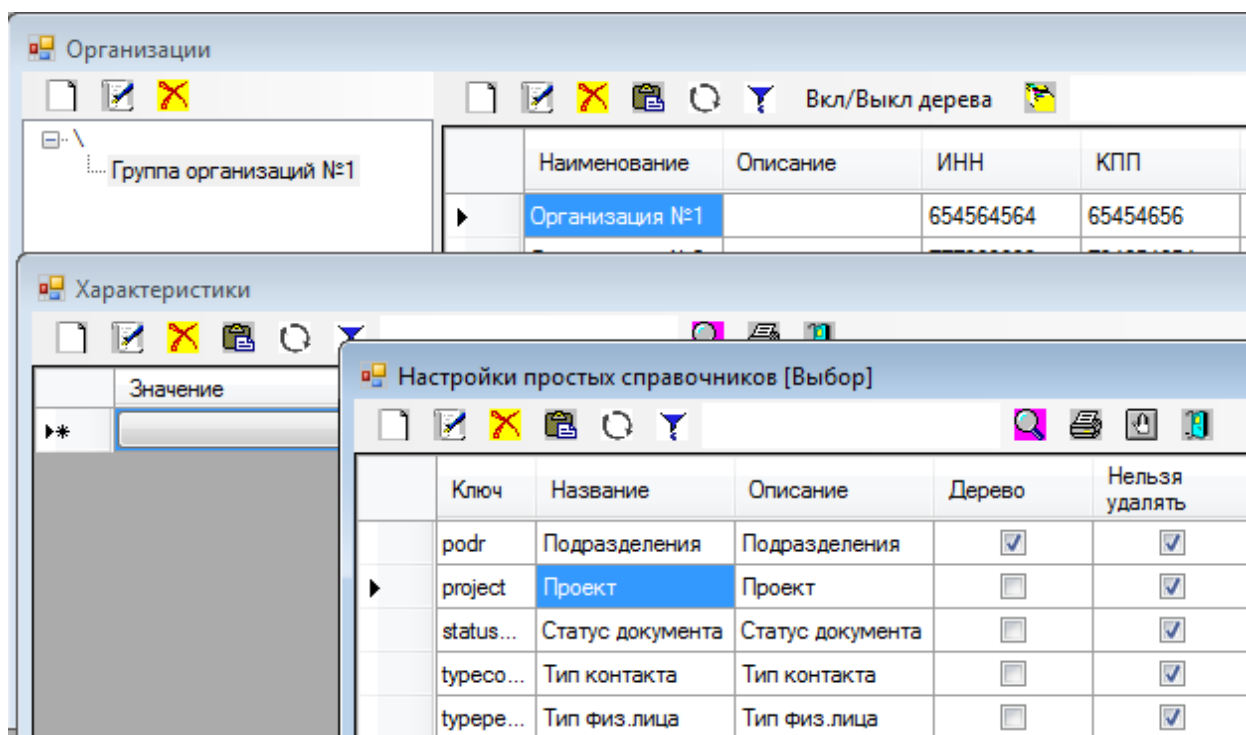


Рисунок 3.30 — Окно выбора «простейшего» справочника

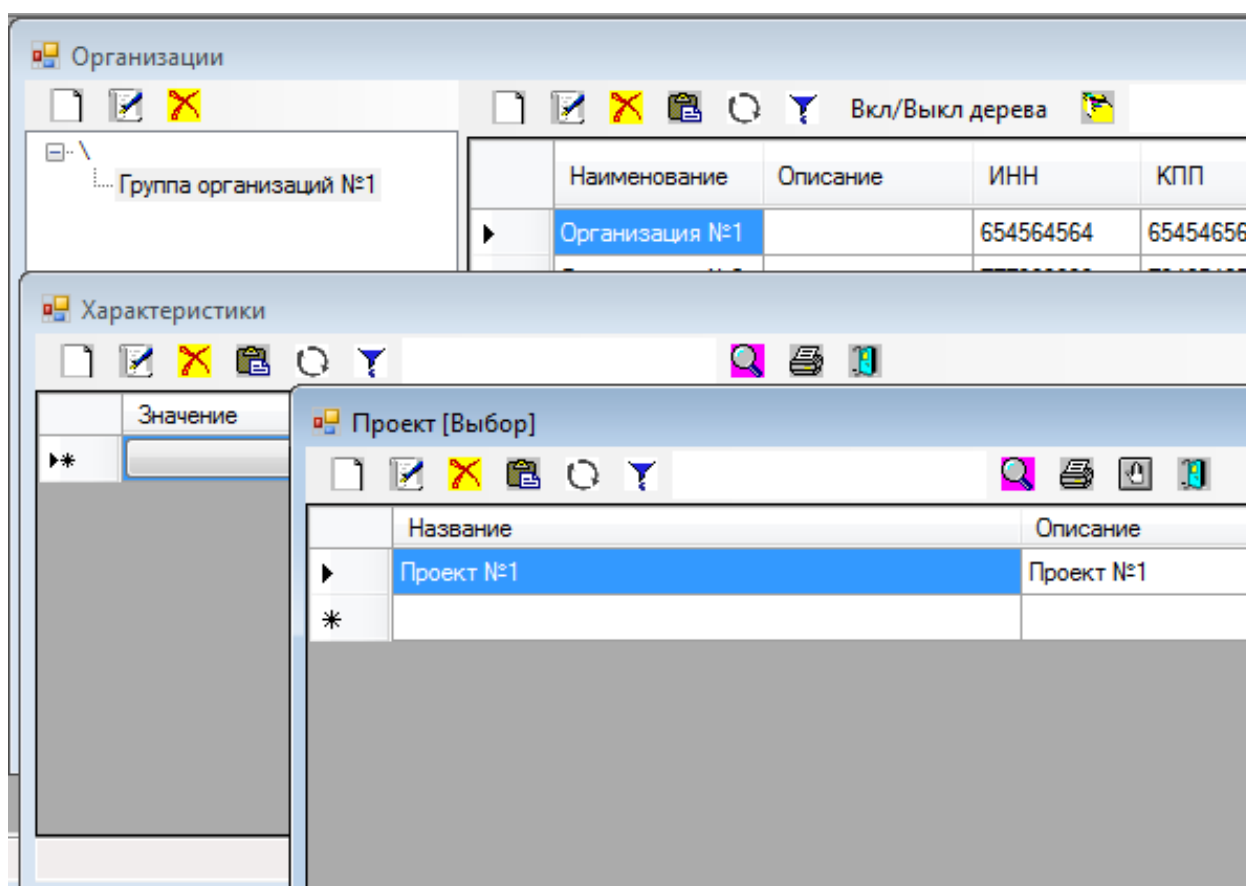


Рисунок 3.31 — Окно подбора значения «простейшего» справочника

В результате выполненных действий у элемента справочника будет новая характеристика.

Для «основных» таблиц ИС можно вести описание «дополнительных» свойств. За это отвечает пункт «контекстного» меню «Подчиненные таблицы» «Дополнительные свойства» (рисунки 3.32-3.35).

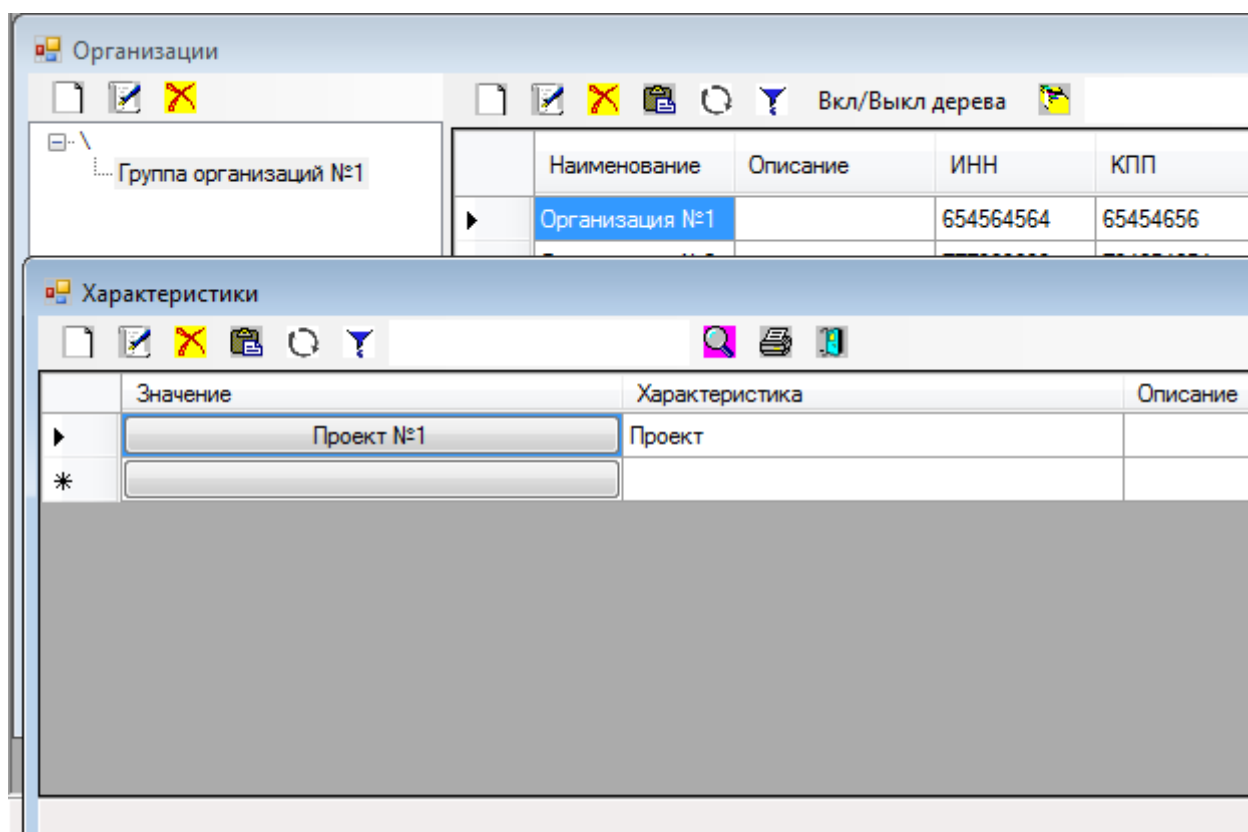


Рисунок 3.32 — Новая характеристика

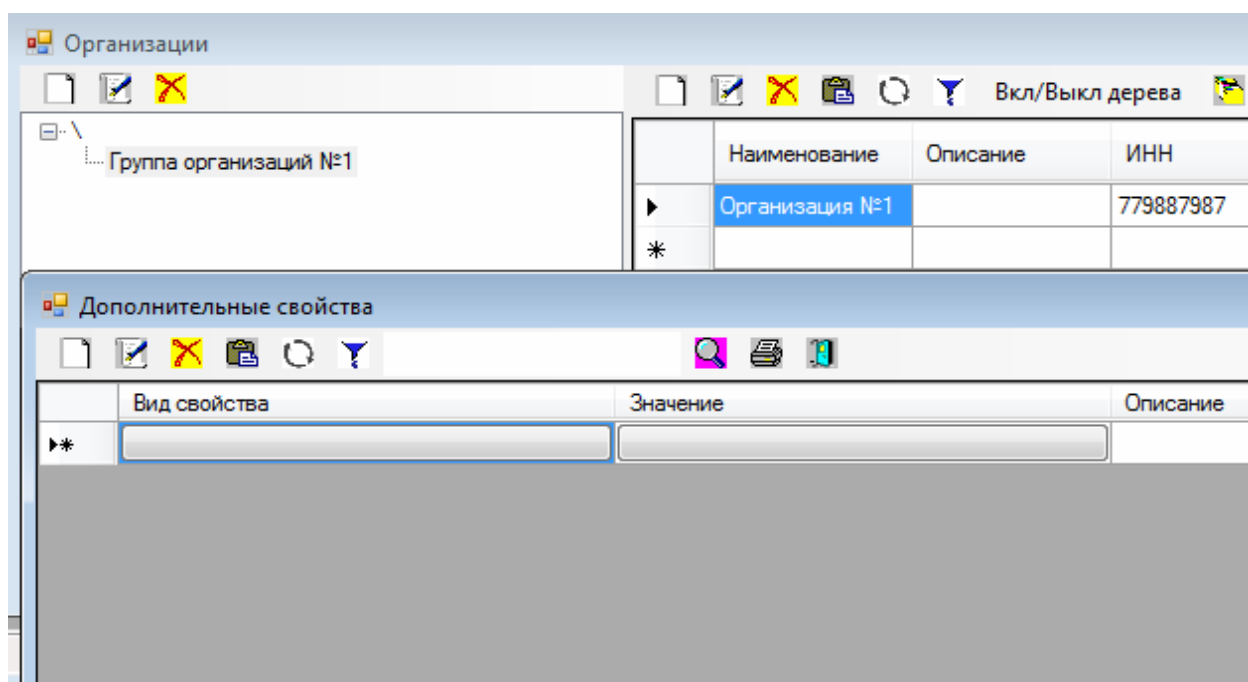


Рисунок 3.33 — «Дополнительные» свойства

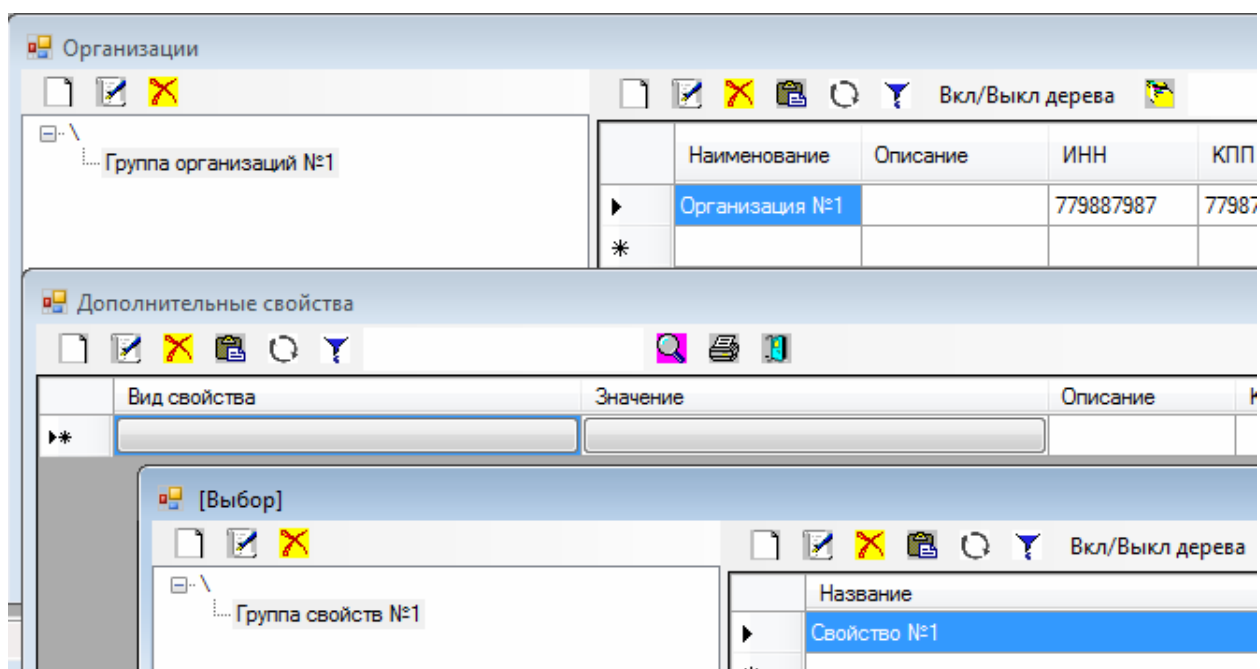


Рисунок 3.34 — Окно подбора вида свойств

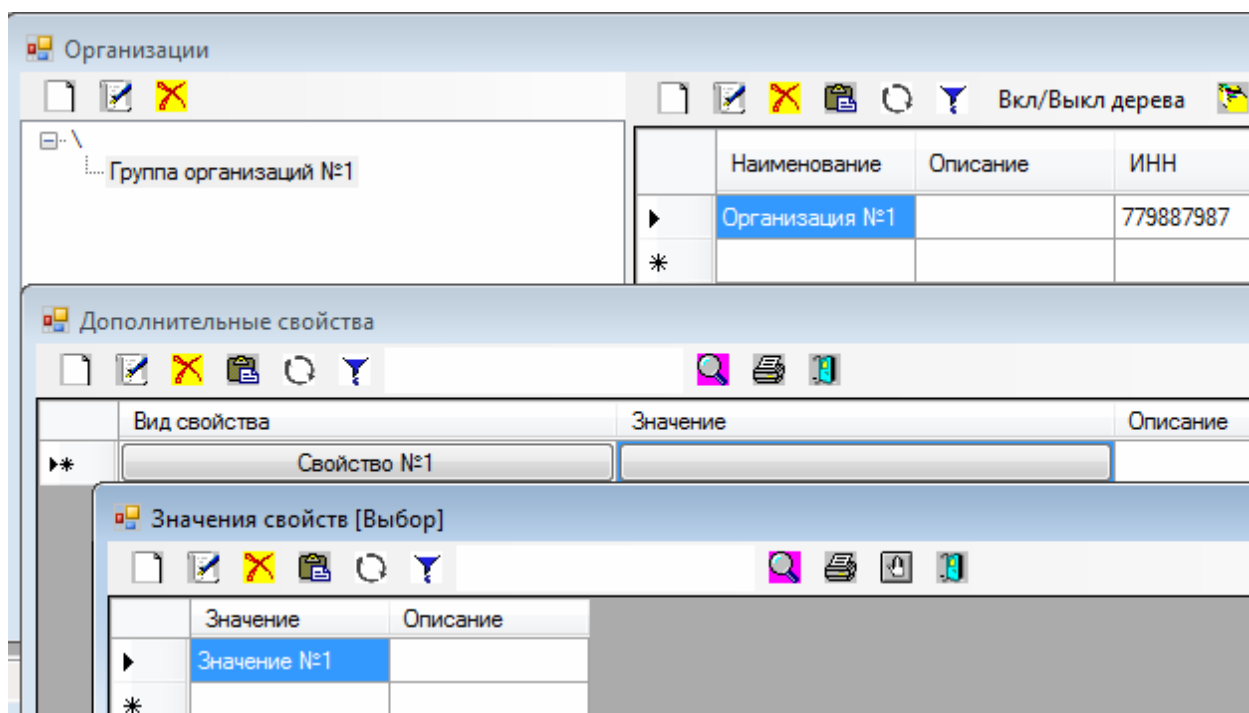


Рисунок 3.35 — Окно подбора значения свойств

В результате выполненных действий элементу «основного» справочника будет назначено «дополнительное» свойство (рисунок 3.37).

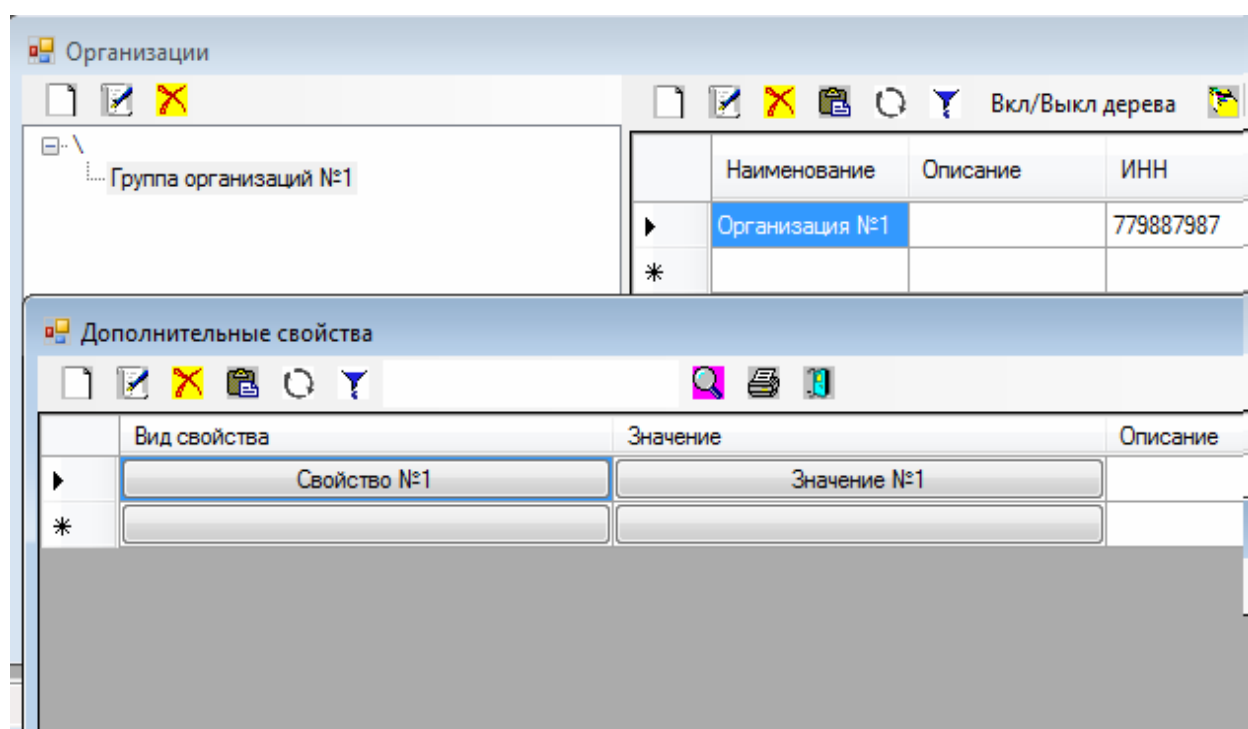


Рисунок 3.36 — Результат

ЗАКЛЮЧЕНИЕ

В результате работы была спроектирована и реализована ИС учета документов, решающая широкий спектр задач внесения и учета документов различного типа и с различными реквизитами.

Во время выполнения задачи были освоены основные понятия организации СЭД, проведен анализ существующих бизнес-процессов изучаемого предприятия. Доказана необходимость автоматизации и необходимость применения ИТ для внедрения СЭД. Был обоснован и разработан свой способ решения задачи.

Для проектируемой ИС были разработаны информационная модель и модель «сущность-связь», а также построена «ER - диаграмма».

Проанализированы существующие системы разработки приложений и СУБД. В результате анализа был произведен выбор из них.

В качестве системы разработки была выбрана «Visual Studio C#». Этот выбор обоснован гибкостью технологической платформы, являющейся основой системы, большим набором возможностей, предоставляемых ею для решения самого широкого круга задач.

В качестве СУБД использована «MS SQL Server».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ПНИЭИ [Электронный ресурс] – URL: <http://pniei.ru/> Дата обращения: 10.12.2020
2. Пауэрс Л. «Microsoft Visual Studio» — СПб.: БХВ-Петербург, 2019
3. ДИРЭКТУМ [Электронный ресурс] – URL: [https://www.tadviser.ru/index.php/Продукт:Directum_\(СЭД/ЕСМ-система\)](https://www.tadviser.ru/index.php/Продукт:Directum_(СЭД/ЕСМ-система)). Дата обращения: 30.12.2020
4. Docsvision [Электронный ресурс] – URL: <https://docsvision.com/> Дата обращения: 15.01.2021
5. Globus Professional [Электронный ресурс] – URL: <http://www.ecmonline.ru/software/globus/> Дата обращения: 16.01.2021
6. PayDox [Электронный ресурс] – URL: <http://www.paydox.ru/> Дата обращения: 17.01.2021
7. 1С: Документооборот [Электронный ресурс] – URL: <https://v8.1c.ru/doc8/vozmozhnosti-1s-dokumentoorobota/> Дата обращения: 11.02.2021
8. БОСС-Референт [Электронный ресурс] – URL: http://www.docva.ru/docflow/sys_review/7.php Дата обращения: 12.02.2021
9. ДЕЛО [Электронный ресурс] – URL: https://eos.ru/eos_products/eos_delo/ Дата обращения: 13.02.2021
10. ЕВФРАТ [Электронный ресурс] – URL: <https://evfrat.ru/sed-e1-evfrat/obzor-sistemy/> Дата обращения: 14.02.2021
11. МОТИВ [Электронный ресурс] – URL: <https://www.embarcadero.com/ru/products/cbuilder> Дата обращения: 15.02.2021
12. Подбельский В.В., Фомин С.С. Программирование на языке Си: Учеб. Пособие. – 2-е доп. Изд. – М: Финансы и статистика, 2017. – 600с.
13. IDE «C++ Builder» [Электронный ресурс] – URL: <https://www.embarcadero.com/ru/products/cbuilder> Дата обращения: 14.03.2021

14. Eclipse [Электронный ресурс] – URL: <https://javarush.ru/groups/posts/2359-obzor-eclipse-java-sreda-razrabotki-pod-sebja> Дата обращения: 14.03.2021
15. Зиборов В.В. «Visual C# на примерах» СПб.: BHV - БХВ-Петербург, 2017
16. Хортон Айвор «Visual C++: базовый курс» ООО «И.Д. Вильямс»
17. Давыдов В. Г. «Visual C++. Разработка Windows-приложений с помощью MFC и API-функций.» — СПб.: БХВ-Петербург, 2018
18. Ч. Петзолд Программирование для Windows; в двух томах - СПб.: BHV - Санкт-Петербург, 2019
19. Шумаков В.П. Delphi и создание приложений баз данных. М., Нолидж, 2018
20. Ч. Петзолд Программирование для Windows; в двух томах - СПб.: BHV - Санкт-Петербург, 2019
21. Культин Н. Б. «Microsoft Visual C# в задачах и примерах.» — СПб.: БХВ-Петербург, 2018. — 320 с.

ПРИЛОЖЕНИЕ А

Листинг программы

Файл FTreeGrid.cs

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace GRID
{
    public partial class FTreeGrid : Form
    {
        #region ***** ПЕРЕМЕННЫЕ *****
        *****
        public bool FIRedOnly = false; // Разрешено редактирование в форме или нет
        public Form FormEdit; // форма для редактирования
        // public ClassTable CTEditTabl; // табличная часть

        public ClassTable CT; // ОСНОВНАЯ ТАБЛИЦА
        public ClassTable CTEdit; //для редактирования в диалоге (ContextMenu)
        public ClassTable CTGroup; // таблица для вызова окна "ГРУППЫ" ПКМ
        public DM dm;

        public int ID=-1; //текущий ID
```

```

public string StrSelect = "";      // строка выбранной записи
public int IndBeg = 1, IndEnd = 1; // активные радиокнопки для НАСТРОЙКИ
ПЕРИОДА

```

```

private SqlConnection Conn;
private SqlCommand cmd;
private SqlDataAdapter da;
// private DataTable dt;
#endregion

```

```

public FTreeGrid()
{
    InitializeComponent();
}

```

```

// ***** ИНИЦИАЛИЗАЦИЯ формы
*****

```

```

private void FTreeGrid_Load(object sender, EventArgs e)
{
    FlReadOnly = CT.FlReadOnly;

```

```

    #region Форма вызвана для выбора группы
    toolStripMenuItemTreeVib.Visible = CT.FlSelectTree;
    if (CT.FlSelectTree)
    {
        splitContainer1.Panel2Collapsed = true;
    }
    #endregion

```

```

    #region Контекстное меню для изменения вершины
    if (CTGroup == null)
        группыToolStripMenuItem.Visible = false;
    else
        группыToolStripMenuItem.Visible = true;
    #endregion

```

```

#region ReadOnly
dgv.AllowUserToAddRows    = !CT.NotAllowUserToAddRows;
dgv.AllowUserToDeleteRows = !CT.NotAllowUserToDeleteRows;
if (FIReadOnly)
{
    dgv.ReadOnly = true;
    toolStripMenuItemDGVAdd.Enabled = false;
    toolStripMenuItemDGVEdit.Enabled = false;
    toolStripMenuItemDGVDelete.Enabled = false;
    toolStripMenuItemCopy.Enabled = false;

    toolStripMenuItemTreeAdd.Enabled = false;
    toolStripMenuItemTreeEdit.Enabled = false;
    toolStripMenuItemTreeDel.Enabled = false;
}
else
{
    dgv.ReadOnly = false;
    toolStripMenuItemDGVAdd.Enabled = true;
    toolStripMenuItemDGVEdit.Enabled = true;
    toolStripMenuItemDGVDelete.Enabled = true;
    toolStripMenuItemCopy.Enabled = true;

    toolStripMenuItemTreeAdd.Enabled = true;
    toolStripMenuItemTreeEdit.Enabled = true;
    toolStripMenuItemTreeDel.Enabled = true;
}
#endregion

if (FormEdit != null) // Если редактирование происходит в другой форме
{
    dgv.ReadOnly = true;
}

```

```

CT.ConnectionString = dm.GetConnectionString();

Conn = new SqlConnection();
Conn.ConnectionString = dm.GetConnectionString();
Conn.Open();

#region Дерево для таблицы CLASS
if (CT.TableName.ToUpper() == "CLASS" && CT.ClassObject != String.Empty)
// таблица CLASS и есть настройки для дерева
{
    SqlCommand cmdEdit = new SqlCommand();
    SqlDataReader RD;
    cmdEdit.Connection = Conn;
    cmdEdit.CommandTimeout = 600;
    cmdEdit.CommandText = " select tree from ClassName where Object = '" +
CT.ClassObject + "'";

    RD = cmdEdit.ExecuteReader();

    while (RD.Read())
    {
        string strTree = RD["tree"].ToString();
        if (strTree != String.Empty)
            CT.FlTree = !Convert.ToBoolean(strTree);
        else
            CT.FlTree = true;
        break;
    }
    RD.Close();
    RD.Dispose();
    cmdEdit.Dispose();
}

#endregion

```

```

CT.BeginSQL(); //начальный запрос

this.Text = CT.TextForm;          // Заголовок окна
dgv.Tag = CT.TextForm;

IndBeg = 1; IndEnd = 1;

toolStripMenuItemVibor.Visible = CT.FlSelect;
if (CT.FlSelect) this.Text = this.Text + " [Выбор]"; // Заголовок окна для выбора

int tempLocalID = CT.LocateID;
string tempLocalColumn = CT.LocateColumn;

bool rez = SQLSelect();          // Выполнить основной запрос для таблицы

#region Создание столбцов ГРИДА
dgv.AutoGenerateColumns = false;

DataGridViewTextBoxColumn dgvCol; // Текстовый столбец
// без кнопки-столбца
//      DataGridViewTextBoxColumn dgvBut; // Столбец - кнопка
DataGridViewButtonColumn dgvBut; // Столбец - кнопка

DataGridViewCheckBoxColumn dgvCh;

for (int i = 0; i < CT.ArrFields.Count; i++)
{
    ClassField temp = (ClassField)CT.ArrFields[i];
    if (temp.Butt == 2)
    {
        dgvCh = new DataGridViewCheckBoxColumn();

        dgvCh.HeaderText = temp.NameRus;
        dgvCh.Width = temp.Width;
    }
}

```

```

        if (temp.flUpd)
            dgvCh.Tag = "UPD:1";
        else
            dgvCh.Tag = "UPD:0";
        dgvCh.Name = temp.Name;
        dgvCh.DataPropertyName = temp.Name;
        dgvCh.ReadOnly = temp.ReadOnly;

        dgvCh.ToolTipText = temp.ToolTipText;

        dgv.Columns.Add(dgvCh);

    }
    else
    {
        if (temp.Butt == 1)
        {
            // без кнопки-столбца
            //dgvBut = new DataGridViewTextBoxColumn();
            dgvBut = new DataGridViewButtonColumn();

            dgvBut.HeaderText = temp.NameRus;
            dgvBut.Width = temp.Width;
            if (temp.flUpd)
                dgvBut.Tag = "UPD:1";
            else
                dgvBut.Tag = "UPD:0";
            dgvBut.Name = temp.Name;
            dgvBut.DataPropertyName = temp.Name;
            dgvBut.ReadOnly = temp.ReadOnly;

            dgvBut.ToolTipText = temp.ToolTipText;

            dgv.Columns.Add(dgvBut);
        }
    }

```

```

else
{
    dgvCol = new DataGridViewTextBoxColumn();
    dgvCol.HeaderText = temp.NameRus;
    dgvCol.Width = temp.Width;
    if (temp.flUpd)
        dgvCol.Tag = "UPD:1";
    else
        dgvCol.Tag = "UPD:0";
    dgvCol.Name = temp.Name;
    dgvCol.DataPropertyName = temp.Name;
    dgvCol.ReadOnly = temp.ReadOnly;

    dgvCol.ToolTipText = temp.ToolTipText;

    dgv.Columns.Add(dgvCol);

    dgv.Columns[temp.Name].Visible = temp.Visible;
}
}
}

```

#endregion

```

CT.LocateID = tempLocalID;
CT.LocateColumn = tempLocalColumn;
rez = SQLSelect();          // Выполнить основной запрос для таблицы
if (!rez) this.Close();

```

#region

ПОДЧИНЕННЫЕ

СПРАВОЧНИКИ *****

```

// получить список подчиненных справочников
bool flTables = false;
for (int i = 0; i < CT.ArrTables.Count; i++)
{
    ClassTable tCT = (ClassTable)CT.ArrTables[i];

```

```

//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
if ((CT.FlReadOnly) && (tCT.TableName.ToUpper() == "QUESTS"))
//тестируемый не может видеть вопросы и ответы
    continue;
tCT.FlReadOnly = CT.FlReadOnly;

ToolStripMenuItem tsi = new ToolStripMenuItem(); // создать выпадающий
ЭЛЕМЕНТ МЕНЮ

tsi.Text = tCT.TextForm;
tsi.Name = tCT.TableName;
tsi.Tag = i;
tsi.Click += new System.EventHandler(this.TablesToolStripMenuItem_Click);

toolStripMenuItemTables.DropDownItems.Add(tsi);

flTables = true;
}
toolStripMenuItemTables.Visible = flTables;

#endregion

#region ***** ДЕРЕВО
*****
splitContainer1.Panel1Collapsed = CT.FlTree;
toolStripMenuItemTree.Visible = !CT.FlTree;
if (!CT.FlTree)
{
    TreeInit();
    //----- ПОИСК УЗЛА В котором находится выбранный элемент
(LOCALID)
    string IDN = "";
    if (dgv.CurrentRow != null)
    {

```

```

        IDN = dgv.CurrentRow.Cells["TreeID"].Value.ToString();
        CT.LocateID = tempLocalID;
        treeView.SelectedNode = TreeFindNode(IDN);
    }
}
#endregion

#region ЖУПНАЛ ДОКУМЕНТОВ
if (CT.FindField("DATEDOC") != null) //есть поле DateDoc
{
    toolStripMenuItemSetupPeriod.Visible = true;
    toolStripStatusLabelPeriod.Visible = true;
    toolStripStatusLabelPeriod.Text = "Установлен период: ... - ...";

    toolStripMenuItemCopy.ToolTipText = "Ввести документ на основании";
}
else
{
    toolStripMenuItemSetupPeriod.Visible = false;
    toolStripStatusLabelPeriod.Visible = false;

    toolStripMenuItemCopy.ToolTipText = "Копировать запись";
}
#endregion

/*    if (Conn.State == ConnectionState.Open)
        Conn.Close();
    Conn.Dispose();*/
    GC.Collect();
}

private void FTreeGrid_FormClosing(object sender, FormClosingEventArgs e)
{
    if (Conn.State == ConnectionState.Open)
        Conn.Close();
}

```

```

        Conn.Dispose();
    }

    // Главный Select
    public bool SQLSelect()
    {
        bool Rez = true;

        /*      Conn = new SqlConnection();
        Conn.ConnectionString = dm.GetConnectionString();
        Conn.Open();*/

        cmd = new SqlCommand();
        cmd.Connection = Conn;
        cmd.CommandTimeout = 600;
        cmd.CommandText = CT.SQLText;

        #region ----- WHERE -----
        bool flWere = false;
        string strWere = "";

        for (int i = 0; i < CT.ArrInsertName.Count; i++) // подчиненная таблица с
накладываемыми условиями по ключу
        {
            if (strWere != String.Empty) strWere = strWere + " AND ";
            if ((System.Type)CT.ArrInsertType[i] == Type.GetType("System.String"))
                strWere = strWere + " " + CT.ArrInsertName[i].ToString() + " = " +
CT.ArrInsertValue[i].ToString() + "";
            else
                strWere = strWere + " " + CT.ArrInsertName[i].ToString() + " = " +
CT.ArrInsertValue[i].ToString();

            flWere = true;
        }

```

```

if (CT.StrWhereGlobal != String.Empty) // глобальные условия
{
    if (flWere)
        strWere = strWere + " AND " + CT.StrWhereGlobal;
    else
        strWere = CT.StrWhereGlobal;
    flWere = true;
}
if (!CT.MinusTree) // настройка вывода списка
{
    if (CT.StrWhereTree != String.Empty) // условия вхождения в дерево
    {
        if (flWere)
            strWere = strWere + " AND " + CT.StrWhereTree;
        else
            strWere = CT.StrWhereTree;
        flWere = true;
    }
}

if (CT.StrWhereLocal != String.Empty) // локальные условия
{
    if (flWere)
        strWere = strWere + " AND " + CT.StrWhereLocal;
    else
        strWere = CT.StrWhereLocal;
    flWere = true;
}

if (CT.StrWherePeriod != String.Empty) // условия по периоду (DateDoc)
{
    if (flWere)
        strWere = strWere + " AND " + CT.StrWherePeriod;
    else
        strWere = CT.StrWherePeriod;
}

```

```

        flWere = true;
    }

    if (flWere)
        cmd.CommandText = cmd.CommandText + " where " + strWere;

#endregion

da = new SqlDataAdapter();
da.SelectCommand = cmd;
try
{
    dataSet.Clear(); //очистка
    da.Fill(dataSet);
//    dgv.DataSource = dataSet.Tables[0];

    bindingSource.DataSource = dataSet.Tables[0];
    dgv.DataSource = bindingSource;

#region Позиционирование по ID
    if (CT.LocateID != -1)
    {
        //            DataRow[] rows = dTable.Select(String.Format("ID like '{0}%",
LocateID.ToString()));
        /* DataRow[] rows = dTable.Select("ID = "+LocateID.ToString());
        int idx = dTable.Rows.IndexOf(rows[0]);
        this.bindingSource.Position      =      this.bindingSource.Find("ID",
rows[0]["ID"].ToString());*/
        this.bindingSource.Position = this.bindingSource.Find(CT.PrimaryKeyName,
CT.LocateID.ToString());

        if (CT.LocateColumn != String.Empty)
            dgv.CurrentCell                                     =
dgv.Rows[dgv.CurrentCell.RowIndex].Cells[CT.LocateColumn];
    }

```

```

CT.LocateID = -1; CT.LocateColumn = "";
#endregion

#region Поиск по строке
if (CT.FindStr != String.Empty && CT.FindColumn != String.Empty)
{
    //          this.bindingSource.Position = this.bindingSource.Find(CT.FindColumn,
CT.FindStr.ToString());

    string type = dgv.Columns[CT.FindColumn].ValueType.ToString();    // тип
значения ячейки

    string whereFIND = "";

    ClassField temp = CT.FindField(CT.FindColumn);    //Описание текущего
поля

    if (type.Contains("System.Int32"))
    {
        whereFIND = whereFIND + " " + CT.Pref + CT.FindColumn + "=" +
CT.FindStr;
    }

    if (type.Contains("System.String"))
    {
        whereFIND = whereFIND + " " + CT.Pref + CT.FindColumn + " LIKE '" +
CT.FindStr + "%'";
    }

    if (type.Contains("System.DateTime"))
    {
        //          whereFIND = whereFIND + " " + CT.Pref + CT.FindColumn
+ "=" + CT.FindStr + """;
        whereFIND = whereFIND + " " + CT.Pref + CT.FindColumn + " between
'" + CT.FindStr + " 00:00' and '" + CT.FindStr + " 23:59'";
    }

    if (type.Contains("System.Boolean"))
    {

```

```

//            if (CT.FindStr.Contains("True"))
if ((CT.FindStr == "1") || (CT.FindStr.ToUpper() == "TRUE") ||
(CT.FindStr.ToUpper() == "ДА"))
    whereFIND = whereFIND + " " + CT.Pref + CT.FindColumn + "=" + 1;
else
    whereFIND = whereFIND + " (" + CT.Pref + CT.FindColumn + "=" + 0
+ " OR " + CT.Pref + CT.FindColumn + " is NULL )";
}

```

```

SqlCommand cmdFind = new SqlCommand();
cmdFind.Connection = Conn;
cmdFind.CommandTimeout = 600;
cmdFind.CommandText = CT.SQLText;
if (flWere)
    cmdFind.CommandText = cmd.CommandText + " AND " + whereFIND;
else
    cmdFind.CommandText = cmd.CommandText + " where " + whereFIND;

```

```

SqlDataReader RD = cmdFind.ExecuteReader();

```

```

while (RD.Read())
{
    CT.LocateID = Convert.ToInt32(RD["id"].ToString());
    break;
}
RD.Close();
cmdFind.Dispose();

```

```

this.bindingSource.Position = this.bindingSource.Find(CT.PrimaryKeyName,
CT.LocateID.ToString());
dgv.CurrentCell
dgv.Rows[dgv.CurrentCell.RowIndex].Cells[CT.FindColumn];
//            dgv.CurrentCell.C
} //FIND

```

```

        CT.LocateID = -1; CT.FindColumn = ""; CT.FindStr = "";
toolStripTextBoxFind.Text = String.Empty;
        #endregion
    }
    catch (Exception e2)
    {
        MessageBox.Show(e2.Message);
//        MessageBox.Show(e2.InnerException.ToString());
        Rez = false;
        CT.LocateID = -1; CT.LocateColumn = "";
        CT.FindColumn = ""; CT.FindStr = ""; toolStripTextBoxFind.Text =
String.Empty;
    }
    cmd.Dispose();

    /*        if (Conn.State == ConnectionState.Open)
            Conn.Close();
        Conn.Dispose();*/

    //        SendKeys.Send("{TAB}");

    return Rez;
}

// Получить SQL запрос INSERT по полям DataGridView
// у Tag колонки д.б. true
public string GetInsertStr(DataGridView dgvTemp)
{
    bool FLField = false;
    string Str = "";
    string Str1 = "insert into " + CT.TableName + " ( ";
    string Str2 = " values ( ";

    foreach (DataGridViewColumn col in dgvTemp.Columns)
    {

```

```

string FlUpd = Procs.GetAttribValue("UPD:", col.Tag.ToString(), 1);
if (FlUpd == "0") continue; // пропустить поле

FLField = true;
Str1 = Str1 + col.DataPropertyName + ", ";
Str2 = Str2 + "@" + col.DataPropertyName + ", ";
}
if (FLField) //есть хоть одно поле для добавления
{
    Str1 = Str1.Trim(); //удалить пробелы с начала и конца строки
    Str2 = Str2.TrimEnd(' ');

    Str1 = Str1.Remove(Str1.Length - 1); //удалить последнюю запятую
    Str2 = Str2.Remove(Str2.Length - 1);

    Str1 = Str1 + " ";
    Str2 = Str2 + " ";
    Str = Str1 + Str2;
}

return Str;
}

```

ПРИЛОЖЕНИЕ Б

Презентация

Программа для обработки, хранения и визуализации информации

Автор работы: магистрант группы 20ВВм1 Щетинин К.А.
Научный руководитель: к.т.н., доцент Р.Н. Федюнин

Пенза, 2022

Цель и объект работы

Цель работы: разработка информационной системы управления электронным документооборотом предприятия.

Объект работы: информационная система, реализующая следующие возможности:

- ведение дополнительных справочников;
- ведение организаций;
- ведение физических лиц (ФЛ);
- ведение различных параметров вводимой документации
- возможность прикрепления «электронных» файлов различных форматов
- ведение журнала документов
- внесение пользователей и их авторизация.

Система электронного документооборота

Система электронного документооборота (СЭД)– это «многопользовательская» программа, облегчающая управление предприятия в сфере учета различных документов, применяемых в жизнедеятельности организации.

2

Задачи системы электронного документооборота

- обеспечение эффективного управления
- поддержка системы контроля качества
- накопление информации
- управление данными
- регламентирование доступа
- формализация деятельности каждого сотрудника
- оптимизация бизнес-процессов
- экономия ресурсов за счет сокращения расходов

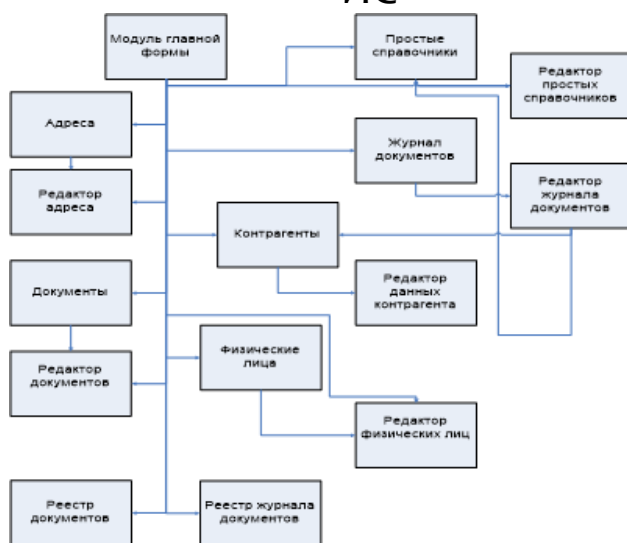
3

Информационные системы

Информационная система есть совокупность технического, программного и организационного обеспечения, а также персонала, предназначенная для того, чтобы своевременно обеспечивать надлежащих людей надлежащей информацией

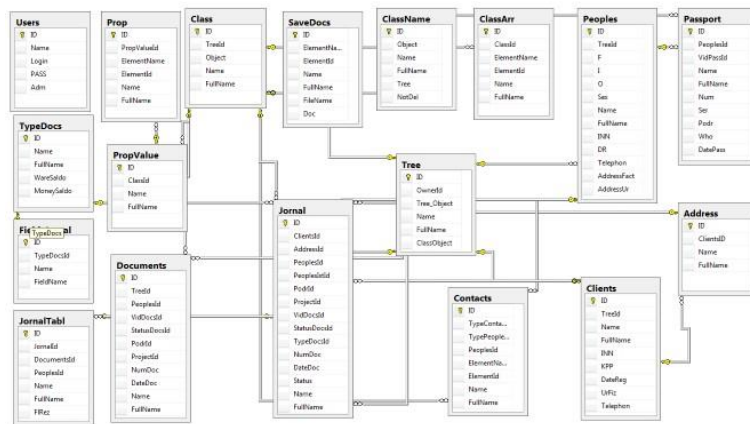
4

Структурная схема разрабатываемой ИС



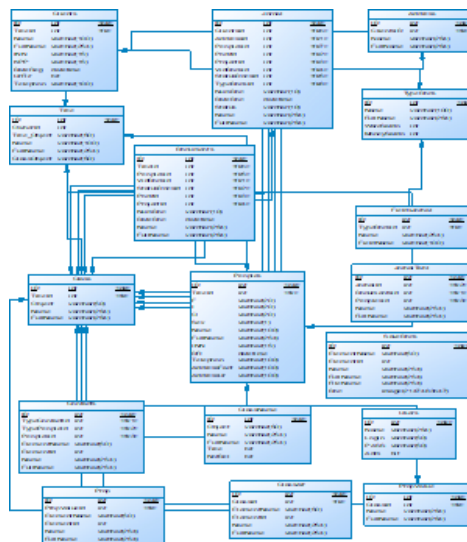
5

Модель данных ИС



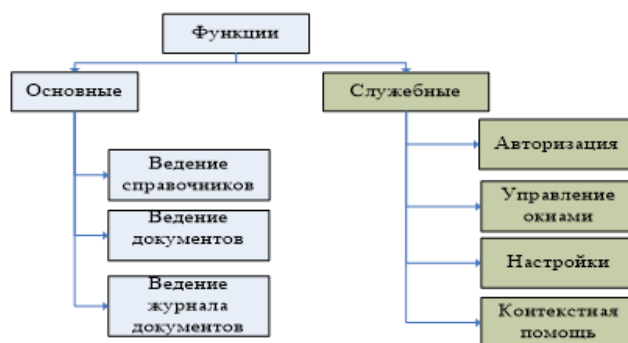
6

ER-модель ИС



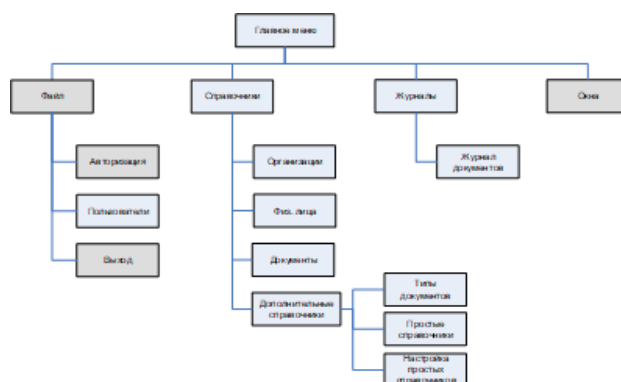
7

Дерево функций ИС



8

Использование ИС



9

Заключение

В ходе проведённой работы были достигнуты следующий результат:

- была спроектирована и реализована ИС учета документов решаящую широкий спектр задач внесения и учета документов различного типа и с различными реквизитами.

10

Спасибо за внимание!