

# 1 Оглавление

1	Оглавление.....	2
2	Аннотация.....	3
3	Введение .....	4
4	Глава 1. Литературный обзор .....	7
4.1	Обзор существующих решений .....	7
4.2	Постановка задачи.....	13
4.3	Выбор архитектуры новой системы автоматизации.....	14
4.4	Выбор структуры информационной базы и средств её реализации ...	17
5	Глава 2. Специальная часть .....	19
5.1	Разработка структуры базы данных.....	19
5.2	Диаграмма классов .....	25
5.3	Разработка приложения .....	26
6	Глава 3. Обзор пользовательских форм.....	34
7	Заключение .....	52
8	Список литературы и интернет ресурсов .....	<b>Ошибка! Закладка не определена.</b>

## **2 Аннотация**

Выпускная квалификационная работа посвящена разработке системы автоматизированного электронного хранения договоров организации, предназначенной для ведения документации одного из подразделений ОАО «Российские Космические Системы» и выполнении преопределённых отчётов.

В данной работе описывается актуальность темы, постановка задачи и описание планируемого функционала, обоснование выбора архитектуры информационной системы, описание процесса проектирования и реализации системы, описание интерфейсов с примерами.

### 3 Введение

Открытое акционерное общество «Российская корпорация ракетно-космического приборостроения и информационных систем» (далее ОАО РКС), основанное в 1946г., стала одной из первых организаций ракетно-космической отрасли нашей страны по разработке сложной наукоемкой продукции. Организация обладает уникальным опытом разработки, изготовления, авторского сопровождения и эксплуатации космических и наземных систем различного назначения. (1)

В услуги ОАО «Российские космические системы» входит:

- создание, развитие и целевое использование глобальной навигационной системы ГЛОНАСС;
- космические системы: поиска и спасания КОСПАС-САРСАТ, геодезии, гидрометеорологического обеспечения, связи и ретрансляции, дистанционного зондирования Земли, планет и других космических объектов, радиотехнического обеспечения научных исследований космического пространства;
- бортовые и оптоэлектронные приборы и комплексы космических аппаратов специального и гражданского назначения;
- наземные пункты приема информации дистанционного зондирования Земли, автоматизированные комплексы управления космическими аппаратами, ракетами-носителями и разгонными блоками;
- полигонные измерительные комплексы;
- автоматизированные системы управления, измерения, мониторинга ресурсов и объектов.

Организация имеет большое количество подразделений специализирующихся на разработке сложной наукоёмкой продукции. Для сопровождения, согласования и учёта технических договоров, организационной рассылки документации требуется система, способная эффективно обрабатывать большое количество документов.

Основу информационной среды любого предприятия, организации или учреждения составляют документы. Содержащаяся в них информация будет обладать юридической силой и может быть использована в профессиональной деятельности только при соблюдении ряда обязательных делопроизводственных операций.

Эффективный документооборот является обязательной составляющей эффективного управления организацией. Документооборот исключительно важен для правильной организации финансового и управленческого учета, его нельзя рассматривать в отрыве от специфических бизнес-процессов конкретной организации.

При работе с документами ежедневно приходится решать большой комплекс вопросов, связанных с регистрацией входящих, исходящих и внутренних документов, с подписанием, согласованием, отправкой, формированием их в дела, определением сроков хранения, передачей в архив либо уничтожением. Без правильной организации работы невозможно справиться с потоком документов, быстро найти требуемый документ, навести по нему справки, а также проконтролировать его исполнение или обеспечить сохранность.

В связи с внедрением в делопроизводство новых технологий очень важны знания об организации безбумажного делопроизводства, организация которого требует не только применения современных технических средств (компьютеров, компьютерных сетей и программ), приобретения дополнительных знаний о создании, использовании, хранении документов на машинных носителях.

При внедрении системы электронного документооборота важно понимать, что ее основное назначение не в экономии расходных материалов, а в организации современной и качественной структуры управления, контроля и анализа хозяйственной деятельности компании. Главными преимуществами при работе с системой электронного документооборота являются (2):

- быстрый доступ к определенной категории документов;
- всеобъемлющий контроль над документооборотом;
- исключение неумышленных ошибок;
- автоматизация составления отчетности движения документов среди подразделений организации;
- общий доступ к базе данных и совместная обработка документов;
- интеграция СЭД с корпоративными информационными системами организации;
- поиск документов в базе данных по заданным атрибутам;
- сокращение материальных расходов организации;
- организация безопасной мобильной работы сотрудников с документами;
- интеграция с офисными программами.

К минусам относят большие расходы на приобретение и внедрение ПО. Так же, сотрудникам необходимо время, чтобы освоить и эффективно использовать систему. Но эти недостатки – временное явление, которые сильно окупаются спустя несколько месяцев.

## 4 Глава 1. Литературный обзор

### 4.1 Обзор существующих решений

Электронный документ – это некий набор информации (текст, изображение, файлы Word, Excel и т.п.), сохраненный на компьютере. Этот набор информации сопровождается карточкой с атрибутами, по которым документ можно быстро найти.

Электронный документооборот (ЭДО) — совокупность автоматизированных процессов по работе с документами, представленными в электронном виде, с реализацией концепции «безбумажного делопроизводства».

Организации, работающие в системах электронного документооборота, имеют возможность вести упрощенное делопроизводство и сосредоточиться на бизнес-процессах. То есть регистрация документов и контроль поручений ответственным лицом (исполнителем) используется только при работе с исходящей и входящей корреспонденцией, для внутренних документов эти этапы исключаются. Исполнение тех или иных поручений осуществляется в рамках задач и заданий, отслеживая состояние которых можно также контролировать прохождение процесса. При этом возрастает количество горизонтальных задач, в рамках типовых бизнес-процессов, при которых не требуется прохождение цепочек согласования с вышестоящими руководителями.

В качестве потенциальных претендентов для решения основных задач были рассмотрены и проведен анализ функциональных возможностей систем СЭД «ДЕЛО», Docsvision 5 и 1С:Документооборот.

#### 4.1.1 «ДЕЛО»

Система «ДЕЛО», – комплексное промышленное решение, обеспечивающее автоматизацию процесса делопроизводства, а также ведение полностью электронного документооборота организации. Система эффективно используется как в небольших коммерческих компаниях, так и в распределенных холдинговых или ведомственных структурах (3).

Система «ДЕЛО» представляет собой законченный тиражируемый «коробочный продукт», который решает задачи автоматизации делопроизводства и электронного документооборота большинства организаций. Удобство и простота внедрения электронного документооборота и дальнейшей работы пользователей обеспечиваются благодаря уникальным эксплуатационным качествам системы, а также профессионализму сотрудников компании:

- легкость и быстрота установки и настройки программы делопроизводства (в том числе силами заказчика), короткие сроки ввода в промышленную эксплуатацию;
- многократно проверенная на практике технология внедрения системы электронного документооборота фирмы и обучения работе с ней пользователей и администраторов;
- квалифицированные и опытные специалисты по внедрению, тренингу и консультанты;
- служба технической поддержки клиентов.

На основе стандартной «коробочной версии» может быть также реализована заказная система автоматизации электронного документооборота, полностью учитывающая конкретные особенности деятельности организации независимо от масштаба, вида деятельности и формы собственности. Пользователи данного продукта получают уже готовую и настроенную систему, а консультанты и техническая поддержка, в свою очередь, помогут разобраться во всём функционале.

Так же, существует демо-версия продукта. Временная версия полностью аналогична полной версии по своим функциональным возможностям. Срок ее использования ограничивается 3 месяцами. Этого времени достаточно чтобы полностью проверить подходит ли данная система для нужд компании.

Стоимость лицензии одного рабочего места в локальной вычислительной сети имеет разумную цену. Расчет полной стоимости лицензии производится нарастающим итогом. Но для большого числа рабочих мест общая стоимость является достаточно высокой.

#### **4.1.2 Docsvision 5**

Система Docsvision 5 предназначена для автоматизации управления документами и бизнес-процессами, включая как общую управленческую деятельность, так и различные функциональные задачи подразделений, и операционные процессы в деятельности организации (4).

Docsvision 5 может применяться на различных предприятиях и в организациях, независимо от их вида деятельности и размеров. Везде есть документы, задания и процессы – базовые объекты, из которых строятся решения на основе Docsvision 5. Специфика различных отраслевых и функциональных задач отражается в расширяемом наборе приложений на платформе Docsvision 5, выпускаемых компанией «ДоксВижн», её партнёрами, а также в заказных решениях, создаваемых при внедрении.

Приложение «Управление документами» – готовое прикладное решение для электронного документооборота. Оно входит в платформу и служит основой для других, более специализированных, приложений Docsvision 5 и заказных решений, создаваемых при внедрении.

В нём реализованы основные функции электронного документооборота, востребованные в любой организации. С ним вы можете сделать повседневную работу с электронными документами и заданиями удобной и эффективной.



В приложении доступны все основные действия с электронными документами:

- надежное хранение и удобная классификация;
- гибкое разграничение прав доступа;
- изменение документов и сохранение их версий;
- поиск по атрибутам и тексту документов;
- согласование, подписание и утверждение документов, а также ознакомление с ними сотрудников;
- наложение электронной подписи на документ;
- работа с заданиями: их создание, исполнение и контроль;
- возможна полная интеграция с «1С:Предприятие 8».

компания предоставляет множество специальных предложений, в том числе и систему немного упрощённую версию ПО. Docsvision Экспресс – система электронного документооборота для компаний или подразделений компаний с лимитированным числом пользователей (5-20 сотрудников). Данное предложение можно использовать для оценки возможностей СЭД без крупных финансовых вложений.

Цена на полную версию продукта закрыта от общего доступа. Однако, цена лицензии Экспресс версии на 5 рабочих мест на срок до года сопоставима по цене с аналогичной системой на одно рабочее место. А приобретение лицензии без ограничения срока действия в три раза дороже.

#### **4.1.3 «1С:Документооборот 8 КОРП»**

«1С:Документооборот 8 КОРП» ориентирован на бюджетные учреждения, а также средние и крупные коммерческие предприятия и предназначен для комплексного решения широкого спектра задач автоматизации учета документов, взаимодействия сотрудников, контроля и анализа исполнительской дисциплины (5).

Учет документов ведется в разрезе видов документов, в соответствии с положением о документообороте предприятия. Принципы учета входящих,

исходящих и внутренних документов, заложенные в программу, соответствуют российскому законодательству, ГОСТам, рекомендациям Росархива и отечественной делопроизводственной практике.

«1С:Документооборот 8 КОРП» не имеет отраслевой специфики и может эффективно использоваться как в бюджетном секторе, так и на коммерческих предприятиях, будь то распределенная холдинговая структура с большим количеством пользователей или среднее предприятие.

Система поддерживает многопользовательскую работу в локальной сети или через Интернет, в том числе и через веб-браузеры.

Система имеет широкий функционал, позволяющий использовать её для различных видов компаний. В данном случае будет рассмотрен учёт договоров. В системе «1С:Документообороте 8 КОРП» автоматизирован полный жизненный цикл договорных документов:

- подготовка проекта договора;
- согласование проекта договора, как внешнее, так и внутреннее;
- учет всех связанных с договором документов, например, акты, дополнительные соглашения;
- учет и контроль сроков действия договоров;
- автоматическое продление сроков действия по правилам, указанным в договоре;
- мультивалютный учет сумм договоров;
- учет и контроль исполнения финансовых обязательств по договору;
- контроль наличия сопроводительных документов по договору, например, счет-фактура;
- контроль возврата переданных экземпляров документов по договору;
- расторжение договора.

Для анализа договорной работы предусмотрены такие отчеты, как «Список заключенных договоров», «Договоры с истекающим сроком действия», «Сопроводительные документы по договору», «Динамика

количества заключенных договоров», «Динамика сумм заключенных договоров», «Расторгнутые договоры».

Цена на систему документооборота от 1С в десять раз больше, чем у конкурентов, без указания количества рабочих мест. Так же, существуют специальное предложение на 100 рабочих мест по более выгодной цене за штуку, но количество рабочих мест под требуемую систему гораздо меньше.

#### 4.1.4 Сравнение рассматриваемых систем

Результаты сравнения функциональных возможностей СЭД по указанным критериям представлены в приведенной ниже таблице.

<b>Функциональные возможности</b>	<b>«ДЕЛО»</b>	<b>Docsvision 5</b>	<b>1С:Документооборот</b>
Наличие Web-клиента	+	Приобретается отдельно	+
Ведение договоров	+	Приобретается отдельно	+
Управление работами	+	+	Только в расширенной версии
Интеграция с 1С	+	Приобретается отдельно	+
Архив документов	+	+	+
Распознавание документов (OCR)	Функция не заявлена	Приобретается отдельно	Приобретается отдельно
Итоговая стоимость в минимальной комплектации (10 раб.)	94 490 руб.	58 590 руб.	195 890 руб.
Итоговая стоимость в максимальной комплектации (200 раб.)	1 544 490 руб.	919 490 руб.	1 084 490 руб.

## 4.2 Постановка задачи

Так как рассмотренные системы документооборота имеют более широкий функционал, чем требуется в настоящее время, а финансовые затраты на покупку, внедрение системы и обучение персонала могут оказаться слишком велики, руководством было принято решение о внедрении системы электронного документооборота в одном из подразделении Организации. Однако Организация имеет квалифицированный штат сотрудников, которые способны самостоятельно разработать автоматизированную систему электронного хранения договоров Организации.

Разрабатываемая система должна соответствовать следующим требованиям:

- быть полностью совместимой с операционной системой Windows 7 и выше;
- иметь локальную базу данных, хранимую на рабочем месте;
- обладать возможностью получения свободных отчётов по данным договоров;
- поддерживать несколько пользователей, разделённых ролями и правами доступа;
- иметь понятный интерфейс для удобного использования;
- обладать читаемым программным кодом для последующего наращивания функционала.

### 4.3 Выбор архитектуры новой системы автоматизации

Выбор средств проектирования и разработки программного обеспечения являются одними из важнейших задач создания автоматизированной системы.

Организация ОАО «Российские космические системы» использует программные продукты фирмы Microsoft.

#### 4.3.1 C#

Для создания системы автоматизированного электронного хранения договоров организации применена среда программирования C#. Она получила в последние годы большую популярность, характеризуется следующими положительными особенностями (6):

- Главной особенностью языка C# является его ориентированность на платформу Microsoft.
- Сочетает объектно-ориентированные и контекстно-ориентированные концепции. Язык хорошо зарекомендовал себя в проектах многих типов, в том числе для разработки пользовательских приложений и баз данных.
- Предоставляет широкий выбор компонент для построения и работы с различными СУБД.
- Имеет высокопроизводительный компилятор в машинный код.
- Является хорошей основой для создания устойчивых к сбоям различных программных продуктов, таких как: настольные приложения, Web сервисы, высокопроизводительные решения автоматизации бизнес-процессов.

### 4.3.2 MS SQL Server 2008

Microsoft SQL Server — система управления реляционными базами данных (СУРБД), разработанная корпорацией Microsoft. Основным используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия (7).

Microsoft SQL Server - одна из наиболее мощных систем работы с базами данных в архитектуре "клиент-сервер". Особенность системы - работа сервера только в операционных системах ряда Microsoft Windows NT - NT Server 4.0, 2000 Server, Server 2003, при этом клиентская часть может взаимодействовать с сервером Microsoft Windows.

В своем составе система имеет средства создания баз данных, работы с информацией баз данных, экспорт и импорт данных, используя другие системы, резервного копирования и восстановления данных, развитую систему транзакций, систему репликации данных, реляционную подсистему для анализа, оптимизации и выполнения запросов клиентов, систему безопасности для управления правами доступа к объектам базы данных и пр.

Для создания данной системы использовалась Express версия. MS SQL Server Express предоставляет возможность использования локальной базы данных, которая ускоряет и повышает эффективность разработки приложений.

### 4.3.3 MS Visual Studio 2012

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, и др. (8).

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода, добавление новых наборов инструментов.

Microsoft Visual Studio объединяет в себе огромное количество функций, позволяющих осуществлять разработки под операционную систему Windows всех версий. Visual Studio имеет упрощенную программную среду, для которой характерна высокая производительность, не зависящая от особенностей оборудования.

С помощью расширенных средств моделирования, обнаружения и проектирования можно максимально полно описать систему, которая позволит наиболее удачно реализовать конкретную концепцию архитектуры.

## 4.4 Выбор структуры информационной базы и средств её реализации

База данных (БД) — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

БД обеспечивает хранение информации, а также удобный и быстрый доступ к данным. Она представляет собой совокупность данных различного характера, организованных по определенным правилам. Основным структурным компонентом БД является таблица, содержащая записи определенного вида и формы. Каждая запись таблицы содержит всю необходимую информацию об отдельном элементе БД. Такие отдельные структурные элементы называют полями таблицы.

Информация в БД не должна быть:

- противоречивой;
- избыточной;
- не целостной.

Формирование базы данных представляет собой основную часть программы, позволяющую максимально быстро создавать и изменять записи, хранящиеся в базе данных.

В зависимости от вида организации данных различают следующие основные модели представления данных в базе:

- иерархическую;
- сетевую;
- реляционную;
- объектно-ориентированную.

В системе автоматизированного электронного хранения договоров организации будет использоваться реляционная БД.

Реляционные базы данных состоят из таблиц, поля которых представляют собой переменные определённого типа. Каждая запись в



таблицах идентифицирует один объект, связь между которыми устанавливается с помощью полей, указывающих на идентификатор другой записи в БД.

Реляционные базы данных требуют поддержания следующих ограничений целостности:

- каждая строка в таблице должна иметь уникальный идентификатор, называемый первичным ключом;
- атрибуты таблицы, ссылающиеся на первичные ключи других таблиц, должны иметь одно из значений этих первичных ключей (целостность ссылок между таблицами);
- структура взаимосвязей таблиц не должна зависеть от содержимого таблиц.

Персональные СУБД обеспечивают возможность создания локальных БД, работающих на одном компьютере. К персональным СУБД относятся Paradox, dBase, FoxPro, Access и другие.

Многопользовательские СУБД позволяют создавать информационные системы, функционирующие в архитектуре «клиент-сервер». Наиболее известными многопользовательскими СУБД являются Oracle, Informix, SyBase, Microsoft SQL Server, InterBase.

При создании системы автоматизированного электронного хранения договоров организации будет использоваться персональная СУБД Microsoft SQL Server Express, позволяющая создать локальную базу данных, которая будет хранить в себе всю информацию, не подключаясь к общей сети.

Так как система будет разрабатываться в среде программирования Visual C#, то оптимальный вариант СУБД является MS SQL, которые полностью совместимы между собой. Это объясняется еще и тем что, его технические характеристики полностью подходят для реализации БД автоматизированной системы учёта договоров.

## 5 Глава 2. Специальная часть

### 5.1 Разработка структуры базы данных

Реляционная база данных состоит из нескольких таблиц, связь между которыми устанавливается с помощью специальных полей. Первым этапом при создании таблиц является определение набора таблиц и полей, из которых они должны состоять, их типов и размеров.

Каждая таблица содержит идентификатор для корректного обращения и поиска определённой записи.

Для того чтобы создать эффективную и защищённую базу данных, ее необходимо правильно спроектировать. Процесс преобразования отношений БД называется нормализацией.

Процесс нормализации базы данных предназначен для уменьшения избыточности информации, хранящейся в ней, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Конечной целью нормализации является уменьшение потенциальной противоречивости хранимой в базе данных информации. Общее назначение процесса нормализации заключается в следующем:

- исключение некоторых типов избыточности;
- устранение некоторых аномалий обновления;
- разработка проекта базы данных, который является достаточно «качественным» представлением реального мира, интуитивно понятен и может служить хорошей основой для последующего расширения;
- упрощение процедуры применения необходимых ограничений целостности.

После нормализации были созданы следующие таблицы:

### 5.1.1 «Договор»

Таблица «Договор» является ключевой таблицей всей работы. Она содержит в себе информацию о договоре организации, такую как: Номер договора, Дополнительное соглашение, Генеральный заказчик, Вид работ, Тема, Наименование работ, Начало работ, Окончание работ, Количество, Цена, Цена за единицу, Модель цены, Объём собственной работы, Объём К/А, Плановая трудоёмкость, Фактическая трудоёмкость, Страница, Ведущий, Примечание, Редактируется. Поле «Номер договора» используется в качестве уникального идентификатора для записи в БД.

Так же, договор должен включать в себя один и более этапов договора, список иностранных заказчиков и исполнителей договора и набор файлов. Множественные значения в БД выражаются через отношения с другими таблицами.

Имя поля	Тип
Номер	int
Дополнительное соглашение	nvarchar(100)
Генеральный заказчик_id	int
Вид работ	nvarchar(100)
Тема	nvarchar(100)
Наименование работ	nvarchar(100)
Начало работ	date
Окончание работ	date
Количество	int
Цена	money
Цена за единицу	money
Модель цены	int
Объём собственной работы	float
Объём КА	float
Плановая трудоёмкость	float

Фактическая трудоёмкость	Float
Страница	int
Ведущий	nvarchar(100)
Примечание	nvarchar(400)
Редактируется	bit

### 5.1.2 «Юридическое лицо»

Таблица «Юридическое лицо» – одна из основных таблиц, которая хранит данные об организациях, предприятиях, участвующие в заказе проводимых работ или исполнении договора. Содержит название организации и поле, показывающее является ли юридическое лицо иностранным. Каждое юридическое лицо имеет один и более элементов таблицы «Экономический показатель».

Имя поля	Тип
Id	int
Название	nvarchar(100)
Иностранный	bit

### 5.1.3 «Этап договора»

Таблица «Этап договора» содержит определённый набор этапов договора, указанного через его идентификатор, то номер. На каждом этапе договора определяются его начало и окончание работ, цена, аванс, расчёт, плановая трудоёмкость, фактическая трудоёмкость и другие важные характеристики, аналогичны характеристикам договора. Каждый этап договора, так же, содержит в себе один и более элементов таблицы «Учёт материальных ценностей».

Имя поля	Тип
Id	int
Договор_id	int
Номер	int

Начало работ	date
Окончание работ	date
Количество	int
Цена	money
Модель цены	int
Заключение	nvarchar(400)
Аванс	money
Расчёт	money
Плановая трудоёмкость	float
Фактическая трудоёмкость	float
Текущее состояние	int
Номер акта	int
Номер удостоверения	nvarchar(100)

#### 5.1.4 «Учёт материальных ценностей»

Таблица «Учёт материальных ценностей» содержит информацию о вложениях организации в часть имущества, здания, помещения, оборудование и другие ценности, имеющие материально вещественную форму, предоставляемые организацией за плату во временное владение.

Имя поля	Тип
Id	int
Этап договора_id	int
Наименование	nvarchar(100)
Цена	money
Местонахождение	nvarchar(100)
Документы	nvarchar(100)
Акт изготовления	nvarchar(100)
Накладная	nvarchar(100)

### 5.1.5 «Иностранный заказчик»

Таблица «Иностранный заказчик» это связующая таблица, содержащая в себе идентификатор договора и юридического лица. Таким образом, используя вспомогательную таблицу, было построено отношение «многие ко многим», с помощью которого впоследствии можно быстро осуществлять поисковые запросы.

Имя поля	Тип
Id	int
Договор_id	int
Юридическое лицо_id	int

### 5.1.6 «Исполнитель договора»

Таблица «Исполнитель договора» – ещё одна связующая таблица, задающее отношение «многие ко многим». Она описывает набор юридических лиц, которым поручено исполнение договора.

Имя поля	Тип
Id	int
Договор_id	int
Юридическое лицо_id	int

### 5.1.7 «Экономический показатель»

Таблица «Экономический показатель» характеризует величины или характеристики, показывающие состояние экономики юридического лица за определённый промежуток времени, в данном случае за год.

Имя поля	Тип
Id	int
Юридическое лицо_id	int
Год	int
РОТ	money
Социальное страхование	float

Накладные расходы	float
Рентабельность	float
Лимит трудоёмкости	float

### 5.1.8 «Файл договора»

Таблица «Файл договора» содержит данные файла договора, загружаемого в БД. Загружаемые файлы могут быть любого формата, такими как: отсканированные страницы документа, электронная версия документа, электронная таблица и другие. В таблице содержится название файла, данные файла и его порядок сортировки в данной системе.

Имя поля	Тип
Id	int
Договор_id	int
Файл	image
Название	nvarchar(50)
Порядок	int

### 5.1.9 «Пользователь»

Таблица «Пользователь» хранит информацию о пользователях, которые допущены к работе с автоматизированной системой электронного хранения договоров и их права доступа. Пароль автоматически шифруется для уменьшения шанса взлома системы.

Имя поля	Тип
Id	int
Имя	nvarchar(25)
Пароль	uniqueidentifier
Права	int

## 5.2 Диаграмма классов

Используя полученные данные о таблицах, полей и их типах данных была построена диаграмма классов, наглядно демонстрирующая всю информацию об архитектуре БД, включающая в себе так же отношения между таблицами (Рисунок 1).

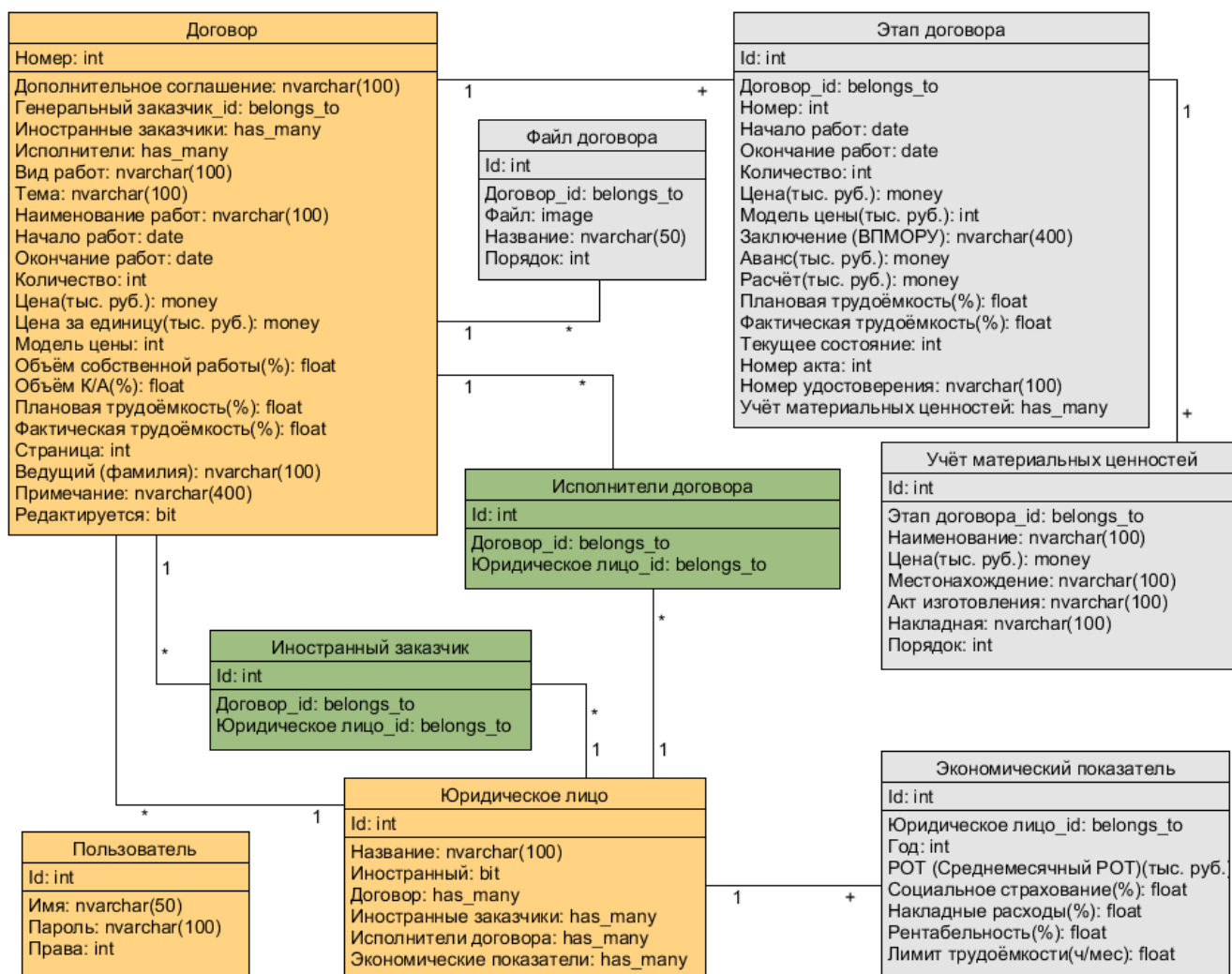


Рисунок 1. Диаграмма классов системы.



### 5.3 Разработка приложения

Выбор среды программирования Microsoft Visual Studio существенно упростил разработку автоматизированной системы учёта договоров организации, в силу того, что к моменту написания выпускной квалификационной работы был накоплен опыт программирования в этой среде. К тому же, приложения, созданные в среде разработки Microsoft Visual Studio, имеют максимальную совместимость с операционной системой Windows, поскольку язык программирования Visual C#, использующийся в этой среде программирования, является реализацией языка C# корпорацией Майкрософт. Выбранный язык программирования прост в освоении, типобезопасен и объектно-ориентирован, что положительно сказывается на качестве реализованных программных продуктов (9).

Для облегчения дальнейшей разработки, поддержки и расширения системы, было создано несколько классов:

#### 5.3.1 BoxItem

Класс-контейнер, имеющий две переменные – «Text» и «Value». Используется в элементах управления ListBox и ComboBox для получения системной информации о выбранном элементе списка, например id договора или юридического лица. Без него работа со списками была бы неэффективна.

#### 5.3.2 CheckNull

Данный класс предназначен для проверки значений на пустоту и конвертации значения к определённому типу перед записью в БД. Он содержит статичные функции, которым подаются значения элементов управления формы. Основная часть пользовательских элементов, характеризующих данные объекта, являются текстовыми полями textbox, поэтому на вход обрабатывающих функций подаются строки этих элементов. Функция принимает значение элемента управления, проверяет, является ли оно пустым, и возвращает либо DBNull (аналог null в БД), либо

конвертированное значение в необходимый тип данных. Класс содержит следующие методы:

- String, Int, Float, Decimal – преобразование в стандартные типы данных.
- File – типы данных byte[] предназначен для хранения файлов. Не конвертируется, только проверяется на пустоту.
- DateTime – формат данных даты и времени. Время обнуляется.
- Combo – выпадающий список. Возвращает системную информацию выбранного элемента.

### 5.3.3 Crypto

Был создан отдельный класс, имеющий статические функции для шифрования и дешифрования текста криптографическим алгоритмом AES. В реализации функций была применена встроенная библиотека «Cryptography». Данный алгоритм хорошо подходит для шифрования произвольного текста любым паролем. Последовательность действий в этом случае такая:

1. Нужно получить хэш от пароля
2. По специальным правилам, описанным в стандарте AES, преобразовать хэш в ключ
3. Разбить текст на блоки по 128 бит
4. Зашифровать каждый блок полученным ключом
5. Объединить получившиеся блоки в конечную строку

Основываясь на полученных знаниях, были созданы функции «EncryptStringAES» и «DecryptStringAES». Первая, в качестве входных аргументов, принимает текст, требуемый для шифрования, и пароль, а вторая функция – зашифрованную строку и пароль (10).

### 5.3.4 Extensions

Это класс, в котором содержится набор функций, расширяющий существующие классы. Благодаря этому классу были добавлены статические функции, имеющие в качестве первого аргумента указатель self.<Название другого класса>. Вызов такой функции производится из экземпляра класса,

но, фактически, функция находится в классе «Extensions». Этот класс содержит функции:

- ToString2 – применяется к массиву. Возвращает строку, содержащую список элементов массива в удобном для пользователя виде. Используется только для отладки массивов.
- isnull – применяется к любому объекту. Возвращает «истину», если объект равен null или имеет тип DBNull, иначе «ложь».
- GetListBoxSelected – применяется к элементу управления ListBox. Функция возвращает массив значений элементов списка, которые выбраны пользователем.
- AddArrayParameters<T> – применяется к объекту SQL запроса. К параметру, включающего в себя аргументы запроса, функция добавляет массив параметров, преобразуя элементы массива типа <T> к типу, возможному для его передачи в качестве аргумента запроса.
- AddUserControl – существует перегрузка метода, применяющегося к объекту, характеризующий запрос к БД. Функция применяется к классу «UserControlFileEdit» и к «UserControlFormStage», добавляя все переменные пользовательского элемента управления, описывающего вложенную таблицу данных к элементу таблицы «Договор», в запрос.
- LoadFile – создаёт поток данных, загружая файл, по указанному пути, в оперативную память.
- SaveFile – создаёт поток данных, сохраняя массив бинарных данных, являющийся файлом, по указанному пути.

### 5.3.5 Fill

Содержит набор функций, которые упрощают заполнение элементов управления данными, полученными от БД. Класс создан для корректного заполнения элементов управления, таких как текстовые поля, значениями, предварительно преобразовав их в текстовую форму. Каждая функция проверяет значение на пустоту, затем, если равно не null, записывает в поле

текстовое представление значения, иначе записывает пустой текст. Набор функций этого класса такой: `TextBox`, `TextBoxInt`, `TextBoxFloat`, `TextBoxDecimal`, `DateTimePicker`.

### 5.3.6 Global

Данный класс содержит глобальные переменные системы, такие как: основные системные цвета, минимальную дату, строка «пустого узла запроса», строка подключения к БД, последняя директория и схема таблицы «Договор».

### 5.3.7 OwnTabControl и OwnTabPage

Эти классы являются наследниками стандартных классов Visual C# – «`TabControl`» и «`TabPage`», соответственно. В классе «`OwnTabControl`» переопределены функции отображения элемента и компонентов вкладок. Класс «`OwnTabPage`» является модифицированным классом, описывающий вкладки, для которого была добавлена переменная, хранящая текущий цвет заголовка вкладки. Так же были добавлены функции для смены и сброса цвета вкладки.

### 5.3.8 OwnTreeView и OwnTreeNode

Как и предыдущие, «`OwnTreeView`» и «`OwnTreeNode`» наследуются от классов со схожими названиями «`TreeView`» и «`TreeNode`». Стандартных возможностей было не достаточно, поэтому классы были расширены, переопределяя некоторые функции. Класс «`OwnTreeView`» используется в форме составления отчётов и представляет собой список атрибутов таблицы «Договор» с возможностью пометки любого из атрибутов для его последующего добавления в отчёт. Функция отображения списка была изменена, поскольку такой атрибут, как «Номер договора» должен обязательно присутствовать в отчёте, и требовалось убрать возможность для пользователя отключать его. Класс «`OwnTreeNode`» имеет дополнительную переменную, с помощью которой можно скрыть элемент метки.

### 5.3.9 SQL

Этот класс разработан для исключения из программы однотипных фрагментов кода, описывающих обращение к БД. Он содержит большой набор статических функций для составления различных запросов к БД и получения требуемых данных, такие как:

- LoadSchema – возвращает все атрибуты указанной таблицы в виде массива строк. Используется при инициализации программы для загрузки названия атрибутов таблицы «Договор»;
- GetOneOrder – выполняет запрос по одному атрибуту таблицы и возвращает отсортированный список значений. Направление сортировки указывается с помощью аргумента бинарного типа;
- GetOneFirst – аналогична предыдущей функции, но возвращает первый элемент. Функция используется, когда заведомо известно, что такой элемент единственный;
- GetTwoOrder – выполняет запрос по двум атрибутам таблицы, с возможностью указания дополнительного условия, поля по которому будет происходить сортировка и направления сортировки;
- SelectMinMaxYears – имеет два аргумента с ключевым словом «out», благодаря чему переменные, поданные в качестве аргументов, во время выполнения функции меняют значения на минимальный год начала и максимальный год окончания работ договора;
- GetBeginsYearCount, GetContinuesYearCount и GetEndsYearCount – возвращают количество договоров, сгруппированных по указанному году проведения работ. Указанный год для функции GetBeginsYearCount проверяется в поле начала работ, для GetContinuesYearCount – в промежутке между годами начала и окончания работ, а для GetEndsYearCount – в поле окончания работ;
- GetBeginsNodes, GetContinuesNodes и GetEndsNodes – возвращают список элементов, характеризующие номер договора и даты. GetBeginsNodes возвращает дату начала работ, GetContinuesNodes – даты

начала и окончания работ, а GetEndsNodes – дату окончания работ, с применением сортировки элементов по дате;

- ReadAll – считывает все атрибуты первого объекта в хэш таблицу.

Ключом для итоговой таблицы является название атрибута;

- ReadAllMultiple – аналогична предыдущей функции, за исключением того, что она возвращает список хэш таблиц и имеется возможность указания поля, по которому будет проводиться сортировка;

- ReadManyToMany – возвращает массив целых чисел указанной таблицы. Используется для получения идентификаторов вложенных таблиц, обеспечивающих связь с другими таблицами отношением «многие ко многим»;

- CheckUnique – проверяет требуемую запись на уникальность, возвращая бинарное число;

- ReadToCombo – заполняет выпадающий список данными из таблицы БД, указанными в качестве входных аргументов. Используя дополнительный класс «BoxItem», в качестве отображаемого текста загружается указанный атрибут, а системная информация содержит идентификатор объекта и таблице;

- ReadToList – аналогична функции «ReadToCombo», заполняя пользовательский элемент управления – список;

- InsertManyToMany – сохраняет в БД данные объектов, заданные отношением «многие ко многим»;

- DeleteManyToMany – удаляет из БД данные объектов, заданные отношением «Многие ко многим».

У методов, возвращающих список из двух или трёх атрибутов, элементом списка является объект кортеж «Tuple», содержащий в себе эти объекты атрибутов.

### 5.3.10 Validator

Данный класс был создан, чтобы предотвратить занесение в БД некорректных данных. Перед сохранением данных происходит валидация по значению каждого пользовательского элемента управления, задающего определённый атрибут. Поля с некорректными данными отслеживаются и их названия подсвечиваются красным цветом. Вкладки, в которых находятся такие поля, тоже окрашиваются. Класс состоит из набора статических функций «Apply», перегруженной несколько раз для различных случаев, и нескольких функций проверки строковых значений на соответствие определённым типам данных:

- UInt, Int, Float, Decimal, Filename – эти функции принимают в качестве аргументов строку, полученную от текстового поля, и бинарное число, показывающее, может ли этот атрибут быть пустым, то есть равным null. Благодаря регулярным выражениям, строка проверяется на корректность введённых данных, в соответствии с типом атрибута, к которому она принадлежит. «Filename» проверяет, есть ли в данной строке символы, которые не могут быть использованы в названиях файла в данной операционной системе.
- Apply – функция, которая окрашивает элемент управления красным цветом, в случае некорректно введённых данных, и возвращает «истину», либо «ложь». Существует 4 перегрузки данной функции:
  1. используется, когда нужно проверить значение в текстовом поле на принадлежность определённому типу данных. Помимо типов, описанных выше, функция может принять тип «File», при котором проверяется существование файла по указанному пути;
  2. проверяет пользовательский элемент управления, такие как выпадающий список или поле выбора даты, на непустое значение. Так же, при проверке даты, можно указать ещё один пользовательский элемент выбора даты, чтобы второе значение не могло превышать первое;

3. подаётся двоичное число, по которому определяется результат.
4. В качестве аргументов используется объект класса «OwnTabPage» и двоичное число. При значении «ложь» вкладка окрашивается в красный цвет.



## 6 Глава 3. Обзор пользовательских форм

### 6.1.1 Начальное окно. Окно авторизации.

Система обладает иерархией пользовательских форм. Начальным окном является экран входа в систему, имеющий два поля – имя пользователя и пароль (Рисунок 2).

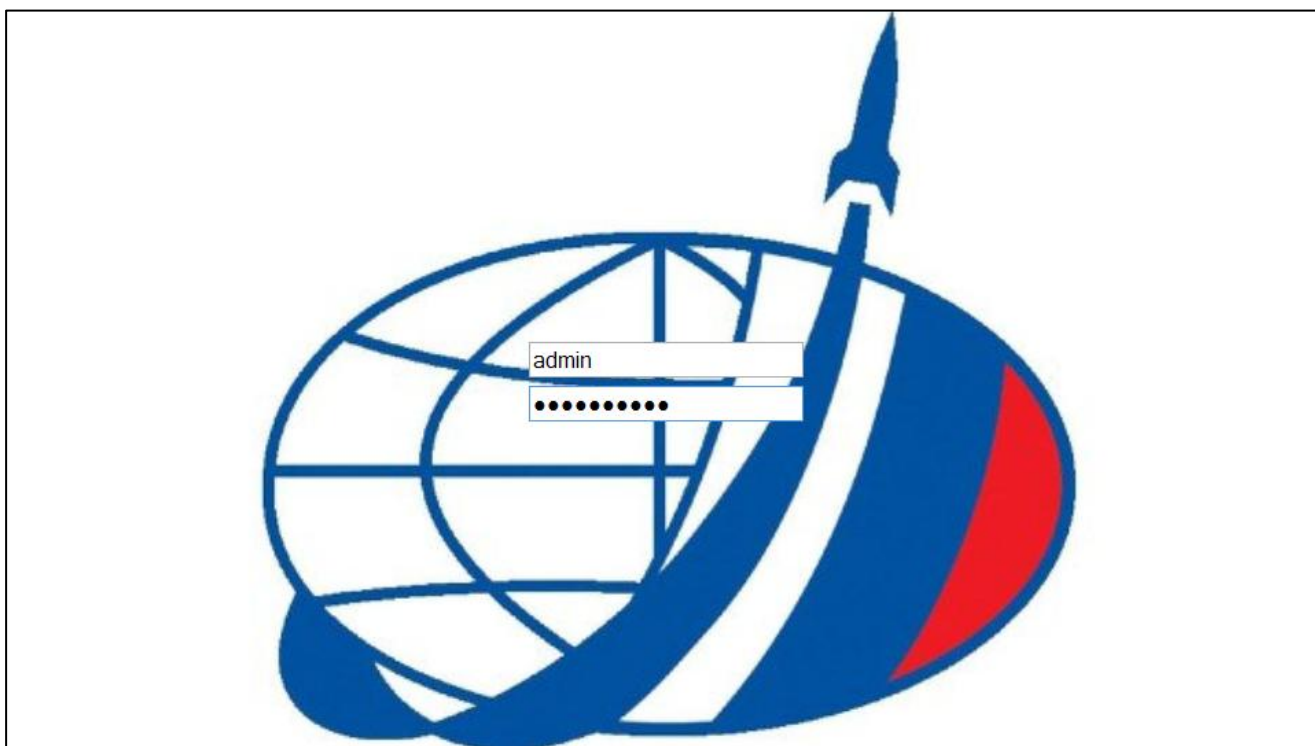


Рисунок 2. Начальное окно. Окно авторизации.

В наше время каждый человек, пользующийся компьютером и различными сервисами, сталкивался с такими понятиями как регистрация, авторизация и аутентификация. В основном, для защиты информации и предоставление доступа к ней только некоторым пользователям, системы используют пару строковых значений – логин и пароль.

Логин и пароль каждого пользователя хранятся в БД, как и все остальные данные системы. Но, в отличие от обычных данных, логин и пароль, в некоторых системах, являются наиболее ценными для злоумышленников. Ведь человек, получивший доступ к базе данных автоматизированной системы, получает всю информацию, хранящейся в ней, а так же доступ к системе и всем её возможностям, включая возможности администратора.

В этом случае целесообразно шифровать пароли. При регистрации нового пользователя или смене пароля, в БД заносится предварительно зашифрованный пароль пользователя.

Из соображений безопасности, в данной выпускной работе для шифрования паролей будет использоваться алгоритм шифрования AES. Это симметричный алгоритм блочного шифрования. Такой алгоритм разбивает данные на блоки размером 128 бит и шифрует каждый блок ключом 128, 192 или 256 бит. На данный момент является одним из самых эффективных методов шифрования.

Данный криптографический алгоритм реализован в классе Crypto, использующий встроенную библиотеку «Cryptography».

### 6.1.2 Основное окно

При успешной авторизации пользователя, ему предоставляется основное окно системы. Причём, для каждой роли, указанной правами доступа пользователя, отображается различная форма.

Рассмотрим пользовательскую форму для «Администратора», поскольку она собрала в себя весь функционал, доступный пользователю, с полными правами доступа (Рисунок 3).

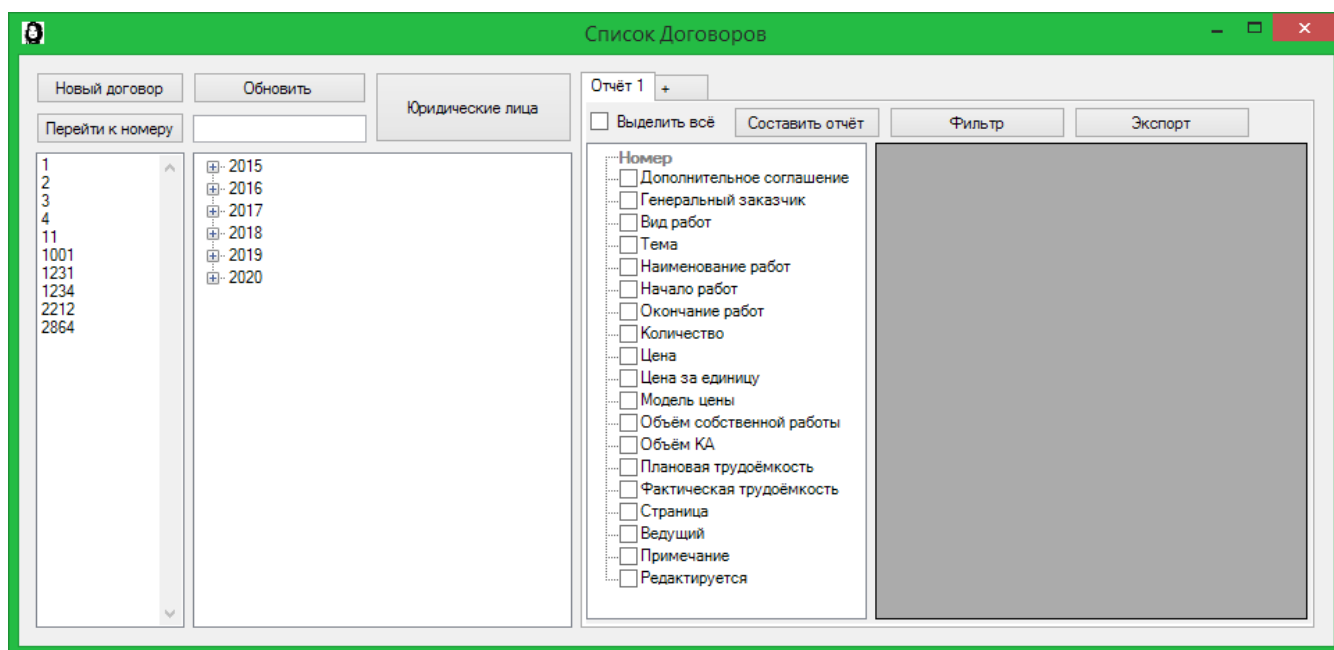


Рисунок 3. Главное окно.

Главное окно предоставляет администратору быстрый доступ ко всем возможностям системы. Из данного меню запускаются основные функции системы:

- Создание нового договора;
- Редактирование договоров;
- Редактирование списка юридических лиц;
- Создание отчётов.

Главное окно можно визуально поделить на две части.

В левой части окна располагаются элементы управления для взаимодействия с данными системы. Основную часть занимают список договоров и древовидное представление списка договоров, сгруппированное по годам (Рисунок 4).

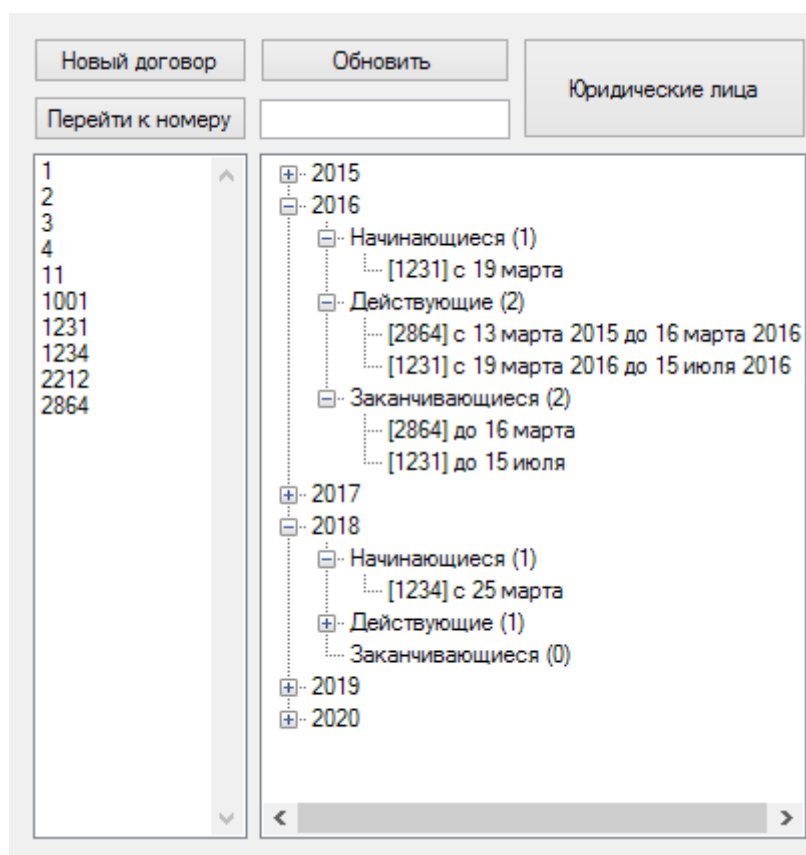


Рисунок 4. Левая часть основного окна.

В списке содержатся номера договоров, в порядке возрастания. При двойном нажатии клавиши мыши на элементе списка, открывается окно редактирования договора, выбранного номера.

Древовидное представление списка договоров устроено сложнее. Первыми узлами дерева выступают года, в которых, так или иначе, действует договор. При составлении первичного списка договоров по годам, в БД поступает запрос, возвращающий минимальный год значения «Начало работ» и максимальный год значения «Окончание работ». Дальше, создаются все узлы, начиная с минимального и заканчивая максимальным годом, и добавляются в древовидный список. Так же, внутри каждого узла помещается «пустой узел запроса» с названием "<<LOAD DATA>>" для возможности последующего развёртывания узла дерева.

При развёртывании такого узла, при наличии внутри «пустого узла запроса», внутренний узел замещается тремя: «Начинающиеся (X)», «Действующие (X)» и «Заканчивающиеся (X)». Где символом «X» обозначается число договоров, относящееся к данной категории и

получаемое запросом к БД. Внутри каждого из трёх узлов, так же, добавляется «пустой узел запроса», если полученное количество договоров больше нуля. В другом случае раскрытие узла является невозможным.

При раскрытии каждого из последующих узлов, если имеется всего один «пустой узел запроса», внутренние узлы удаляются, и выполняется запрос, заполняющий узел списком договоров. Однако, строки договоров каждого из трёх узлов отличаются между собой:

- «Начинающиеся» – "[<Номер договора>] с <День и месяц начала работ>"
- «Действующие» – "[<Номер договора>] с <День и месяц начала работ> до <День и месяц окончания работ>"
- «Заканчивающиеся» – "[<Номер договора>] с <День и месяц окончания работ>"

При двойном клике на узле, представляющем договор, открывается окно редактирования выбранного договора.

Так же, в левой части имеются несколько кнопок:

- «Новый документ» – открывает окно создания нового договора;
- «Обновить» – обновляет список и древовидный список договоров;
- «Перейти к номеру» – открывает окно редактирования договора, номер которого указан в поле рядом с кнопкой. Если договора с указанным номером не существует, то поле подсвечивается красным цветом.

В правой части основного окна располагаются элементы управления для составления свободных отчётов. Составление отчётов является важной частью мониторинга и контроля выполнения работ. В данной системе имеется возможность поддерживать несколько отчётов одновременно, для этого они разделены на разные вкладки. При нажатии на «+» создаётся новая вкладка с формой создания отчёта, а при нажатии правой клавиши мыши закрывается указанная вкладка со всем её содержимым (Рисунок 5).

Отчёт 1   Отчёт 2   Отчёт 3   +

☐ Выделить всё        

Номер	Генеральный заказчик	Начало работ	Окончание работ	Цена
1	Роскосмос	16.03.2015	22.08.2015	1,0000
2	Зенит	01.04.2015	01.04.2015	
3	Зенит	03.03.2015	17.07.2015	
4	ОАО Чиним быстро	07.01.2015	29.08.2015	
11	Зенит	02.04.2015	18.04.2015	
1001	ЗИП	01.06.2015	15.10.2015	
1231	Роскосмос	19.03.2016	15.07.2016	1123,0000
1234	Роскосмос	25.03.2018	25.06.2020	
2212	Роскосмос	08.04.2015	09.04.2015	120,0000
2864	Роскосмос	13.03.2015	16.03.2016	

☐ Дополнительное соглашение  
☒ Генеральный заказчик  
☐ Вид работ  
☐ Тема  
☐ Наименование работ  
☒ Начало работ  
☒ Окончание работ  
☐ Количество  
☒ Цена  
☐ Цена за единицу  
☐ Модель цены  
☐ Объём собственной работы  
☐ Объём КА  
☐ Плановая трудоёмкость  
☐ Фактическая трудоёмкость  
☐ Страница  
☐ Ведущий  
☐ Примечание  
☐ Редактируется

Рисунок 5. Правая часть основного окна.

Форма для создания отчётов включает в себя:

- список атрибутов договора, по которым будет составляться отчёт. Каждый атрибут, кроме номера, имеет элемент под названием checkbox. Используя этот элемент, пользователь может пометить какие из атрибутов договора следует включать в отчёт. Номер по умолчанию всегда присутствует в отчёте.
- флажок «Выделить всё» немного облегчает выбор атрибутов. При активации, все атрибуты помечаются как активные. При повторном нажатии со всех атрибутов снимается выделение.
- элемент отображения данных в настраиваемой сетке DataGridView позволяет выводить данные из базы данных в удобную для пользователя форму. Имеется возможность менять порядок столбцов атрибутов и сортировать строки по выбранному атрибуту, что существенно помогает пользователю при составлении отчётов.
- кнопка «Фильтр» открывает окно фильтра текущей вкладки отчёта. Благодаря фильтру, пользователь имеет возможность более гибко управлять отчётами, отображая только те строки, которые удовлетворяют заданным условиям.

– Кнопка «Экспорт» формирует файл формата «.xlt» (Excel) для сохранения всех данных из таблицы отчёта и предлагает пользователю выбрать местоположение для сохранения файла.

### 6.1.3 Фильтр

Окно фильтра отчёта представляет собой большой набор различных элементов управления. Благодаря фильтру, пользователь имеет возможность отсеять малозначимые записи, оставив в таблице только необходимые.

Элементы, такие как: радио кнопки, текстовые поля, поля выбора даты и выпадающий список, собраны в группы. Каждая такая группа отвечает за определённый атрибут объекта «Договор» и подписана соответствующе (Рисунок 6).

The screenshot shows a software window titled "Фильтр отчёта 1" (Filter report 1). It contains two columns of filter settings. The left column includes filters for "Дополнительное соглашение" (Additional agreement), "Вид работ" (Type of work), "Тема" (Topic), "Наименование работ" (Work name), "Начало работ" (Start of work) with date pickers, "Окончание работ" (End of work) with date pickers, "Количество" (Quantity), "Цена (тыс. руб.)" (Price in thousands of rubles), "Цена за единицу (тыс. руб.)" (Price per unit in thousands of rubles), and "Модель цены" (Price model). The right column includes filters for "Генеральный заказчик" (General contractor) with a dropdown menu, "Объём собственной работы (%)" (Volume of own work (%)), "Объём К/А (%)" (Volume of K/A (%)), "Плановая трудоёмкость (%)" (Planned labor intensity (%)), "Фактическая трудоёмкость (%)" (Actual labor intensity (%)), "Страница" (Page), "Ведущий" (Lead), "Примечание" (Note), and "Редактируется" (Is being edited). At the bottom right, there are four buttons: "Применить фильтр" (Apply filter), "Применить без фильтра" (Apply without filter), "Сбросить" (Reset), and "Отмена" (Cancel).

Рисунок 6. Окно фильтра.

Внутри каждой группы расположены несколько радио кнопок, являющиеся основным элементом управления, с помощью которого можно выбрать один из вариантов условия для фильтрации определённого атрибута.

По умолчанию, в каждой группе выбран вариант «Любой», означающий, что по данному атрибуту фильтрация производиться не будет. Также, в некоторых группах есть вариант «Пустой», при котором будут отфильтровываться записи со значением атрибута не null.

В полях текстовых атрибутов вроде «Дополнительное соглашение», «Вид работ», «Тема», «Ведущий» и «Примечание», пользователь может вписать подстроку, по которой будет происходить поиск.

Поля «Начало работ» и «Окончание работ» не могут быть пустыми, а также, для них есть как выбор определённого дня, месяца и года, с помощью поля выбора дат, так и промежутка времени, используя два поля дат.

Атрибуты «Количество», «Модель цены» и «Страница» целочисленного типа, поэтому для них проверяется правильность введённых данных. В случае, когда в поле введены неверные данные, заголовок группы окрашивается в красный цвет. У атрибута «Количество» можно выбрать промежуток значений.

Поля денежного типа «Цена» и «Цена за единицу», могут быть указаны как точная сумма или как интервал между двумя значениями. Поля, также, должны быть введены корректно.

Значения полей «Объём собственной работы», «Объём К/А», «Плановая трудоёмкость» и «Фактическая трудоёмкость» требуется ввести в виде чисел с плавающей точкой. Поскольку эти атрибуты следует указывать в процентном соотношении, то значения должны быть в пределах от 0,0 до 100,0.

Поскольку последний атрибут «Редактируется» имеет тип Boolean и не может быть пустым, то помимо «Любое» существует ещё два варианта выбора «Да» и «Нет».



Кроме того, пользовательская форма фильтра содержит 4 кнопки, при нажатии которых осуществляются основные действия формы:

- «Применить фильтр» – по выбранным значениям составляется хэш атрибутов. Дополнительно создаётся список характеристик, которые не были выделены для составления отчёта, но были помечены пользователем, как необходимые для условий фильтрации. Список характеристик добавляется к основному запросу, но, впоследствии, эти столбцы атрибутов будут скрыты от пользователя. Когда все данные получены от БД, система проверяет все строки на соответствие условиям фильтрации, оставляя видимыми подходящие. Таким образом, значения в полях фильтра можно изменять, не производя запрос к БД. В случае, когда требуется добавить условие фильтрации, то производится дополнительный неполный запрос, с указанием номера договора и требуемых характеристик, которые добавляются в основную таблицу. После этого система заново проверяет все строки.
- «Применить без фильтра» – окно фильтра закрывается, сохраняя значения, но данные в таблице обновляются, будто условия фильтрации не заданы.
- «Сбросить» – как понятно из названия, сбрасывает значения в окне, то есть выбирает для каждого атрибута значение «Любое».
- «Отмена» – закрывает окно без сохранения значений. То же самое действие происходит, если закрыть окно вручную.

### 6.1.4 Редактирование договора

Окно редактирования договора является самой объёмной формой системы. Она содержит три вкладки, обеспечивающие удобный для пользователя доступ ко всем атрибутам элемента таблицы «Договор», а также к её вложенным атрибутам, связанным отношениями с другими таблицами, в пределах одного окна (Рисунок 7).

Редактирование Договора №1001			
Договор	Этапы	Архив	
Номер*	1001	Количество	1
Дополнительное Соглашение		Цена (тыс. руб.)	86,0000
Генеральный Заказчик*	ЗИП	Цена За Единицу (тыс. руб.)	2,1500
Иностранные Заказчики	Intel 1 ООО Молот Тора	Модель Цены	
Исполнители Договора	Роскосмос Зенит ЗИП Intel 1	Объём Собственной Работы (%)	76
Вид Работ	Расширение области влияния	Объём К/А (%)	34
Тема	Новые владения	Плановая Трудоемкость (%)	84
Наименование Работ		Фактическая Трудоемкость (%)	73
Начало Работ*	1 июня 2015 г.	Страница	
Окончание Работ*	15 октября 2015 г.	Ведущий	Вольнов А.И.
		Примечание	
<div>Сохранить    Удалить</div>			

Рисунок 7. Вкладка «Договор».

Во вкладке «Договор» перечислены атрибуты одноимённой таблицы БД. Большинство этих атрибутов вводятся пользователем в текстовые поля. Для выбора генерального заказчика был использован выпадающий список из списка не иностранных юридических лиц. Выбор дат начала и окончания работ осуществляется посредством элементов управления, позволяющим выбрать дату. Так же, в этой вкладке есть возможность выбрать элементы смежных таблиц, такие как иностранные заказчики и исполнители договора. Эти множественные отношения реализованы с помощью списков, позволяющие выбрать несколько вариантов.

При создании нового договора, текстовое поле «Номер» активно, а при редактировании существующего договора, это поле заблокировано от изменений.

Атрибут «Редактируется» автоматически ставится в значение «истина», если такой договор не открыт другим пользователем в режиме редактирования, иначе пользователю выводится сообщение «Данный договор открыт другим пользователем».

Редктирование Договора №1001

Договор    Этапы    Архив

Этап №1    Этап №2    +

Номер\* 1

Начало Работ\* 1 июня 2015 г.

Окончание Работ\* 28 августа 2015 г.

Количество 2

Цена (тыс. руб.) 37,0000

Модель Цены

Заключение Все работы закончены в срок

Аванс (тыс. руб.) 34,0000

Расчёт (тыс. руб.) 34,0000

Плановая Трудоемкость (%) 68

Фактическая Трудоемкость (%) 63

Текущее состояние

Номер акта 864

Номер удостоверения 246

**Учёт материальных ценностей**

Грузоперевозки    Аренда цеха №4    Аренда оборудования    +

Наименование Грузоперевозки

Цена (тыс. руб.) 12

Местонахождение г. Москва, ул. Авиамоторная, д. 53

Акт изготовления к800

Накладная 1252

Сохранить    Удалить

Рисунок 8. Вкладка «Этапы».

Вкладка «Этапы» включает в себя элементы управления, для редактирования данных объектов смежной таблицы «Этапы договора» текущего договора (Рисунок 8).

Набор этапов представлен в виде элемента управления вкладками, а каждая вкладка описывает атрибуты этапа договора. Все поля, предоставленные в этих вкладках аналогичны полям вкладки «Договор». Поскольку, каждый этап договора связан с таблицей «Учёт материальных

ценностей» отношением «один ко многим», то в каждой вкладке имеется элемент управления вкладками для заполнения данных смежной таблицы.

Вкладки этапов и материальных ценностей имеют возможность добавления дополнительных вкладок нажатием на «+», а так же удаления при нажатии правой клавишей мыши на заголовок вкладки.

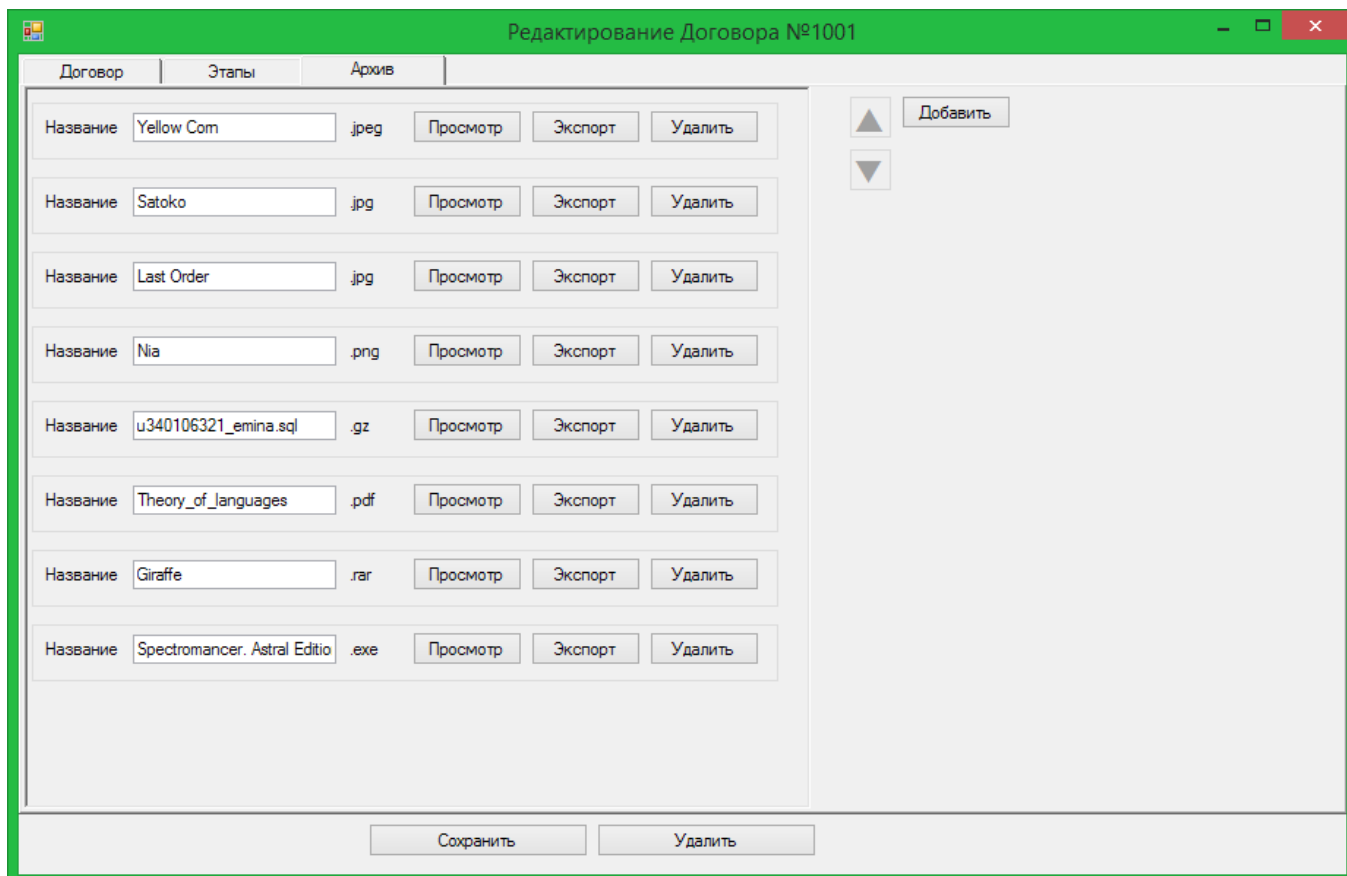


Рисунок 9. Вкладка «Архив».

Третья вкладка «Архив» содержит список групп, содержащих пользовательские элементы управления атрибутами таблицы «Файлы договора», и кнопки управления файлами (Рисунок 9).

В левой части окна находится область динамического отображения пользовательских элементов управления. Каждый файл в этой области представляет собой объект класса «UserControlFile», в котором сгруппированы элементы управления:

- текстовое поле хранит в себе название файла;
- надпись правее поля показывает расширение файла;

- кнопка «Просмотр» сохраняет файл во временную директорию системы «%TEMP%» и открывает его программой по умолчанию;
- кнопка «Экспорт» открывает стандартное сохранения файла окно для выбора директории и названия файла. После подтверждения возникает диалоговое окно, предлагающее пользователю открыть папку с сохранённым файлом. При последующем подтверждении, открывается проводник, с выделенным файлом;
- кнопка «Удалить» деактивирует кнопки и текстовое поле, а её текст меняется на «Восстановить». Если файл помечен для удаления, то при сохранении документа, файл удаляется из БД, а в случае, когда файл только добавлен, то он не просто сохраняется.

При нажатии на внутреннюю часть группы элементов можно выделить файл. Задний фон этой группы меняется на более тёмный, после чего файлам можно менять порядок в этой области, с помощью кнопок, расположенных в правой части окна:

- две кнопки со стрелками вверх и вниз меняют порядок выделенной группы элементов, характеризующей файл;
- кнопка «Добавить» показывает пользователю окно открытия файла. При подтверждении, создаётся группа элементов. В текстовое поле загружается название файла, в надпись записывается расширение файла, а во внутреннюю память записывается двоичное представление файла.

Важно понимать, при инициализации окна, в память загружаются только названия и порядок следования файлов. Двоичное представление файла загружается в память при нажатии на одну из функциональных кнопок.

Ниже основного элемента управления вкладками располагаются две кнопки «Сохранить» и «Удалить». При нажатии на кнопку «Сохранить», прежде чем данные будут записаны в БД, система проверяет каждое введенное пользователем значение на корректность. Каждый атрибут проверяется функцией класса «Validator» в соответствии с его типом данных.

The screenshot shows a software window titled "Редактирование Договора №1001". It features three tabs: "Договор" (selected), "Этапы", and "Архив". The "Договор" tab contains several data entry fields:

- Номер\***: Text field with value "1001".
- Дополнительное Соглашение**: Empty text field.
- Генеральный Заказчик\***: Dropdown menu with "ЗИП" selected.
- Иностранные Заказчики**: List box with "Intel", "1", and "ООО Молот Тора" visible.
- Исполнители Договора**: List box with "Роскосмос", "Зенит", "ЗИП", "Intel", and "1" visible.
- Вид Работ**: Text field with value "Расширение области влияния".
- Тема**: Text field with value "Новые владения".
- Наименование Работ**: Empty text field.
- Начало Работ\***: Date field with value "1 июня 2015 г.".
  - Окончание Работ\***: Date field with value "1 апреля 2009 г.".
- Количество**: Text field with value "-3".
- Цена (тыс. руб.)**: Text field with value "86,0000".
- Цена За Единицу (тыс. руб.)**: Text field with value "2,1500".
- Модель Цены**: Empty text field.
- Объем Собственной Работы (%)**: Text field with value "76".
- Объем К/А (%)**: Text field with value "34".
- Плановая Трудоемкость (%)**: Text field with value "84".
- Фактическая Трудоемкость (%)**: Text field with value "73".
- Страница**: Text field with value "qqq".
- Ведущий**: Text field with value "Вольнов А.И.".
  - Примечание**: Large empty text area.

At the bottom of the window, there are two buttons: "Сохранить" (Save) and "Удалить" (Delete).

Рисунок 10. Валидация данных.

В связи с большим объёмом формы «Договор» проверка корректности каждой вкладки является отдельной функцией (Рисунок 10):

- ValidateDoc – проверяет правильность заполнения полей вкладки «Договор». При создании нового договора, поле «Номер» активно пользователю и требует обязательного заполнения. Поле проверяется функцией «Validator.UInt» и, если введенное число соответствует целому беззнаковому типу, то это число проверяется на уникальность атрибута «Номер» остальных записей во избежание перезаписи существующих данных. Поле «Генеральный заказчик» реализовано выпадающим списком и по умолчанию оно пустое, а поскольку оно не может быть пустым, то от

пользователя требуется, чтобы был выбран какой-либо из вариантов списка. «Начало работ» и «Окончание работ» так же должны быть непустыми, однако дата окончания работ должна быть задана большей либо равной дате начала работ. Остальные поля, за исключением иностранных заказчиков и исполнителей договора, проверяются в соответствии с типом данных, указанным в БД, и могут быть не заполнены.

– `ValidateStages` – проверяет правильность заполнения пользовательских форм «Этап договора» и вложенных в неё «Учёт материальных ценностей» вкладки «Этапы». Поле «Номер» используется для упорядочивания вкладок, описывающих каждый этап. Это поле проверяется функцией «`Validator.UInt`» при изменении и, в случае когда номер введён корректно, порядок вкладки упорядочиваются. Поля дат выбираются аналогичным образом как во вкладке «Договор». Все остальные поля проверяются в соответствии с их типом данных. Поскольку каждый этап договора содержит в себе несколько объектов из таблицы «Учёт материальных ценностей», то в каждой вкладке, описывающей определённый этап, выполняется валидация полей вложенной таблицы.

– `ValidateArchive` – отвечает за правильность заполнения полей вкладки «Архив». Каждый файл представляет собой группу элементов, в которой пользователь может изменять название файла, поэтому система проверяет отсутствие в этих строках запрещённых символов.

У полей, не прошедших валидацию фон их названия окрашивается в красный цвет, показывая пользователю, что данные введены некорректно. Так же, вкладка, содержащая поля с неверными данными, меняет цвет, что упрощает их поиск.

В случае успешной валидации, договор сохраняется в несколько этапов:

1. Составляется SQL запрос, сохранения данных для таблицы «Договор», причём если это новый договор, то используется фраза «INSERT INTO», а при редактировании «UPDATE»;
2. Затем происходит сохранение данных списков «Иностранный заказчик» и «Исполнитель договора», заданных отношением «многие ко многим», используя функции «SQL.InsertManyToMany» и «SQL.DeleteManyToMany», описанными ранее;
3. Для сохранения файлов договора был составлен специальный «MERGE» запрос, который автоматически добавляет или обновляет запись, в зависимости от её наличия в БД. Сначала последовательно проверяются пользовательские формы файлов, те, которые помечены для удаления и их идентификатор не пуст, добавляются в список, для последующего удаления из БД. Формируется строка, содержащая два запроса: один для добавления файлов, а второй для их удаления.
4. Этапы договора сохраняются в два действия. На первом сохраняются атрибуты всех этапов, используя «MERGE» запрос и удаляются все этапы, которые не были перечислены, но принадлежат этому номеру договора. Вторым действием является сохранение объектов таблицы «Учёт материальных ценностей», которые были учтены при первом проходе по списку этапов договора.

Кнопка «Удалить» видима и активна только при редактировании договора. При нажатии на неё, пользователю показывается диалоговое окно с текстом «Вы действительно хотите удалить договор?» с вариантами, да и нет. При подтверждении удаления все данные о договоре и связанные с ним смежные таблицы удаляются из БД.



### 6.1.5 Список юридических лиц

В этом окне пользователю предоставлен список юридических лиц системы для просмотра информации по ним и редактирования их характеристик (Рисунок 11). Это окно содержит:

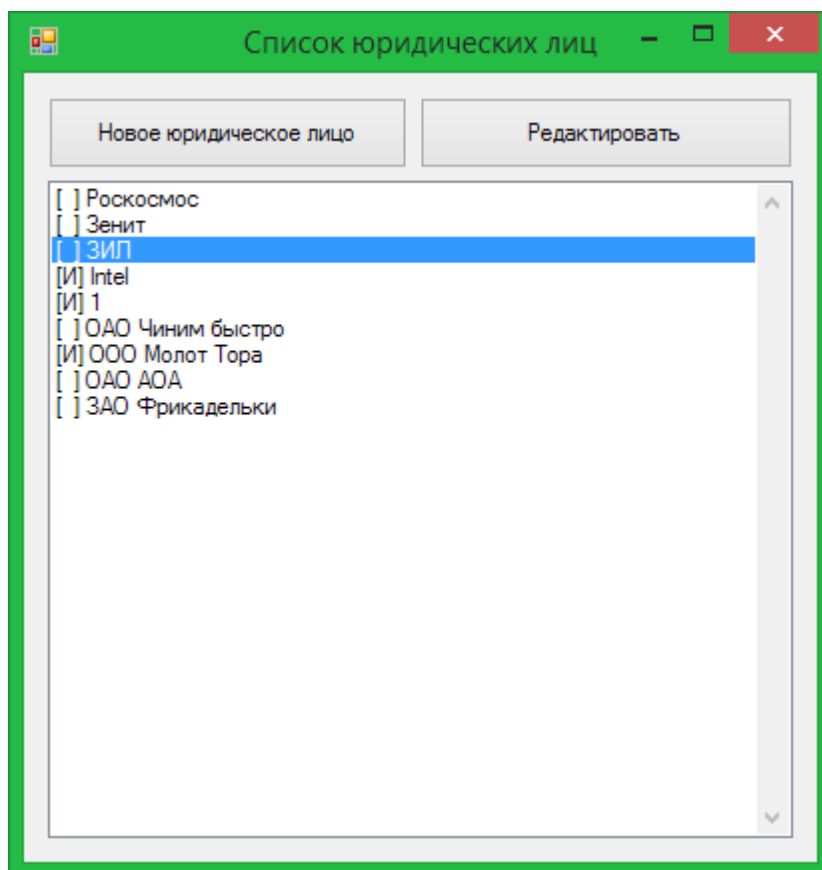


Рисунок 11. Список юридических лиц.

- кнопку «Новое юридическое лицо», при нажатии на которую, открывается форма создания юридического лица;
- кнопку «Редактировать», при нажатии на которую открывается окно редактирования выбранного юридического лица из списка;
- список, содержащий названия юридических лиц. В квадратных скобках перед названиями иностранные юридические лица имеют букву «И». При двойном нажатии на элемент списка открывается окно редактирования выделенного юридического лица.

### 6.1.6 Редактирование юридического лица

Окно редактирования юридического лица имеет элементы управления, позволяющие пользователю вводить значения атрибутов не только для таблицы «Юридическое лицо», а так же атрибуты таблицы «Экономические показатели», связанной отношением «один ко многим» (Рисунок 12).

Экономические показатели	
Год*	2013
Среднемесячный РОТ (тыс. руб.)	124
Социальное страхование (%)	13
Накладные расходы (%)	37
Рентабельность (%)	79
Лимит трудоёмкости (ч/мес.)	120

Рисунок 12. Редактирование юридического лица.

Пользователю доступны следующие элементы управления:

- текстовое поле, характеризующее название;
- флажок «Иностранный», может быть отмечен пользователем;
- элемент управления вкладками, содержащий атрибуты таблицы «Экономические показатели». Каждая вкладка имеет обязательный атрибут «Год», при изменении которого, текст вкладки меняется на это число, а так же происходит сортировка вкладок;
- кнопка «Сохранить», которая записывает данные в БД. Перед сохранением все поля проверяются на корректность введенных данных методами, содержащиеся в классе «Validator»;
- кнопка «Удалить», удаляющая текущее юридическое из БД, а так же все его экономические показатели.

## 7 Заключение

Совершенствование организационной деятельности является актуальной задачей, решение которой выступает важным условием ускорения принятия решений и планирования. Поскольку ведение бумажного делопроизводства в наше время является неэффективным, всё больше организаций начинают использовать автоматизированные системы.

Благодаря комплексу технических и программных средств, была создана автоматизированная система учёта договоров, предназначенная для автоматизации ряда процессов.

В результате использования системы электронного хранения договоров в ОАО «Российские космические системы» значительно снизятся временные затраты, повысится производительность труда и оперативность руководящего состава подразделения. Система позволит упростить ведение документации и организации эффективных экономических расчётов, ускорить поиск и составление договоров.

Использование системы автоматизированного электронного хранения договоров организации обладает следующими плюсами:

1. исключение доставки и хранения бумажных документов;
2. минимизация влияния человеческого фактора при обработке документов;
3. сокращение времени составления отчётов;
4. повышение достоверности, оперативности и надёжности.

## 8 Список литературы и интернет ресурсов

1. Сайт ОАО «РОССИЙСКИЕ КОСМИЧЕСКИЕ СИСТЕМЫ. [В Интернете] <http://www.spacecorp.ru>.
2. Преимущества использования систем электронного документооборота. [В Интернете] <http://lektsii.net>.
3. Системы электронного документооборота (СЭД) «ДЕЛО». [В Интернете] [http://www.eos.ru/eos\\_products/eos\\_delo](http://www.eos.ru/eos_products/eos_delo).
4. СЭД Docsvision 5. [В Интернете] <http://www.docsvision.com>.
5. 1С:Документооборот 8. [В Интернете] <http://v8.1c.ru/doc8>.
6. MSDN — сеть разработчиков Майкрософт. [В Интернете] <https://msdn.microsoft.com>.
7. Microsoft SQL Server. [В Интернете] <http://datasql.ru>.
8. Википедия — свободная энциклопедия. [В Интернете] <https://ru.wikipedia.org>.
9. ГОСТ 34.601-90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания». [В Интернете] <http://dit.mos.ru>.
10. «Хабрахабр» — ресурс для IT-специалистов. [В Интернете] <http://habrahabr.ru>.

## 9 Приложения

Пример MERGE запроса для сохранения набора данных таблицы «Этап договора», принадлежащие к определённому договору.

```
MERGE INTO [Этап договора]
  USING (
    VALUES (4, 1, 1, '02.06.15', '18.06.15', 2, 31, NULL, NULL,
16, 24, 64, 61, NULL, 4267, '2143k'),
    VALUES (-1, 1, 2, '19.06.15', '03.07.15', 1, 12, NULL,
NULL, 8, 12, 76, 74, NULL, 4284, '2151k'),
  ) AS source ([Id], [Договор_id], [Номер], [Начало работ],
[Окончание работ], [Количество], [Цена], [Модель цены], [Заключение],
[Аванс], [Расчёт],[Плановая трудоёмкость], [Фактическая трудоёмкость],
[Текущее состояние], [Номер акта], [Номер удостоверения])
  ON [Этап договора].[Id] = source.[Id]
WHEN MATCHED THEN
  UPDATE SET
    [Номер] = source.[Номер],
    [Начало работ] = source.[Начало работ],
    [Окончание работ] = source.[Окончание работ],
    [Количество] = source.[Количество],
    [Цена] = source.[Цена],
    [Модель цены] = source.[Модель цены],
    [Заключение] = source.[Заключение],
    [Аванс] = source.[Аванс],
    [Расчёт] = source.[Расчёт],
    [Плановая трудоёмкость] = source.[Плановая трудоёмкость],
    [Фактическая трудоёмкость] = source.[Фактическая трудоёмкость],
    [Текущее состояние] = source.[Текущее состояние],
    [Номер акта] = source.[Номер акта],
    [Номер удостоверения] = source.[Номер удостоверения]
WHEN NOT MATCHED THEN
  INSERT ([Договор_id], [Номер], [Начало работ], [Окончание работ],
[Количество], [Цена], [Модель цены], [Заключение], [Аванс], [Расчёт],
[Плановая трудоёмкость], [Фактическая трудоёмкость],
[Текущее состояние], [Номер акта], [Номер удостоверения])
    VALUES ([Договор_id], [Номер], [Начало работ],
[Окончание работ], [Количество], [Цена], [Модель цены], [Заключение],
[Аванс], [Расчёт],[Плановая трудоёмкость], [Фактическая трудоёмкость],
[Текущее состояние], [Номер акта], [Номер удостоверения]);
```

Функция сбора данных для составления свободного отчёта и последующая их загрузка в элемент управления данными при нажатии кнопки.

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        string args = "[Номер],";
        bool joingen = false;
        foreach (TreeNode node in treeView1.Nodes)
        {
            if (node.Checked)
            {
                switch (node.Name) {
                    case "Генеральный заказчик_id":
                        joingen = true;
                        args += "gen.[Название]as[Генеральный заказчик],";
                        break;
                    default:
                        args += "[" + node.Name + "],";
                        break;
                }
            }
        }
        string selectCommand = "SELECT " + args.Remove(args.Length - 1)
+ " FROM [Договор] AS d";

        if (joingen)
            selectCommand += " LEFT OUTER JOIN [Юридическое лицо] AS gen
ON d.[Генеральный заказчик_id]=gen.Id";

        SqlDataAdapter dataAdapter = new SqlDataAdapter(selectCommand,
Global.ConnectionString);

        SqlCommandBuilder commandBuilder = new
SqlCommandBuilder(dataAdapter);

        DataTable table = new DataTable();
        dataAdapter.Fill(table);
        dataGridView1.DataSource = table;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```