МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
"ПОЛЕССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"

Кафедра высшей математики
и информационных технологий

Допущено к защите
Заведующий кафедрой
_____ Т.М.Кадильникова
_____ 2018г.

ДИПЛОМНЫЙ ПРОЕКТ
на тему:
"Автоматизация деятельности отдела маркетинга организации, на примере филиала "Автомобильный парк № 8" ОАО "Брестоблавтотранс"

Студент:
информационные системы
и технологии финансово-
кредитной системы, 4-курс,
14 ИТ-1

Барнюк Мария Геннадьевна
_____2018г.

Научный руководитель
к.т.н., доцент

Вишняков Юрий Михайлович
_____2018г.

Пинск 2018

РЕФЕРАТ

Дипломный проект: 78с., 4 таблиц, 56 рисунков, 30 источников.

ИНФОРМАЦИОННАЯ СИСТЕМА, МАРКЕТИНГ, ТОРГОВО-ЗАКУПОЧНАЯ ДЕЯТЕЛЬНОСТЬ, ПРОГРАММНЫЙ ПРОДУКТ, БАЗА ДАННЫХ, УЧЕТ И АНАЛИЗ, ОБЪЕКТЫ, КЛАССЫ, ПРИЛОЖЕНИЯ, СПРАВОЧНИКИ, ЗАКАЗЫ, ДОКУМЕНТЫ, МЕТОДЫ.

Объектом исследования является компания по предоставлению услуг в сфере грузо-пассажирских перевозок, ремонта и обслуживания транспортных средств и торгово-закупочной деятельности.

Предметом исследования является торгово-закупочная деятельность предприятия.

Цель дипломного проекта: автоматизация документооборота, учета и анализа ведения торгово-закупочной деятельности предприятия.

Методы исследования: в процессе работы использовались методы систематизации и логического обобщения, наблюдение, анализ материалов, тестирование.

Задачи исследования:

1.Проанализировать предметную область и выявить объект автоматизации.

2. Построить информационную модель предметной области и предложить структуру базы данных.

3. Обосновать выбор средств проектирования и последующей разработки информационной системы.

4. Сделать технико-экономическое обоснование проекта.

5. Разработать рекомендации по охране труда на рабочем месте.

Область возможного практического применения: Филиал «Автомобильный парк №8 г. Береза» ОАО «Брестоблавтотранс».

Помимо прямого экономического эффекта, рассчитанного в данном проекте, предприятие получит косвенный эффект в виде снижения рутинной нагрузки на менеджера отдела продаж, и увеличении дополнительного времени для активных продаж и продвижения маркетинговых акций компании.

Автор работы подтверждает, что приведенный в дипломной работе материал правильно и объективно отражает состояние исследуемого процесса, а все заимствованные из литературных и других источников теоретические, методические положения и концепции сопровождаются ссылками на их авторов.

_____
подпись студента

ABSTRACT

Thesis project: 78 pages, 4 tables, 56 figures, 30 sources.

INFORMATION SYSTEM, MARKETING, TRADE AND PROCUREMENT ACTIVITY, SOFTWARE PRODUCT, DATABASE, ACCOUNTING AND ANALYSIS, OBJECTS, CLASSES, APPLICATIONS, HANDBOOKS, ORDERS, DOCUMENTS AND METHODS.

The research object is a company providing services in the field of cargo and passenger transportations, repair and maintenance of vehicles and trade-purchasing activity.

The subject of research is trade and procurement activities of the company.

The purpose of the graduation project: automation of documents circulation, accounting and analysis of trade and procurement activities of the enterprise.

Research methods: in the process, we used the methods of systematization and logical generalization, observation, analysis of materials, testing.

Objectives of the study:

1.To perform the subject area and to identify the automation object.

2. To build an information model of the application domain and propose a structure for the database.

3. Justify the choice of design tools and the subsequent development of an information system.

4. To do a feasibility study of the project.

5. To develop recommendations for occupational safety in the workplace.

The scope of possible practical application: branch of "Car park № 8", JSC "Brestoblavtotrans".

In addition to the direct economic effect calculated in this project, the entity will gain indirect effect in the form of decrease in routine load of the manager of sales office, and increase in extra time for active sales and promotion of marketing shares of the company.

The author of work confirms that the material given in the thesis correctly and objectively reflects a condition of the researched process, and all theoretical, methodical provisions and concepts borrowed from literary and other sources are followed by links to their authors.

_____

signature of the studenit

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ.....5
ВВЕДЕНИЕ.....6
ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННЫХ ПРОДУКТОВ МАРКЕТИНГА.....8
1.1 Маркетинговая информационная система.....8
1.2 Обзор программных решений, предназначенных для автоматизации маркетинговой деятельности.....14
1.3 Анализ предметной области.....19
ГЛАВА 2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ИНФОРМАЦИОННОЙ СИСТЕМЫ.....22
2.1. ОСНОВНЫЕ ТРЕБОВАНИЯ К ИНФОРМАЦИОННОЙ СИСТЕМЕ.....22
2.2 Разработка автоматизированной информационной системы.....23
2.2.1 Концептуальная модель системы учета.....23
2.2.2 Проектирование базы данных.....25
2.2.2.1 Построение концептуальной модели базы данных.....25
2.2.2.2 Построение логической модели базы данных.....27
2.2.3 Архитектура автоматизированной информационной системы.....34
2.2.4 Проектирование пользовательского интерфейса.....34
2.2.5 Проектирование классов.....36
ГЛАВА 3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ.....39
3.1 Средства разработки.....39
3.1.1 Среда разработки.....39
3.1.2 Система управления базами данных.....40
3.2Аномалии и защитное программирование.....41
3.3 Тестирование и отладка.....41
3.4 Анализ результатов решения.....42
3.5 Руководство пользователя.....42
3.6 Выводы по главе.....56
ГЛАВА 4 ЭКОНОМИЧЕСКАЯ ЧАСТЬ.....57
ГЛАВА 5 ОХРАНА ТРУДА.....66
ЗАКЛЮЧЕНИЕ.....76
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....77
ПРИЛОЖЕНИЕ А Код класс документов 79

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

АИС - Автоматизированная информационная система
ОС - Операционная система
БД - База данных
ПО - Программное обеспечение
СУБД - Система управления базами данных
СМИ - Система маркетинговой информации
CRM - Customer Relationship Management
КИС - Корпоративная информационная система
MVC - Model view controller
SQL - Structured query language
ЭИС - Эффективная информационная система

ВВЕДЕНИЕ

Основная цель любого предприятия, работающего в условиях рынка – получение прибыли. Как максимизировать прибыль и что для этого следует предпринять – это краеугольные вопросы всякого бизнеса.

Маркетинговая деятельность малого предприятия в значительной степени отличается от аналогичной деятельности средних и крупных предприятий. В малом бизнесе нет возможности образовывать специальные отделы маркетинга, в функции которых входят такие прерогативы, как исследование потребителей, конкурентов и рынков, выработка стратегий по освоению новых рыночных ниш, формирование рекламных кампаний, совершенствование сбытовой политики. Режим повседневной экономии и локальный характер рынков малого бизнеса заставляет руководителей предприятия брать маркетинговые функции на себя и распределять их между отделами. Использование маркетинговой информации становится необходимым условием повышения гибкости и эффективности системы управления предприятием.

Каким бы опытным ни был бы руководитель, как бы хорошо он ни знал свою сферу деятельности, тем не менее, во вкусах клиентов и положении конкурентов постоянно происходят изменения, а это требует систематического обновления и изучения информации.

В бизнесе и маркетинге, как ни в какой другой сфере, определяющими являются проблемы, связанные со сбором, хранением, поиском, переработкой, преобразованием, распространением и использованием информации. Маркетинг – один из наиболее интенсивно развивающихся секторов приложения информационных технологий, поскольку автоматизация информационных процессов в этой области не должна опираться на интуицию, суждения руководителей и опыт прошлого, а необходимо постоянно получать адекватную информацию до и после принятия решений.

Таким образом, маркетинговые исследования при управлении сбытом и закупкой товара являются той основой, на которой разрабатываются и принимаются все решения.

Наличие информационной системы для учета не только значительно облегчает его ведение, но и повышает эффективность труда в целом, позволяя затрачивать на данные операции меньше рабочего времени и предоставляя возможность быстрого получения статистических данных для оптимизации процесса управления прихода и расхода товаров. Если же информационная система предлагает функционал анализа торгово-закупочной деятельности, то это многократно увеличивает эффективность ее применения. Поэтому темой

дипломного проекта выбрана разработка программного приложения «Автоматизация деятельности отдела маркетинга организации».

Внедрение информационной системы состоит из нескольких стадий. Наиболее существенной и влияющим на итоговый результат являются стадии разработка бизнес-модели и проектирование информационной системы, так как именно на этих этапах формируется основа программного продукта.

Объектом исследования дипломного проекта является компания по предоставлению услуг в сфере грузо-пассажирских перевозок, ремонта и обслуживания транспортных средств и торгово-закупочной деятельности.

Предметом исследования дипломного проекта является торгово-закупочная деятельность предприятия.

Цель дипломной работы: автоматизация документооборота, учета и анализа ведения торгово-закупочной деятельности предприятия.

Автоматизированная система должна обеспечивать:

− работу с входными данными (ввод информации о клиентах, поставщиках и расходных материалах, а также оформление приходных и расходных операций);

− получение информации об актуальных складских остатках;

− формирование статистики по заказчикам и клиентам;

− анализ выполнения заказов и формирование количественного состава материалов к закупке;

Хранение всей информации осуществляется в электронной базе данных, что позволит структурировать информацию, появится возможность быстрого поиска необходимых данных.

Для достижения поставленной цели нужно решить следующие задачи:

1. Проанализировать предметную область и выявить объект автоматизации.

2. Построить информационную модель предметной области и предложить структуру базы данных.

3. Обосновать выбор средств проектирования и последующей разработки информационной системы.

4. Сделать технико-экономическое обоснование проекта.

5. Разработать рекомендации по охране труда на рабочем месте.

Актуальность разработки информационной системы связана с необходимостью систематизации договорной деятельности и анализа показателей предприятия для эффективного руководства, принятия управленческих решений по его дальнейшему развитию и повышения эффективности производства организации в целом, а также с возрастанием объемов информации в информационной системе организации, потребности в ускорении ее переработки.

ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННЫХ ПРОДУКТОВ МАРКЕТИНГА

1.1 Маркетинговая информационная система

Чтобы функционировать на должном уровне в системе маркетинга требуется информация. Информация – это сообщение конкретных данных о каком-либо событии или чей-либо деятельности [2, с.43].

Классификация маркетинговой информации:

- историческая – формирует представление об условиях и результатах деятельности предприятия в предшествующем периоде;
- текущая – отражает оперативное состояние бизнеса;
- прогнозная – представляет результаты вероятностной оценки позиций предприятия в обозримой перспективе;
- плановая - применяется в ходе разработок и принятия решения о целях, стратегии и программах маркетинга;
- первичная – формируется непосредственно в процессе проведения специальных исследований (опросов, наблюдений, тестирования), направленных на решение конкретной проблемы;
- вторичная – представляет собой данные, получаемые из бухгалтерской и статистической отчетности в специальных публикациях, справочниках, систематических и аналитических обобщений. Источники вторичной информации подразделяются на внутренние и внешние.

Носители вторичной информации чрезвычайно разнообразны, содержательны и доступны. А доступ к ним возможен при относительно низких затратах и в сжатые сроки. Кроме этого в ряде случаев ее бывает достаточно для роста целей конкретного исследования и принятия соответствующих маркетинговых решений. Поэтому без тщательного поиска и анализа вторичных данных, к сбору первичной информации приступать не рекомендуется.

Подсистемы СМИ.

Сбор информации о рынке и внешней среде должен вестись непрерывно и системно. Для этой цели на предприятии в рамках отдела маркетинга должна быть создана система маркетинговой информации. Система маркетинговой информации (СМИ) – это организованный способ сбора и анализа данных, с целью предоставления менеджерам по маркетингу информации, необходимой для принятия решений [12, с 85].

Система маркетинговой информации.

1.Система внутренней отчетности дает возможность аккумуляции и поиска необходимых сведений внутри предприятия. Такая информация возникает в результате деятельности предприятия и постоянно меняется вместе с ней.

Источники:

- статистическая и бухгалтерская отчетность;
- внутренняя статистика;
- материалы ранее проведенных исследований;
- данные заявок, заказов, договоров;
- акты ревизий, проверок;
- различного рода справки и отчеты;
- оперативная и текущая, производственная и научно-техническая информация;
- деловая переписка.

Система должна быть направлена на полное отражение текущей деятельности и выдачу оперативных сведений, характеризующих состояние дел предприятия.

Выполнение такой задачи требует наличие автоматизированных систем сбора и представления данных, а также внедрение новейших информационных технологий и информационных сетей. Данная информация облегчает руководителям и специалистам выработку и принятие маркетинговых решений, но обычно не используется в отрыве от других видов источников и информации, поскольку не содержит многих необходимых сведений.

2. Система сбора текущей внешней маркетинговой информации.

Она ориентирована на источники и методические приемы, с помощью которых можно получить информацию о событиях и ситуациях, складывающихся во внешней маркетинговой среде.

Источники:

Общей маркетинговой информации:

- книги общей экономической ориентации;
- периодические печатные издания общей экономической ориентации;
- статистические издания;
- справочники;
- телевидение, радио;
- рекламная деятельность массового характера;
- законодательные и нормативные акты, а также другие публикации;
- выставки, совещания, конференции, презентации, дни открытых дверей;
- выступления государственных, политических и общественных деятелей.

Узко профильной маркетинговой информации:

- публикуемые финансовые и бухгалтерские отчеты предприятий;

- отчеты, интервью руководителей и специалистов предприятия;
- узкоспециализированные периодические печатные издания;
- книги, пособия, учебники и справочники по маркетингу;
- печатная реклама предприятий (каталоги, буклеты, проспекты);
- специальные выставки и ярмарки;
- посещение предприятий;
- сведения экономического характера, распространяемые специальными фирмами в форме печатной продукции;
- коммерческие базы и базы данных;
- каналы личной коммуникации (личные контакты с потребителями, обмен информацией с другими фирмами).

3. Система анализа маркетинговой информации основное ее значение состоит в обработке имеющихся данных, что позволяет сделать соответствующие выводы, в рамках изучаемой проблемы и наметить основные пути ее разрешения [12, с.92].

Данная система представляет собой набор современных экономическо-математических и экономико-статистических методов обработки информации.

Методы:

- регрессивный анализ – статистический метод анализа данных для определения зависимости одной переменной от другой или нескольких независимых переменных;
- дисперсионный метод – используется для подтверждения или опровержения влияния изучаемого фактора на определенный экономический показатель;
- вариационный анализ – предназначен для проверки, существенно ли влияние изменения независимых переменных на зависимые;
- дискриминантный анализ – с помощью него можно разделить заранее заданные группы объектов через комбинацию многих независимых переменных и таким образом объяснить различие между группами, кроме того, метод дает возможность отнести новый объект к какой-либо группе на основе его характеристик;
- факторный анализ - используется для исследования взаимосвязи между переменными с целью уменьшения числа влияющих факторов до наиболее существенных;
- кластерный анализ позволяет увеличить переменные или объекты в группы (кластеры) таким образом, чтобы различие между объектами, составляющее 1 кластер были бы меньше их отличий от других кластеров (сегментирование рынка);
- многомерное шкалирование;

- сегментический дифференциал- наряду с анализом информации широко используются и методы прогнозирования, которые способствуют значительному облегчению труда руководителей при выработке стратегий.

Количественные методы прогноза:

- экстраполяции тренда – проекция временного ряда в будущее;

- индикативный – прогнозы на основе индикатора – оценка хода развития процесса на базе одной или многих предпосылок.

Качественные методы прогноза:

- дельфи-метод – форма опроса экспертов, при которой их ответы собираются в течении нескольких туров и через ознакомление с промежуточными результатами получают групповую оценку интересующего процесса (ответы, как правило анонимные).

- сценарии – предсказание роста и будущее состояние факторов, влияющих на предприятие и определение возможных действий руководителей.

Для прогнозирования применяется также интуитивно-творческие методы (мозговая атака). Использование маркетинговой информации становится необходимым условием повышения гибкости и эффективности системы управления предприятием.

Маркетинговая информационная система представляет собой систему информационного обслуживания работников предприятия, занятых в его службах, и обязана выполнять некоторые функции по хранению, накоплению, обработке, и передаче информации.

Сама по себе любая экономическая информационная система, в том числе и маркетинговая информационная система - есть совокупность внешних и внутренних потоков а также наличие прямой и обратной информационной связи данного экономического объекта, средств, методов, специалистов, участвующих в затронутом процессе обработки объемов информации и дальнейшей выработке ключевых управленческих решений.

Сам процесс автоматизации в общем случае представляет из себя комплекс мер и действий организационного, технического и экономического характера, позволяющего снизить степень непосредственного участия или в корне исключить участие человека в процессе осуществления той или иной управленческой функции производственного процесса.

На сегодняшний день можно с уверенностью констатировать факт наличия многообразия программных продуктов, имеющихся на рынке информационно-технических решений и направленных на осуществление задач с учетно-статистическим характером.

С такими задачами информационно и логически тесно связаны маркетинговые цели, составляющие основу внутренней инфраструктуры управления на современном этапе производственно- сбытовой деятельности.

Если учитывать прогрессирующий рост конкуренции, то можно предугадать, что решение таких задач станет все актуальнее. Помимо того, единственно правильной концепцией управления на рынке является маркетинг, обязанности которого исследование а также организация и планирование.

К сожалению полностью готовых и полных систем автоматизированного маркетинга на нашем Белорусском рынке информационных услуг найти практически невозможно.

Автоматизированная система маркетинга обычно рассматривается как четко упорядоченная (имеется в виду последовательности функционирования и назначение) некая совокупность различных программных модулей и общих информационных пакетов данных, служащих хорошим подспорьем в ходе принятия важных маркетинговых решений обязанным управляющим персоналом компании или организации. От типа организации маркетинговой работы на предприятии, структуры маркетингового управления, распределения задач и непосредственных обязанностей персонала в этой сфере зависит то, какое содержание, направление, методов и приемов автоматизации будет установлено. Исходя из этого маркетинговая служба периодически поставляет информацию аналитического и информационного характера другим подразделениям, на которой основываются принятия решений касательно технологической, ассортиментной, финансовой и научно-технической политики предприятия. В основе каждого принятого маркетингового решения лежат особые требования рынка. Вполне понятно, что при этом многократно может возрасти роль небольших статистических подпрограмм.

Такие подпрограммы носят вполне универсальный характер и обычно успешно используются вне зависимости от типа и назначения маркетинга, будь то маркетинг выпускаемой продукции инженерного или производственного назначения, либо товар потребительской сферы. Информационная система маркетинга есть часть общей системы управления микроэкономическим объектом и обязана осуществлять поэтапное проведение необходимых маркетинговых исследований, будь то сбор и регистрация различных данных или анализ и выработка рекомендационных.

Автоматизация маркетинга – является искусством не в меньшей степени, чем им является сам маркетинг [14, с.143]. Стандартизированный подход сдесь фактически невозможен. Реально выделить только общий состав задач автоматизированной системы маркетинга, используя «маркетинг» как определение самого понятия. Исходя из этого здесь и перечень основных задач:

ценообразование, прогноз покупательского спроса, оптимизация ассортимента, анализ внешней среды, выработка рациональных вариантов их продвижения и доставки. Создание автоматизированной системы, так называемого сквозного маркет - моделирования, может быть выгодно многим компаниям, так как можно избегнуть напрасных расходов на производство бесполезной продукции, определить на товары и услуги максимально эффективные цены, а также упорядочить и построить наиболее рациональную рыночную деятельность данного хозяйствующего субъекта.

Маркетологам все чаще неизбежно приходится постоянно учитывать и анализировать огромные массивы разнородных и разнообразных данных, таких как характеристики конъюнктуры рынка (товарная номенклатур а поставщиков, состоящей, к примеру, из семизначного числа позиций и разнообразнейшей (а зачастую и непредсказуемой) ценовой политикой, в том числе сами товары и услуги с растущим числом характеристик, отобранные целевые группы постоянных потребителей с их изысканными вкусовыми предпочтениями, теоретической покупательской способностью и все непрерывно увеличивающимися претензиями как к качеству товара так и качеству сервиса, ну и конкуренты с их максимально коварными замыслами), а также результативность данной маркетинговой политики, и степень необходимости планирования и разработки собственной деятельности (часто включающей не только чисто маркетинговые, но еще и разные рекламные мероприятия) - и делается все это в условиях жёсткого дефицита нужной информации при переизбытке «информационного мусора».

Естественно, эта проблема информатизации деятельности любого маркетолога, а вернее выбора или разработки на заказ специализированного программного продукта, является лишь «подпроблемой» в аспекте информатизации организации в целом. Но, как показывают и растущая активность обсуждения на профильных интернет форумах и экспертные оценки, проблема выбора программного обеспечения, решающего поставленные маркетинговые задачи, является все более актуальным вопросом. Некой «палочкой-выручалочкой» для любого маркетолога в таких условиях обязано стать специальное программное обеспечение, способное удовлетворить критериям надежности, функциональности и цены. Так что же может предложить нам белорусский рынок специализированного программного обеспечения?

По сути, классифицировать все виды программ, относящихся к маркетингу, не так просто, но в первую очередь выделяются нижеследующие категории. Узконаправленные маркетинговые модули, которые входят в состав крупных программных комплексов: «Парус», «Галактика», и решений категории CRM. Аналитические программы, способные удовлетворить нужды маркетинга:

SPSS, Forecast Expert, Statistika, Бизнес-Прогноз, ДА-система. Специализированные программные модули, созданные для решения отдельных маркетинговых задач, задач предприятий или выделения сегментов рынка. Таким программам нет числа, поскольку просты в разработке и распространяются на бесплатной основе или за символическую плату.

Универсальные маркетинговые программы с расширенным функционалом позволяют эффективно решать весь спектр маркетинговых задач организации и совместимы с некоторыми другими программами (бухгалтерский учет и финансы) в рамках корпоративной информационной системы (КИС) предприятия.

1.2 Обзор программных решений, предназначенных для автоматизации маркетинговой деятельности

Такие популярные программы как «ИНФИН-Бухгалтерия», «1С: Бухгалтерия», «Инфо-Бухгалтер» и другие знают практически все. Но эти продукты лишь учетные системы. Они существенно облегчают работ у бухгалтеров, но мало помогают их руководителям. С подачи специализированных консультантов на предприятиях начали внедряться управленческие, аналитические системы и даже самые простейшие из маркетинговых программ. К первым можно отнести такие системы как – «Галактика», «Парус-Корпорация», «NS-2000», «Тектон». Ко вторым «ИНЭК-Аналитик», «Альт-Инвест», «Audit Expert». К третьим – «Marketing – Expert», «БЭСТ- Маркетинг», «Marketing GEO». Вот анализ некоторых из них:

1.«БЭСТ-Маркетинг». Система «БЭСТ-Маркетинг» является по сути удобным и весьма эффективным инструментом, с помощью которого можно оценить позиции организации на рынке в условиях жесточайшей конкуренции. Данная система пригодна к использованию на предприятиях торговли, производства, и сферы услуг. Важным бесспорным достоинством этой программы является максимальная простота и отзывчивость терминологии интерфейса, рассчитанного на самого обыкновенного, незамысловатого пользователя без какого-либо образования в специализации «маркетинг»[27,с.68]. Ввод информации происходит в виде естественных и самоочевидных оценок по простому принципу «важно/второстепенно», «хуже/лучше». Качественные данные, введенные пользователем, преобразуются этой системой в количественные, что по сути позволяет пользователю проводить соответствующие расчеты. Среди методик, базируется на которых «БЭСТ-

Маркетинг» имеет место модель Розенберга, SWOT- анализ, метод 4Р и матрица Анзоффа. Возможности системы «БЭСТ – Маркетинг»:

Анализ рынка:

• определение перспективы в рыночных нишах;

• анализ на конкурентоспособность товара;

• анализ рекламы, различные рекомендации по способам повышения ее эффективности;

• рекомендации по способам стимулирования продаж. Анализ конкурентов:

• фактор присутствия на рынке конкурентов;

• сравнительный анализ товаров;

• сравнительный анализ по рекламным компаниям.

Финансовое планирование маркетинговой деятельности:

• прогнозирование продаж;

• планирование бюджета рекламы;

• расчет сводного бюджета;

• подготовка планируемого бюджета затрат на продвижение продукции и осуществления контроля за исполнением.

Благодаря «БЭСТ-Маркетинг» организация может оценить свои слабые и сильные стороны в рекламной и, в частности, маркетинговой политике и адекватно получить широкий комплекс рекомендаций по возможному укреплению своих позиций на рынке.

2. Marketing Expert. Marketing Expert - система особого стратегического планирования и ведения аудита маркетинга. Производит анализ прибыльности выбранных сегментов рынка и имеющихся каналов сбыта. Сводный анализ слабых и сильных сторон компании. Подробный анализ портфеля производимой продукции. Осуществляет анализ стратегий, разработку плана маркетинга [14, с.96].

Marketing Expert позволяет:

• создать актуальную модель деятельности организации на рынке, в том числе выделяя сопутствующие товары, сегменты, каналы сбыта и многие другие коммерческие характеристики;

• определить прибыльность и доходность организации, учитывая созданную модель;

• проанализировать группу продукции с использованием разнообразных матричных моделей: GE, BCG , ADL и отдельно создаваемых матриц;

• сделать анализ внешней среды, в том числе анализ потребителей, основных конкурентов и справедливую оценку благоприятности рынков;

• выполнить SWOT-анализ, выяснить, сильную и слабую сторону организации, перспективные возможности, используя имеющуюся методику;

• спрогнозировать объем продаж компании.

• выделить важные стратегические пути развития, учитывая целевые сегменты, товарный ассортимент и конкурентные преимущества компании;

• сделать и сопоставить различные стратегии, определить оптимальную;

• составить план маркетинговых мероприятий, направленных на реализацию стратегий и первоначально оценить возможные затраты на выполнение.

• получить оценки экспертов имеющейся и планируемой деятельности на выбранном рынке, составляя анкеты и производя с их помощью экспертные опросы;

• контролировать стратегию по части реализации, сравнивая фактические и плановые показатели.

• существенно сократить время на анализ и обработку данных, при этом используя для этого данные о сбыте, полученные из различных источников.

Система Marketing Expert основывается на классической теории маркетинга и включает потребности полевого маркетинга.

Система создана с использованием известных аналитических методик (Сегментный анализ, GAP-анализ, Portfolio – анализ, SWOT- анализ), и обязана решать ежедневные практические задачи маркетинга.

Marketing Expert - бесценный консультант любого руководителя предприятия, либо руководителя службы сбыта или коммерческого директора. Программа позволяет на практике освоить главные принципы и основные методы анализа и планирования в маркетинге.

3. «Касатка» «Касатка-prof» это современная, но проверенная технология быстрого принятия решений как стратегического, тактического так и оперативного уровней, предназначенная для организаций.

Здесь объединены самые новые разработки в маркетинговой области и области менеджмента, накопленный опыт мировых лидеров бизнеса и широкие возможности компьютерной техники в сфере современных технологий.

«Касатка» - это инструментарий, позволяющий эффективнее использовать время. Работая с «Касаткой», можно выделить два основных пункта – увеличение прибыли и снижение затрат. Программа неплохо структурирована с точки зрения как предметной задачи так и пользовательского интерфейса. Авторы очень хорошо определили уровни принятия решения и способы их реализации, добавив описание формализованных интерфейсов при переходе от задачи к задаче. Касатка может пригодиться на малых и средних предприятиях, впервые

подошедших к внедрению маркетинга. Для специалистов, профессионалов – она слаба в своих возможностях.

При работе с программными продуктами маркетинговой деятельности обнаружилось, что потенциал этих программ очень часто пропадает впустую. Такие программы столкнулись с недостаточной готовностью менеджмента к их пониманию и восприятию. Сложные компьютерные продукты рассчитаны на довольно опытных специалистов, и при том в них не хватает важного элемента – элемента обучения человека шаг за шагом.

Обучать не всякой ерунде, а принципиально новому способу мышления, приемам работы. Потому разработчики таких программных продуктов стремятся исправить эту ситуацию.

4. Системы CRM. Системы CRM (Customer Relationship Management – «управление отношениями с потребителями») инструментарий по оптимизации обслуживания клиентов и продаж.

CRM в основном ориентирован на постоянных клиентов: потому как известно, что постоянное увеличение доли таких покупателей на 5%, в среднем увеличивает объёмы продаж чуть более чем на 25%. Лучший способ увеличить долю постоянных клиентов - предоставить им наилучший сервис, наиболее полно и оперативно стараться учитывать их потребности, выстраивать отношения индивидуально с каждым клиентом. Вот на решение этих задач и была нацелена система CRM. Её основная работа строится на своевременном учёте и точном анализе заранее собранной информации о клиентах.

Основными функциями CRM являются:

• организация единой точки доступа ко всей информации имеющей отношение к клиенту;

• оперативное уведомление об изменениях в этой информации ответственных сотрудников и клиента;

• автоматическое формирование цен и других условий поставки для каждого клиента в зависимости от статуса клиента, его предпочтений, истории;

• формирование специальных предложений клиентам, реализация программ лояльности; предложение новых товаров и услуг;

• автоматизация прохождения заказа (резервирование товара на складе, включение его в план производства);

• автоматический контроль прохождения заказа и действий сотрудников по обслуживанию клиента;

•предоставление инструментов многопланового анализа и прогнозирования продаж;

• информационная поддержка клиента CRM позволяет проанализировать затраты на обслуживание различных клиентов и оптимизировать их, сравнить

экономический эффект от продажи различных товаров, вовремя выявить тенденции спроса и оптимизировать ассортимент.

В условиях ограниченности ресурсов реального предприятия CRM даёт возможность сосредоточиться на наиболее перспективных клиентах и наиболее прибыльных направлениях. Более тесное сотрудничество с потребителями, дополнительные сервисы "привязывают" их к компании, что увеличивает количество постоянных клиентов. В развивающейся экономике по мере насыщения рынка будет возрастать ценность CRM- системы, как инструмента борьбы за потребителя.

5.Marketing Analytic. Программный продукт Marketing Analytic компании «КУРС» осуществляет анализ маркетинговой деятельности, а также оперативное и стратегическое планирование маркетинга. Этот программный комплекс построен по модульному принципу и позволяет решать широкий спектр задач маркетинга.

Так, модуль cCommerce выполняет функции оперативной CRM-системы, его задачами являются автоматизация работы персонала отдела продаж с клиентами и сбор информации, необходимой для маркетингового анализа.

Модуль Analyzer предназначен для анализа данных, полученных из различных источников и накапливаемых в аналитическом хранилище, в качестве инструментария используются различные методы математической статистики и OLAP- технологии.

Главное назначение модуля Geo - анализ информации об объектах, имеющих пространственную привязку, с возможностью отображения результатов на географической карте.

С помощью модулей Portfolio и Predictor можно решать задачи стратегического маркетинга, например моделировать различные сценарии развития рынка. По сравнению с предыдущей - четвертой версией ПО, представленной два года назад на выставке SofTool, в новый вариант программного комплекса внесен ряд существенных изменений. Среди важнейших, по мнению разработчиков, модификаций модуля c-Commerce - возможность визуального представления персональных контактов менеджеров отдела продаж, реализация связи с учетными бухгалтерскими и складскими программами, более дружественный пользовательский интерфейс.

Заново создан модуль Portfolio, обеспечивающий более совершенные функции многокритериальной оценки и новые возможности визуализации сопоставляемых альтернатив.

Кроме того, разработан модуль Conjoint, предназначенный для оценки целесообразности вывода на рынок новой продукции и определения наиболее востребованного набора ее свойств. Как подчеркивают представители компании

КУРС, их цель состоит в том, чтобы сделать свой программный комплекс ядром информационной системы автоматизации маркетинга. По их утверждению, до сих пор не создано ни одной системы, автоматизирующей маркетинг в полной мере.

Маркетинг - один из наиболее интенсивно развивающихся секторов приложения информационных технологий, поскольку автоматизация информационных процессов в этой области в условиях интенсивного развития рыночных отношений является стратегическим фактором конкуренции.

ЭИС маркетинга является системой информационного обслуживания работников, занятых в его службах, и выполняет технологические функции по накоплению, хранению, передаче и обработке информации. Она складывается, формируется и действует в регламенте, определенном методами и управленческой структурой, принятой на конкретном экономическом объекте (предприятии, организации), реализует цели и задачи, стоящие перед ним. Автоматизированная система маркетинга может рассматриваться как упорядоченная совокупность программных модулей и информационных наборов данных, служащих подспорьем в ходе принятия маркетинговых решений управляющим персоналом фирмы или организации.

Успешная работа на рынке невозможна без продуманного маркетинга. Компаниям приходится все больше внимания уделять сбору и анализу значительных объемов количественной и качественной информации, полученной из разнородных источников.

Сделать эту работу эффективной можно лишь посредством ее автоматизации.

1.3 Анализ предметной области

Филиал «Автомобильный парк №8 г. Береза» ОАО «Брестоблавтотранс» - коммерческая организация, осуществляющая свою деятельность на рынке автотранспортных услуг с 1961 года.

В распоряжении филиала порядка сорока единиц подвижного состава - автобусы разной вместимости и уровня комфортности, грузовые автомобили разных моделей, модификации и грузоподъемности, диагностическая станция для прохождения государственного технического осмотра транспортных средств.

Филиал «Автомобильный парк №8 г. Береза ОАО "Брестоблавтотранс" оказывает следующие услуги:

1. Пассажирские перевозки;

2. Грузовые перевозки, в т.ч. международные перевозки грузов;
3. Государственный технический осмотр транспортных средств;
4. Технический осмотр транспорта перед выездом на линию;
5. Медицинское освидетельствование на допуск к работе;
6. Туризм, организация экскурсий;
7. Общественное питание, обслуживание торжеств;
8. Сдача в аренду неиспользуемых площадей;
9. Платные стоянки автомобилей.

Главной целью Филиала является хозяйственная деятельность, направленная на получение прибыли для удовлетворения социальных и экономических интересов трудового коллектива Филиала и Общества.

Предметом деятельности Филиала является обеспечение потребностей юридических и физических лиц в пассажирских, грузовых перевозках, а также проведение диагностики транспортных средств и сдачи в аренду имущества автопарка.

Недавно открыт магазин по продаже одежды, а в ближайшее время откроется станция по техническому обслуживанию и ремонту автомобилей МАЗ, в перспективе открытие кафе.

Для ведения этих видов деятельности потребуется вести учет движения товарно-материальных ценностей, сохранять данные о поставщиках и клиентах, отслеживать своевременность закупки товаров и цен на них у различных поставщиков, что позволить своевременно принимать нужные решения: у кого, когда, сколько и по какой цене закупать товар.

Все это сопровождается оформлением соответствующих документов. На предприятии для ведения учета и сохранения такого типа данных имеется некоторое программное обеспечение.

1С версии 7.7 с конфигурацией "Автотранспорт" (обработка путевых листов, отчеты по топливу, пробегу автомобилей и перевезенных грузов).

Недавно установлена 1С версии 8.2 с конфигурацией "Заработная плата и управление персоналом", а также система СМДО - делопроизводство, регистрация входящих и исходящих документов, слежение за сроками исполнения документов, а также Excel и Word.

Все эти системы не позволяют быстро и своевременно получать и просматривать необходимую информацию о состоянии исполнения принятых заказов, просмотра и оформления соответствующих документов, необходимых для ведения торгово-закупочной деятельности, а также получения информации необходимой для принятия решения на предмет принятия заказа от клиента.

На основании проведенного анализа и по просьбе предприятия, было принято решение о разработке универсального программного приложения,

которое позволило бы ускорить оформление соответствующих документов, необходимых при оформлении заказа на закупку и продажу различных товаров, сохранения данных с возможностью их просмотра в любой момент времени, а также получать информацию о закупках и продажах товаров за определенный период времени и отслеживать состояние выполнения заказов по продаже и закупке товаров, чтобы была возможность определить каких и сколько нужно на данный момент.

Предполагаемый порядок оформления сделки на продажу товара следующий: вводятся данные о контрагенте (клиенте), заключается договор, принимается заявка на товар от клиента, на основании заявки оформляется заказ, содержащий в себе всю информацию о заказываемых товарах (количество, цена, наименование, стоимость и т.д.).

Заказ является дополнением к договору:

- в зависимости от условий оплаты (например, если предоплата, то оформляется счет-фактура);

-выписка товарной накладной на отгрузку товара, по ней ведется учет выполнения заказа.

Предполагаемый порядок оформления сделки на закупку товара следующий:

- оформляем заявку и отправляет ее поставщику (заявка содержит в себе перечень и количество заказываемых товаров без цены);

- вводим данные поставщика товаров;

-вводим данные с полученного заказа или со счет-фактуры (если требуется предоплата), эти данные будут использованы при выборе поставщика товаров;

- после принятия решения вводим данные договора, сохраняем договор на поставку (закупку) товара от поставщика;

- при получении товара, вводим данные товарной накладной.

Вся информация о поставщиках товаров сохраняется в базе данных (БД), что поможет в дальнейшем при выборе поставщика товаров, будет производиться:

- оценка цены;

- условий оплаты (есть ли предоплата);

- своевременность поставки товаров (надежность поставщика);

- условий доставки (доставка или самовывоз).

При оформлении сделки по закупке и продаже товаров должна быть возможность ее оформления без заключения заказа, то есть по факту поставки на основании товарных накладных с заключение договора, если он еще не заключен.

ГЛАВА 2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1. Основные требования к информационной системе

Основными требованиями к ИС являются следующие:

- система должна обеспечивать защиту информационной базы данных от несанкционированного доступа;
- основная программная оболочка должна иметь интуитивно ясный дружественный интерфейс, понятное назначение функций и наглядный результат обработки информации и не должна требовать от пользователей специальной подготовки, не связанной с их профессиональными обязанностями;
- система должна иметь возможность наращивания в программной части.

Успешное функционирование ИС подразумевает накопление и анализ максимально большого объёма информации, что требует поступления данных в систему из как можно большего числа источников. Источники информации:

- справочники;
- документы.

Ввод и изменение данных должны быть доступны только специалистам, допущенным к работе с информацией.

Цели создания АИС:

- накопление, хранение, обработка и выдача достоверной и оперативной информации;
- улучшение качества контроля и учета обрабатываемой информации;
- повышение эффективности бизнес-процессов.

Перечень функций, подлежащих автоматизации:

– оформление и сохранение в базе данных договоров;

– ведение учета движения товарно-материальных средств на основании данных товарных накладных;

– ведение учета оформления заказов на закупку и продажу товаров и сохранение их в базе данных, с возможностью дальнейшего их просмотра ;

– ведение учета и оформления товарных накладных и сохранение их в базе данных, с возможностью дальнейшего их просмотра;

– ведение учета выполнения заказов на основании данных расходных товарных накладных и принятых заказов;

– ввод данных в справочник, содержащих информацию об контрагентах;

– ввод данных в справочники, содержащие информацию, позволяющую облегчить ввод данных.

Пользовательский интерфейс должен соответствовать определенным требованиям.

1. Эффективные интерфейсы должны быть очевидными и внушать своему пользователю чувство контроля. Необходимо, чтобы пользователь мог одним взглядом окинуть весь спектр своих возможностей, понять, как достичь своих целей и выполнить работу.

2.Эффективные интерфейсы не должны беспокоить пользователя внутренним взаимодействием с системой и должны обеспечить бережное и непрерывное сохранение работы, с предоставлением пользователю возможности отменять любые действия в любое время.

3. Эффективные приложения должны выполнять максимум работы, требуя при этом минимум информации от пользователя.

Требования к техническому обеспечению. Должны использоваться следующие технические средства: персональный компьютер (системный блок, монитор, клавиатура, мышь).

Характеристики:

- тактовая частота процессора не ниже 1500 МГц;
- 256 Мб оперативной памяти и выше;
- монитор от 17 дюймов;
-винчестер емкостью от 40 Гб.;
-операционная система Windows XP, Windows 7, Server 2008, .NET Framework 4.5.1;
- установленный и работающий Microsoft SQL Server 2008;
- лазерные принтеры типа LaserJet 1200 или МФУ.

2.2 Разработка автоматизированной информационной системы

2.2.1 Концептуальная модель системы учета

Внедрение автоматизированной информационной системы по учету ведения торгово-закупочной деятельности позволит снизить нагрузку на специалистов и повысить культуру обслуживания клиентов.

В настоящий момент информация, поступающая от подразделений предприятия, проходит ручную обработку – отсутствуют единые классификаторы и данные для отчетности представляют собой информацию, передаваемую в виде xls-файлов, а также в заполненном вручную виде.

Автоматизированная обработка информации позволит объединить все бизнес-процессы в единую технологическую схему, в которой будут присутствовать хранилища данных и единые классификаторы.

Таким образом, на этапе анализа выявлены задачи автоматизации:

- учет договоров;
- учет контрагентов;
- учет расхода и прихода товарно-материальных ценностей;
- учет заказов;
- формирование отчетности.

Входными информационными потоками в технологии по учету совершения сделки являются:

- информация о контрагентах;
- информация о принятых заказах;
- информация о продаже товара;
- информация о закупке товара.

Выходными информационными потоками являются:

- сводная информация о наличии товара на складе;
- сводная информация о товарах, которые еще нужно заказать для выполнения заказов;
- информация о ценах на товары у поставщиков.

Специалисты в рамках работы с автоматизированной системой будут работать как со справочной информацией, необходимой для автоматизированной системы, так и производить ввод документов, формировать отчетность.

Автоматизация данных задач позволит оптимизировать деятельность специалиста по учету ведения торгово-закупочной деятельности, исключить вероятность ошибок вследствие влияния человеческого фактора, позволит оптимизировать деятельность исследуемого предприятия.

На этапе анализа бизнес-процессов были определены информационные объекты:

- справочники:
- контрагенты;
- банковские реквизиты;
- контактные телефоны;
- товары:
- сведения о товаре;

- документы:
- договор: договора купли-продажи;
- заказы:
- сведения заказов;
- накладные:
- сведения накладных;
- счет фактуры:
- сведения счет-фактур.

2.2.2 Проектирование базы данных

Проектирование БД состоит из следующих основных этапов:

1. Концептуальное проектирование;
2. Логическое проектирование;
3. Физическое проектирование.

Концептуальное проектирование – это построение концептуальной модели предметной области, то есть информационной модели наиболее высокого уровня абстракции [25, с.296]. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных.

Логическое проектирование - создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных логическая модель - набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

Физическое проектирование — создание схемы базы данных для конкретной СУБД.

2.2.2.1 Построение концептуальной модели базы данных

Представим концептуальную модель базы данных (рисунок 2.1)



Рисунок 2.1 – Концептуальная модель БД

На рисунке 2.2 представим объекты, которые не имеют связей между собой и не требуют данных от других объектов.

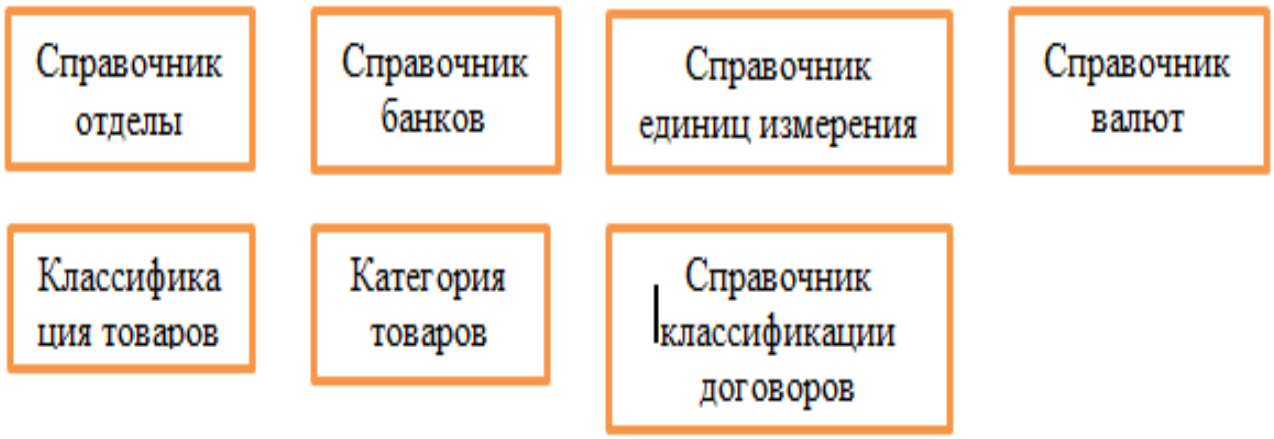


Рисунок 2.2 – Концептуальная модель БД не требующая внешних данных

2.2.2.2 Построение логической модели базы данных

На этапе построения логической модели БД, выделяют логические связи из концептуальной модели, затем конкретизируя эти связи строится логическая модель.

На рисунке 2.3 изображена логическая модель БД.

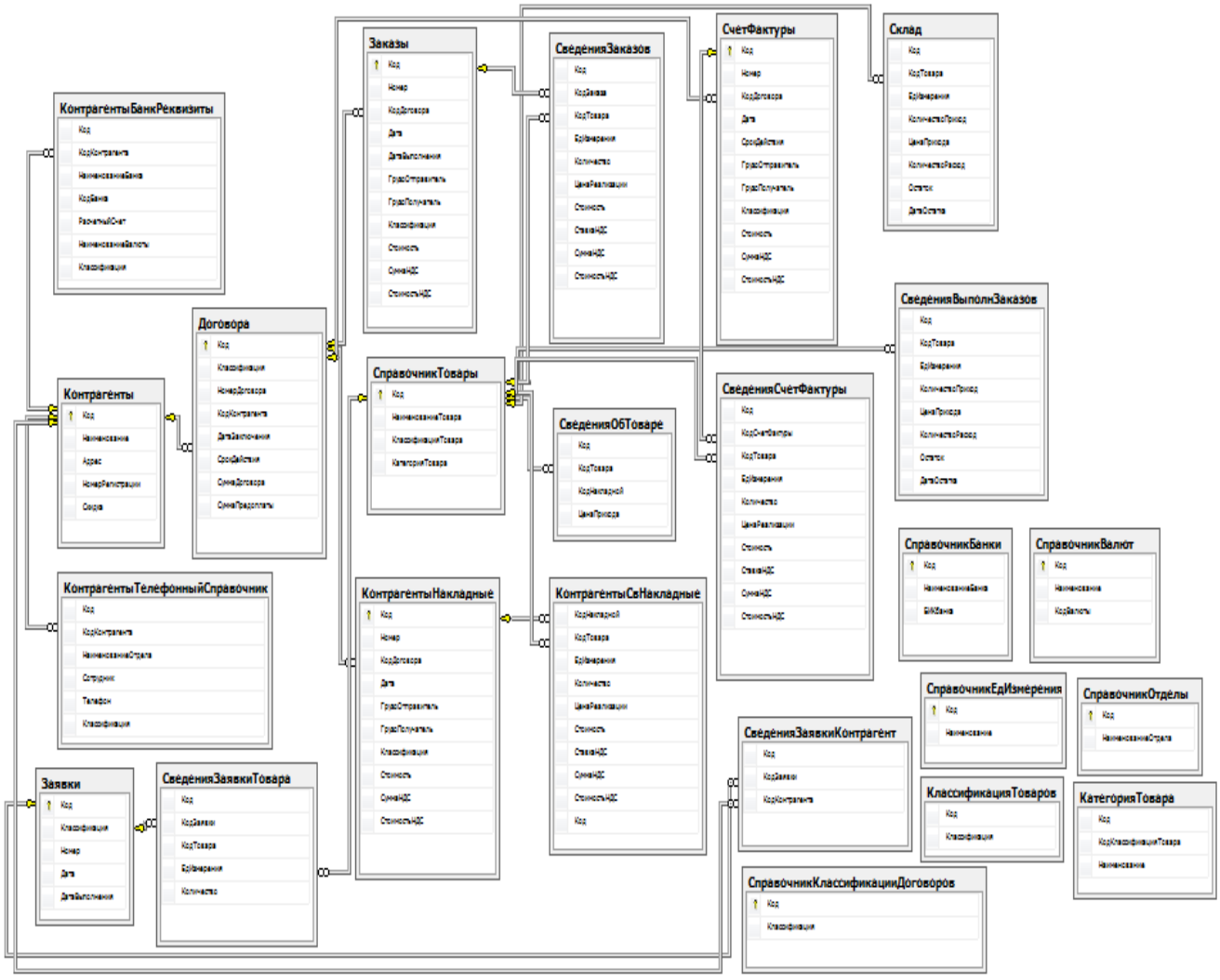


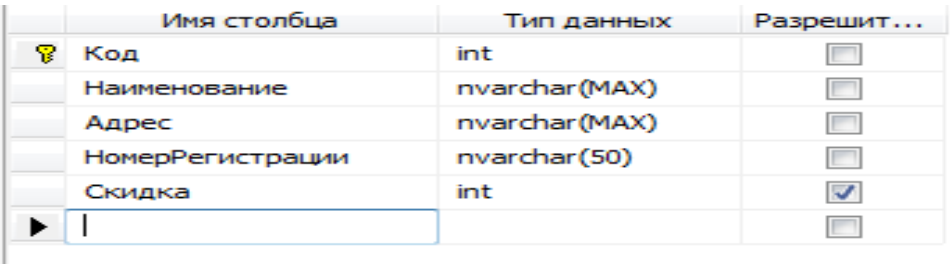
Рисунок 2.3 - Логическая модель базы данных

Для реализации БД была выбрана СУБД Microsoft SQL Server 2008.

Microsoft SQL Server 2008 представляет новое поколение масштабируемых решений в области систем управления базами и хранилищ данных для задач, требующих быстрого получения и анализа информации.

Приведем описание структуры таблиц для выбранной СУБД.

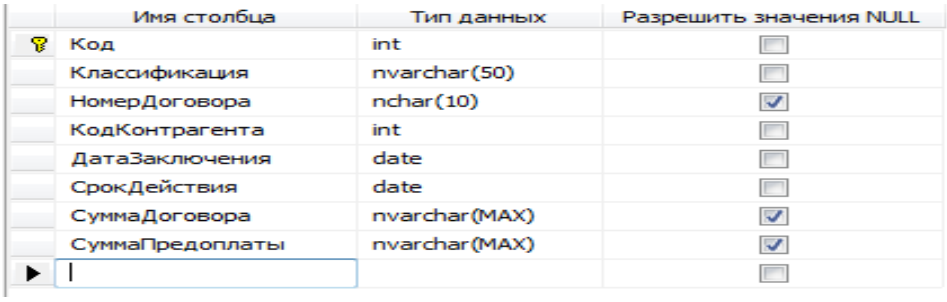
На рисунке 2.4 представлена таблица «Контрагенты»



Имя столбца	Тип данных	Разрешит...
Код	int	
Наименование	nvarchar(MAX)	
Адрес	nvarchar(MAX)	
НомерРегистрации	nvarchar(50)	
Скидка	int	☑

Рисунок 2.4 – Таблица «Контрагенты»

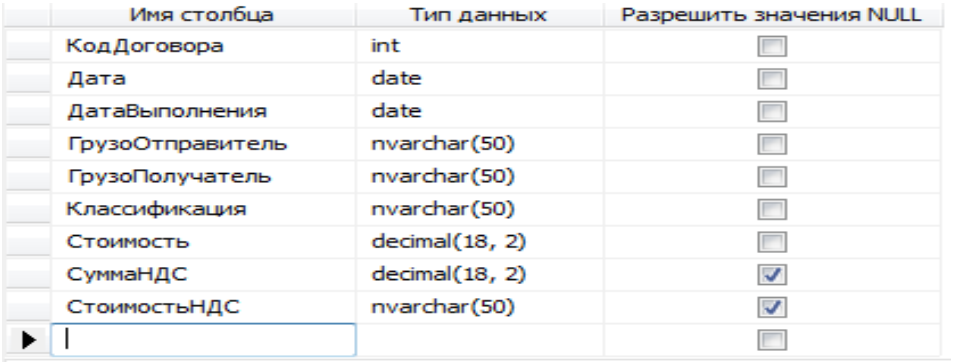
На рисунке 2.5 представлена таблица «Договора»



Имя столбца	Тип данных	Разрешить значения NULL
Код	int	
Классификация	nvarchar(50)	
НомерДоговора	nchar(10)	☑
КодКонтрагента	int	
ДатаЗаключения	date	
СрокДействия	date	
СуммаДоговора	nvarchar(MAX)	☑
СуммаПредоплаты	nvarchar(MAX)	☑

Рисунок 2.5 – Таблица «Договора»

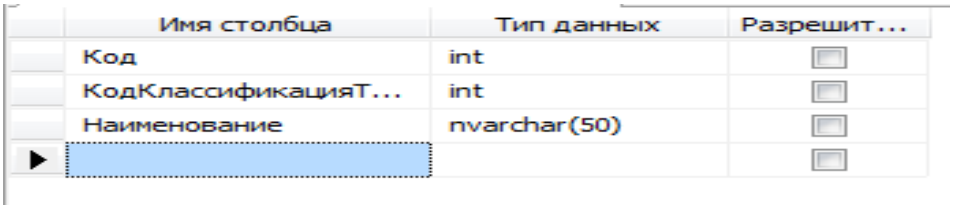
На рисунке 2.6 представлена таблица «Заказы»



Имя столбца	Тип данных	Разрешить значения NULL
КодДоговора	int	
Дата	date	
ДатаВыполнения	date	
ГрузоОтправитель	nvarchar(50)	
ГрузоПолучатель	nvarchar(50)	
Классификация	nvarchar(50)	
Стоимость	decimal(18, 2)	
СуммаНДС	decimal(18, 2)	☑
СтоимостьНДС	nvarchar(50)	☑

Рисунок 2.6 – Таблица «Заказы»

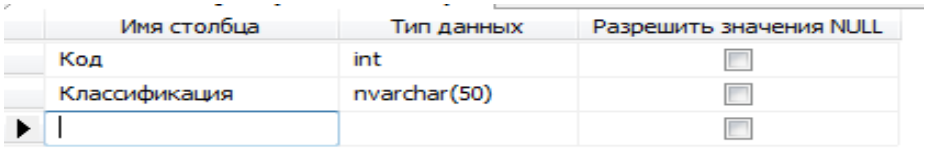
На рисунке 2.7 представлена таблица «КатегорияТоваров»



Имя столбца	Тип данных	Разрешит...
Код	int	
КодКлассификацияТ...	int	
Наименование	nvarchar(50)	

Рисунок 2.7 – Таблица «КатегорияТоваров»

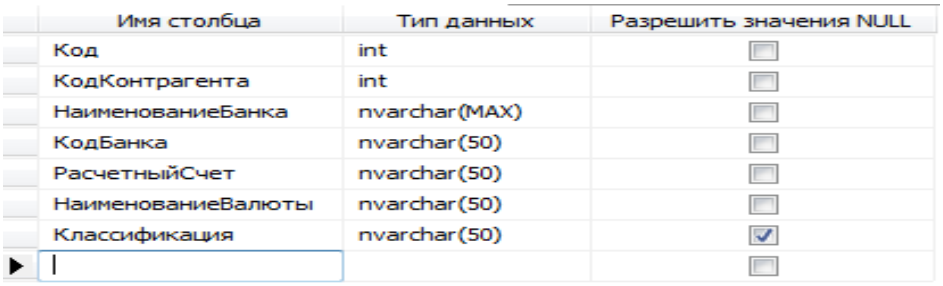
На рисунке 2.8 представлена таблица «КлассификацияТоваров»



Имя столбца	Тип данных	Разрешить значения NULL
Код	int	☐
Классификация	nvarchar(50)	☐
		☐

Рисунок 2.8 – Таблица «КлассификацияТоваров»

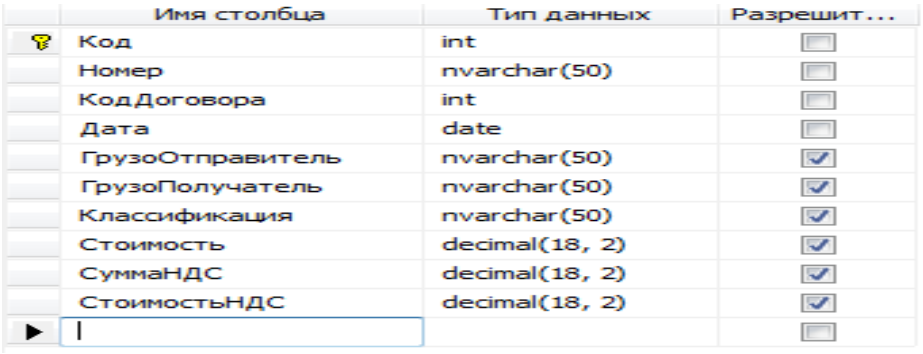
На рисунке 2.9 представлена таблица «КонтрагентыБанкРеквизиты»



Имя столбца	Тип данных	Разрешить значения NULL
Код	int	☐
КодКонтрагента	int	☐
НаименованиеБанка	nvarchar(MAX)	☐
КодБанка	nvarchar(50)	☐
РасчетныйСчет	nvarchar(50)	☐
НаименованиеВалюты	nvarchar(50)	☐
Классификация	nvarchar(50)	☑
		☐

Рисунок 2.9 – Таблица «КонтрагентыБанкРеквизиты»

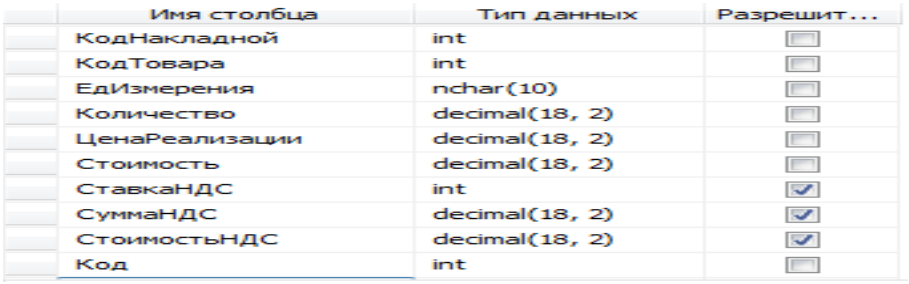
На рисунке 2.10 представлена таблица «КонтрагентыНакладные»



Имя столбца	Тип данных	Разрешит...
Код	int	☐
Номер	nvarchar(50)	☐
КодДоговора	int	☐
Дата	date	☐
ГрузоОтправитель	nvarchar(50)	☑
ГрузоПолучатель	nvarchar(50)	☑
Классификация	nvarchar(50)	☑
Стоимость	decimal(18, 2)	☑
СуммаНДС	decimal(18, 2)	☑
СтоимостьНДС	decimal(18, 2)	☑
		☐

Рисунок 2.10 – Таблица «КонтрагентыНакладные»

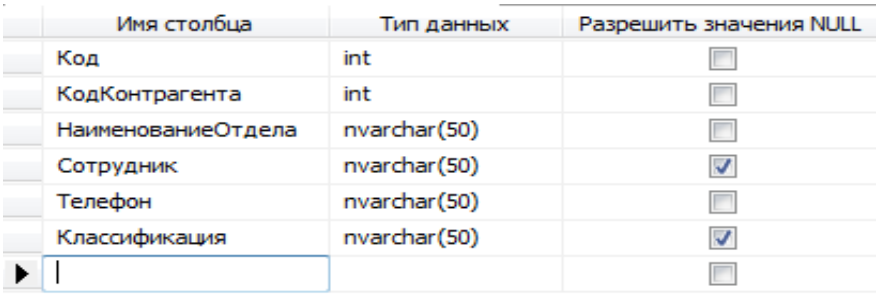
На рисунке 2.11 представлена таблица «КонтрагентыСвНакладные»



Имя столбца	Тип данных	Разрешит...
КодНакладной	int	☐
КодТовара	int	☐
ЕдИзмерения	nchar(10)	☐
Количество	decimal(18, 2)	☐
ЦенаРеализации	decimal(18, 2)	☐
Стоимость	decimal(18, 2)	☐
СтавкаНДС	int	☑
СуммаНДС	decimal(18, 2)	☑
СтоимостьНДС	decimal(18, 2)	☑
Код	int	☐

Рисунок 2.11 – Таблица «КонтрагентыСвНакладные»

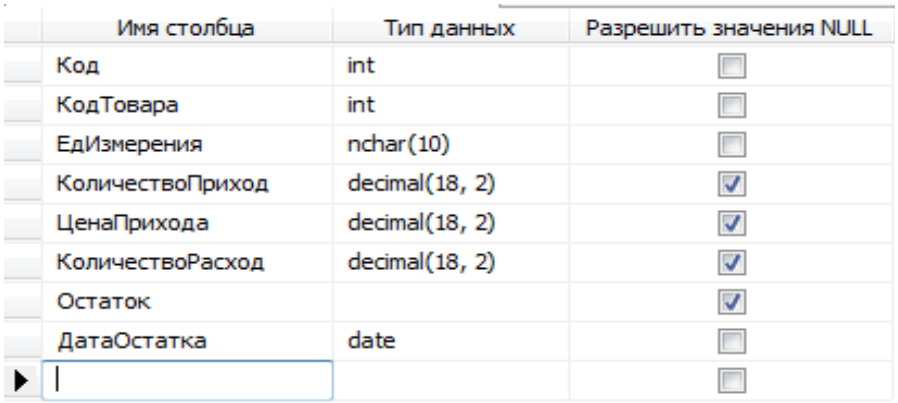
На рисунке 2.12 представлена таблица «КонтрагентыТелефонныйСправочник»



Имя столбца	Тип данных	Разрешить значения NULL
Код	int	☐
КодКонтрагента	int	☐
НаименованиеОтдела	nvarchar(50)	☐
Сотрудник	nvarchar(50)	☑
Телефон	nvarchar(50)	☐
Классификация	nvarchar(50)	☑
		☐

Рисунок 2.12 – Таблица «КонтрагентыТелефонныйСправочник»

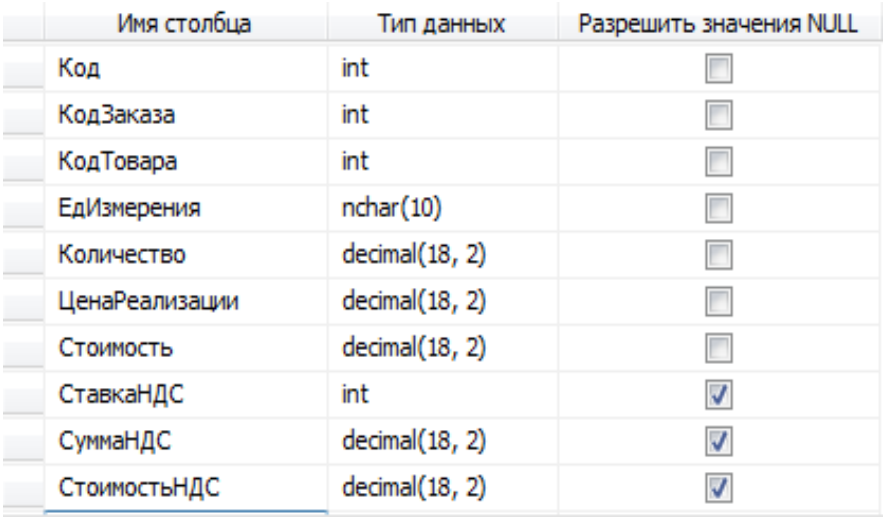
На рисунке 2.13 представлена таблица «СведенияВыполнЗаказов»



Имя столбца	Тип данных	Разрешить значения NULL
Код	int	☐
КодТовара	int	☐
ЕдИзмерения	nchar(10)	☐
КоличествоПриход	decimal(18, 2)	☑
ЦенаПрихода	decimal(18, 2)	☑
КоличествоРасход	decimal(18, 2)	☑
Остаток		☑
ДатаОстатка	date	☐
		☐

Рисунок 2.13 – Таблица «СведенияВыполнЗаказов»

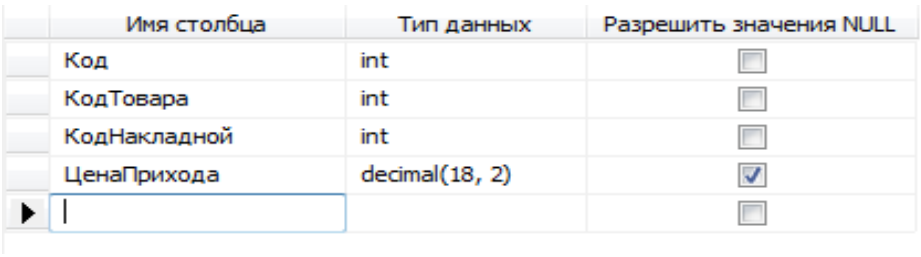
На рисунке 2.14 представлена таблица «СведенияЗаказов»



Имя столбца	Тип данных	Разрешить значения NULL
Код	int	☐
КодЗаказа	int	☐
КодТовара	int	☐
ЕдИзмерения	nchar(10)	☐
Количество	decimal(18, 2)	☐
ЦенаРеализации	decimal(18, 2)	☐
Стоимость	decimal(18, 2)	☐
СтавкаНДС	int	☑
СуммаНДС	decimal(18, 2)	☑
СтоимостьНДС	decimal(18, 2)	☑

Рисунок 2.14 – Таблица «СведенияЗаказов»

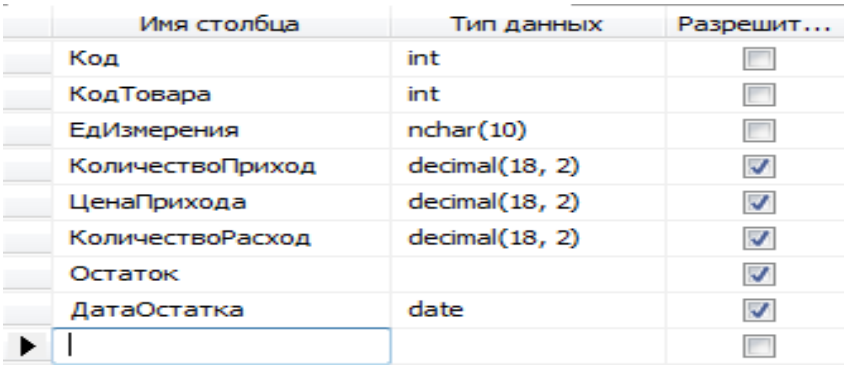
На рисунке 2.15 представлена таблица «СведенияОбТоваре»



Имя столбца	Тип данных	Разрешить значения NULL
Код	int	☐
КодТовара	int	☐
КодНакладной	int	☐
ЦенаПрихода	decimal(18, 2)	☑
		☐

Рисунок 2.15 – Таблица «СведенияОбТоваре»

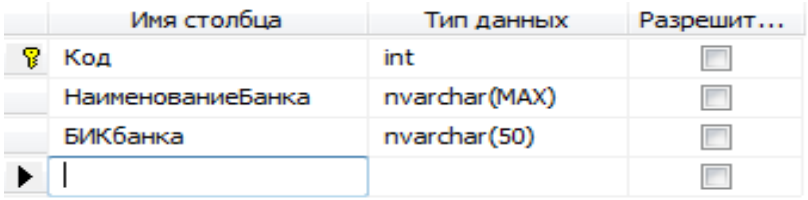
На рисунке 2.16 представлена таблица «Склад»



Имя столбца	Тип данных	Разрешит...
Код	int	☐
КодТовара	int	☐
ЕдИзмерения	nchar(10)	☐
КоличествоПриход	decimal(18, 2)	☑
ЦенаПрихода	decimal(18, 2)	☑
КоличествоРасход	decimal(18, 2)	☑
Остаток		☑
ДатаОстатка	date	☑
		☐

Рисунок 2.16 – Таблица «Склад»

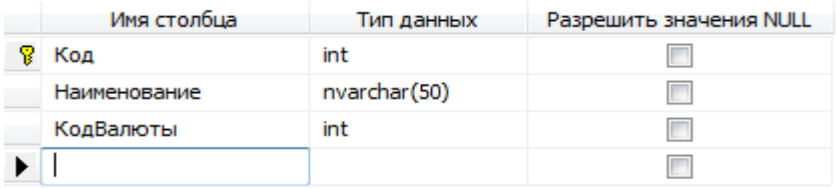
На рисунке 2.17 представлена таблица «СправочникБанки»



Имя столбца	Тип данных	Разрешит...
Код	int	☐
НаименованиеБанка	nvarchar(MAX)	☐
БИКбанка	nvarchar(50)	☐
		☐

Рисунок 2.17 – Таблица «СправочникБанки»

На рисунке 2.18 представлена таблица «СправочникВалют»



Имя столбца	Тип данных	Разрешить значения NULL
Код	int	☐
Наименование	nvarchar(50)	☐
КодВалюты	int	☐
		☐

Рисунок 2.18 – Таблица «СправочникВалют»

На рисунке 2.19 представлена таблица «СправочникЕдИзмерения»

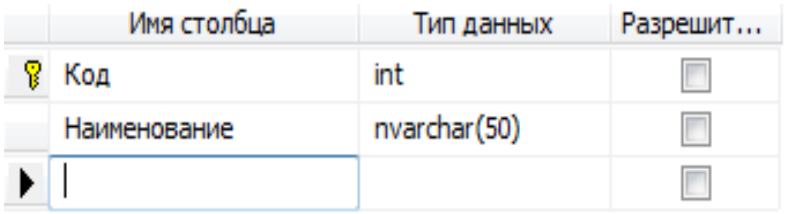


Рисунок 2.19 – Таблица «СправочникЕдИзмерения»

На рисунке 2.20 представлена таблица «СправочникКлассификацииДоговоров»

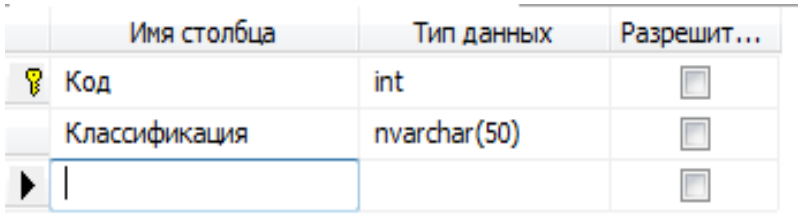


Рисунок 2.20 – Таблица «СправочникКлассификацииДоговоров»

На рисунке 2.21 представлена таблица «СправочникОтделы»

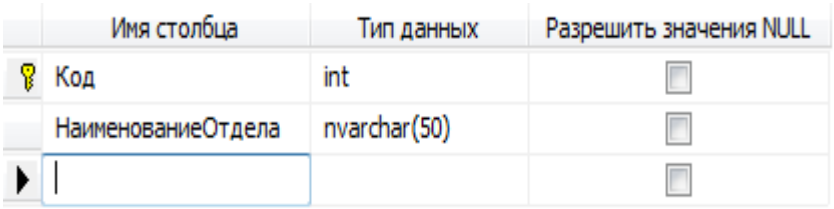


Рисунок 2.21 – Таблица «СправочникОтделы»

На рисунке 2.22 представлена таблица «СправочникТовары»

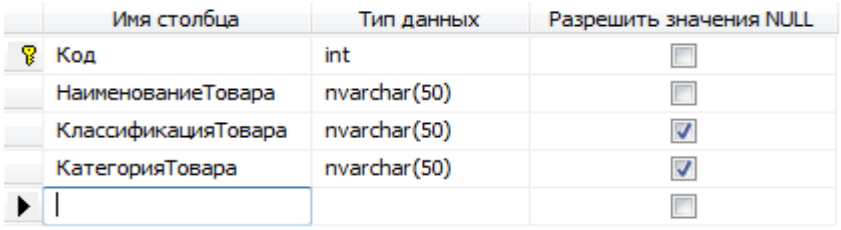
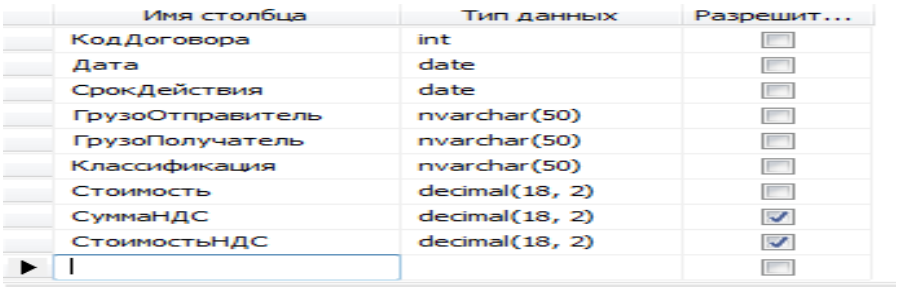


Рисунок 2.22 – Таблица «СправочникТовары»

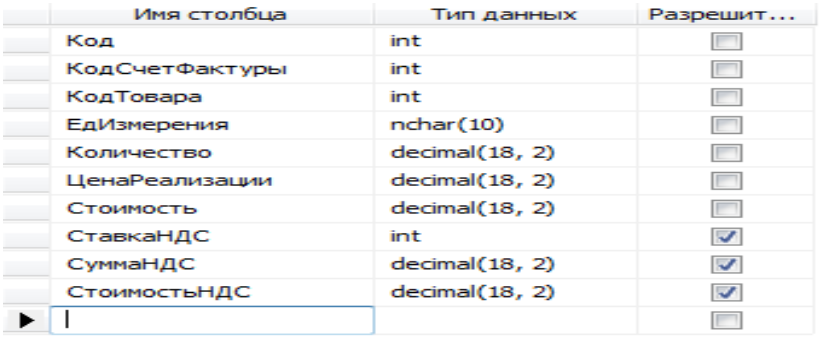
На рисунке 2.23 представлена таблица «СчетФактуры»



Имя столбца	Тип данных	Разрешит...
КодДоговора	int	☐
Дата	date	☐
СрокДействия	date	☐
ГрузоОтправитель	nvarchar(50)	☐
ГрузоПолучатель	nvarchar(50)	☐
Классификация	nvarchar(50)	☐
Стоимость	decimal(18, 2)	☐
СуммаНДС	decimal(18, 2)	☑
СтоимостьНДС	decimal(18, 2)	☑
		☐

Рисунок 2.23 – Таблица «СчетФактуры»

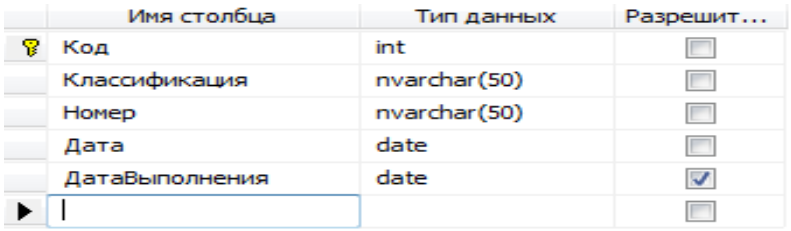
На рисунке 2.24 представлена таблица «СведенияСчетФактуры»



Имя столбца	Тип данных	Разрешит...
Код	int	☐
КодСчетФактуры	int	☐
КодТовара	int	☐
ЕдИзмерения	nchar(10)	☐
Количество	decimal(18, 2)	☐
ЦенаРеализации	decimal(18, 2)	☐
Стоимость	decimal(18, 2)	☐
СтавкаНДС	int	☑
СуммаНДС	decimal(18, 2)	☑
СтоимостьНДС	decimal(18, 2)	☑
		☐

Рисунок 2.24 – Таблица «СведенияСчетФактуры»

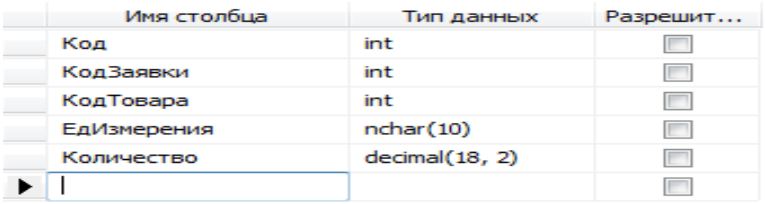
На рисунке 2.25 представлена таблица «Заявки»



Имя столбца	Тип данных	Разрешит...
🔑 Код	int	☐
Классификация	nvarchar(50)	☐
Номер	nvarchar(50)	☐
Дата	date	☐
ДатаВыполнения	date	☑
		☐

Рисунок 2.25 – Таблица «Заявки»

На рисунке 2.26 представлена таблица «СведенияЗаявкиТовара»



Имя столбца	Тип данных	Разрешит...
Код	int	☐
КодЗаявки	int	☐
КодТовара	int	☐
ЕдИзмерения	nchar(10)	☐
Количество	decimal(18, 2)	☐
		☐

Рисунок 2.26 – Таблица «СведенияЗаявкиТовара»

На рисунке 2.27 представлена таблица «СведенияЗаявкиКонтрагент»

Имя столбца	Тип данных	Разрешит…
Код	int	☐
КодЗаявки	int	☐
КодКонтрагента	int	☐
		☐

Рисунок 2.27 – Таблица «СведенияЗаявкиКонтрагент»

Таким образом, используя построенную модель базы данных, можно переходить к стадии проектирования приложения для работы автоматизированной системы учета и анализа торгово-закупочной деятельности.

2.2.3 Архитектура автоматизированной информационной системы

Подводя итоги этапу анализа объекта исследования, определим основные задачи автоматизации технологии управления торгово-закупочной деятельностью предприятия.

Задачи автоматизации технологии документооборота, учета и анализа торгово-закупочной деятельности, решаемые автоматизированной системой:

Оформление, учет принятых заказов, включающий в себя:

- заключение договоров;
- оформление заказа;
- выписку товарных накладных.

Анализ выполнения заказов, включающий в себя:

- обработку данных о заказанных товарах на основе оформленных заказов;
- обработку данных об закупленных товарах;
- обработку данных об проданных товарах;
- на основании обработанных данных определить сколько и каких товаров нужно закупить для выполнения заказа.

2.2.4 Проектирование пользовательского интерфейса

Структура меню программы показана на рисунке 2.28

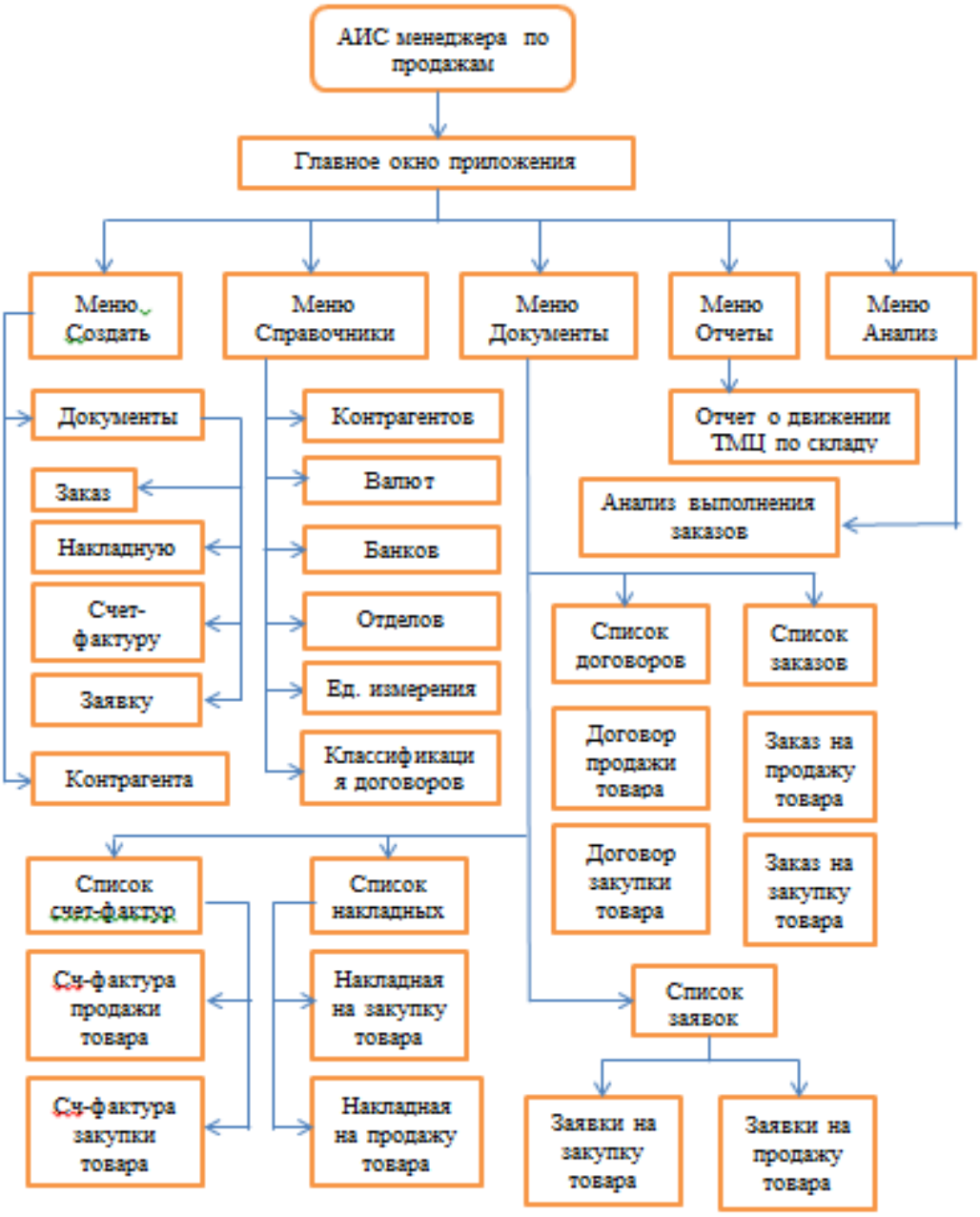


Рисунок 2.28 – Структура меню программы

Как показано на рисунке 2.28, разрабатываемое приложение должно включать в себя возможность работы с главной формой, формами ввода и корректировки справочников, формами ввода документов, формой получения отчетности и формой анализа.

При проектировании АИС была предложена следующая схема меню:

1. Все операции, необходимые пользователю, сгруппированы в четыре основных пункта меню: «Создать», «Справочники», «Документы», «Отчеты» и «Анализ».

2.Меню «Создать» содержит пункты меню, необходимые для ввода данных в документы и справочники.

3.Меню «Справочники» содержит пункт меню, позволяющий просмотреть сведения обо всех контрагентах.

4.Меню «Документы» содержит пункты меню, необходимые для просмотра и изменения, добавления и удаления данных из документов.

5.Меню «Отчеты» позволяет сформировать необходимую информацию о движении товаров согласно товарным накладным.

6.Меню «Анализ» позволяет проанализировать ситуацию с выполнением заказов, для дальнейшего принятия решений.

На рисунке 2.29 приведена технологическая схема ввода документов в базу.

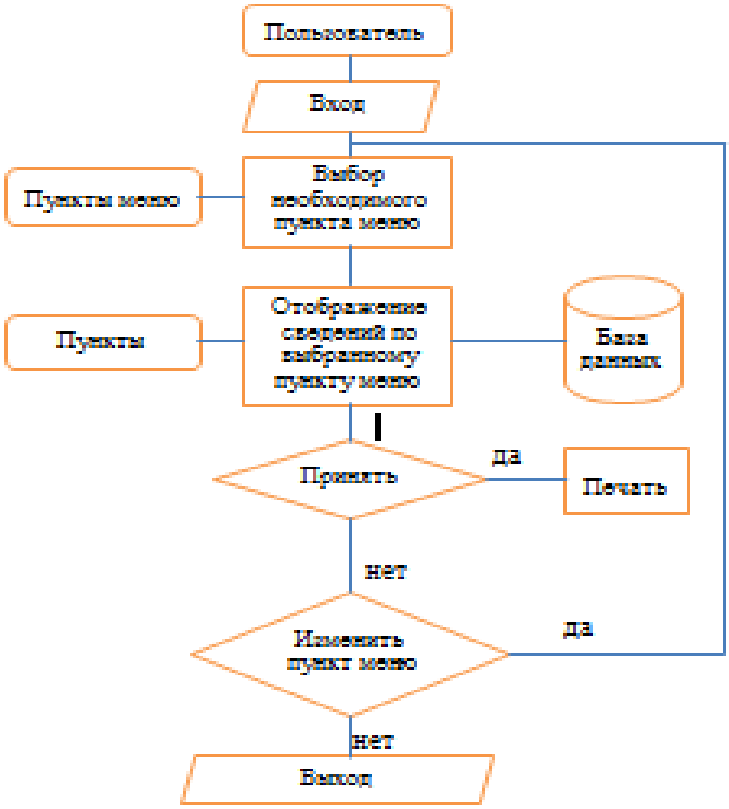


Рисунок 2.29 – Технологическая схема работы с документами

2.2.5 Проектирование классов

На рисунке 2.30 представлена диаграмма классов, отвечающих за реализацию функционала пользовательского приложения.



Рисунок 2.30 – Диаграмма классов

Опишем назначение каждого из классов:

1. Класс «Справочники», является классом оболочкой для других классов группы справочники. Это такие, как «Контрагенты», «СправочникТовары». Это сделано для того, чтобы лучше ориентироваться в коде.
2. Класс «Контрагенты» отвечает за сохранение необходимой информации об участниках сделки .
3. Класс «БанкРеквизитыКонтрагента» является вложенным классом в класс «Контрагенты» и отвечает за сохранение информации о банковских счетах контрагента.
4. Класс «ТелефонныйСправочникКонтрагентов» является вложенным классом в класс «Контрагенты» и отвечает за сохранение информации о контактных телефонах контрагента.
5. Класс «СправочникТовары» отвечает за сохранение необходимой информации о товарах.

6. Класс «Документы», является классом оболочкой для других классов группы документов. Это такие, как Это сделано для того, чтобы лучше ориентироваться в коде (Приложение А).

7. Класс «Договор» отвечает за сохранение необходимой общей информации об условиях сделки и является классом родителем для разновидностей других договоров.

8. Класс «Договор_Купли_Продажи» является классом наследником от класса «Договор» и отвечает за сохранение необходимой информации связанной с куплей-продажей товаров.

9. Класс «Заказы» отвечает за сохранение необходимой информации о сделанных заказах товаров.

10. Класс «СведенияЗаказы» является вложенным классом в класс «Заказы» и отвечает за сохранение необходимой информации об заказанном товаре.

11. Класс «Накладные» отвечает за сохранение необходимой информации о сделанных закупках и продажах товаров.

12. Класс «СведенияНакладные» является вложенным классом в класс «Накладные» и отвечает за сохранение необходимой информации об закупленных и проданных товарах.

13. Класс «СчетФактуры» отвечает за сохранение информации о товарах, поставщиках и плательщиках, необходимой для совершения предоплаты за товар.

14. Класс «СведенияСчетФактуры» является вложенным классом в класс «СчетФактуры» и отвечает за сохранение необходимой информации об закупаемых и продаваемых товарах.

15. Класс «ЗаявкиТовара» является вспомогательным классом и предназначен для заказа товаров у поставщиков, а также для оформления документов на продажу товара клиентам.

16. Класс «СведенияЗаявкиТовара» является вложенным классом в класс «ЗаявкиТовара» и отвечает за сохранение необходимой информации об заказываемом товаре, поставщиках или покупателе.

17. Класс «Таблица» является вспомогательным классом и предназначен для работы с таблицами приложения.

18. Класс «Работа_с_Word» является вспомогательным классом и предназначен для работы с документом Word, т.е. передачи в него данных с приложения, с последующим его просмотром и при необходимости выводом на печать.

ГЛАВА 3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ

3.1 Средства разработки

Языком программирования для реализации приложения был выбран C#. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов, делегаты, атрибуты, события, свойства, обобщенные типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения.

В качестве ОС для разработки была выбрана Windows 7. Среда для разработки ПО- Microsoft Visual Studio. Фреймворк ASP.NET MVC. Рассмотрим их более детально.

3.1.1 Среда разработки

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, Xbox, Windows Phone .NET Compact Framework и Silverlight.

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 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 (плагины) для расширения функциональности практически на каждом уровне.

Microsoft Visual Studio обладает мощными инструментами программирования. Она помогает при написании кода, вне зависимости от используемого языка C#, Visual Basic, F#, C++, HTML, JavaScript, CSS, Python и

других, предоставляя помощь в реальном времени. IntelliSense описывает API по мере ввода, а автоматическое завершение увеличивает скорость и точность работы. Средство подсказки позволяет проверять определения API. Проблемные места выделяются волнистым подчеркиванием, которое зачастую отображаются при вводе.

Visual Studio предоставляет прекрасные возможности отладки вне зависимости от языка, который вы используете: от C#, VB и C++ до JavaScript и Python, XAML и HTML. Возможна отладка всех поддерживаемых языков. Отладка пользовательского интерфейса.

Можно создавать и развертывать современные веб-приложения. Разработка веб-приложений с помощью ASP.NET, Node.js, Python и JavaScript. Используя мощные веб-платформы, такие как AngularJS, jQuery, Bootstrap, Django и Backbone.js.

3.1.2 Система управления базами данных

Для решения поставленной задачи выбираем систему управления базами данных (СУБД) Microsoft SQL Server.

В своем составе система имеет средства создания баз данных, работы с информацией баз данных, перенесения данных из других систем и в 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. Система не содержит средств разработки клиентских приложений [6].

Для взаимодействия с базой данных использовался Entity Framework. Entity Framework представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, то на концептуальном уровне, который нам предлагает Entity Framework, мы уже работаем с объектами.

3.2 Аномалии и защитное программирование

Аномалии и защитное программирование рассмотрим в таблице 3.1

Таблица 3.1- Аномалии и защитное программирование

Описание аномалии	Реакции на аномалию	Способ защиты
Попытка добавить значение NULL	Вывод текстового сообщения без изменения структуры	Защищено программно
Ввод в числовые поля текстовых значений	Запрет ввода текстовых значений в числовые поля	Защищено программно
Ввод в текстовые поля цифровых значений	Запрет ввода цифровых значений в текстовые поля	Защищено программно
Ввод неправильного значения времени (вроде 45:85)	Проверка введённого значения с диапазоном от 00:00 до 23:59. При несоответствии отображение значка ошибки возле текстового поля и передача ему фокуса ввода.	Защищено программно
Ввод несуществующей записи в базе о контрагенте	Выбор из списка	Защищено программно

3.3 Тестирование и отладка

Тестирование - процесс многократного выполнения программы с целью выявления ошибок.

Отладка – исправление ошибок, найденных с помощью тестирования.

Необходимой частью каждого теста должно являться описание ожидаемых результатов работы программы, чтобы можно было быстро выяснить наличие или отсутствие ошибки в ней. В любой организации, занимающийся разработкой программных продуктов имеются специальные люди (Тестеры), занимающиеся тестированием программ которые выявляют ошибки и недоработки в программе и сообщают о них программистам, которые в свою очередь их исправляют [30].

Но наиболее эффективно тестирование проводится при эксплуатации программы в реальных условиях и реальными пользователями, которые сообщают обо всех найденных ошибках разработчику, который в последствии занимается их исправлением.

Так как в данном случае разработкой занимается не организация, а один человек то и тестирование и исправление тоже осуществляется одним человеком, что менее эффективно и является очень длительным процессом.

Данный проект разрабатывался и тестировался одновременно, то есть каждый раз при разработке какой либо новой части программы, написанная ранее программа проверялась на работоспособность и корректировалась с учётом найденных ошибок.

3.4 Анализ результатов решения

В соответствии с пунктом 3.4 тестирование и отладка производились непосредственно во время разработки проекта. После завершения работы над программой было еще раз произведено полное тестирование.

Все найденные ошибки были успешно устранены.

Получившийся в результате программный продукт удовлетворяет всем предъявленным требованиям, полностью работоспособен, не требует принципиальных доработок, и готов к эксплуатации.

3.5 Руководство пользователя

Программа служит для эффективной и быстрой работы с информацией, а также для автоматического составления различных отчетов на основе информации, которая занесена в базу данных.

Программа позволяет значительно экономить рабочее время сотрудника организации.

Внедрение программы поможет существенно снизить нагрузку при обработке информации и оформлении необходимой документации.

Запуск программы.

Запуск программы осуществляется двойным нажатием левой клавиши мыши на ярлыке программы. После запуска программного продукта на экране появится «**Главная форма**», которая выглядит следующим образом:

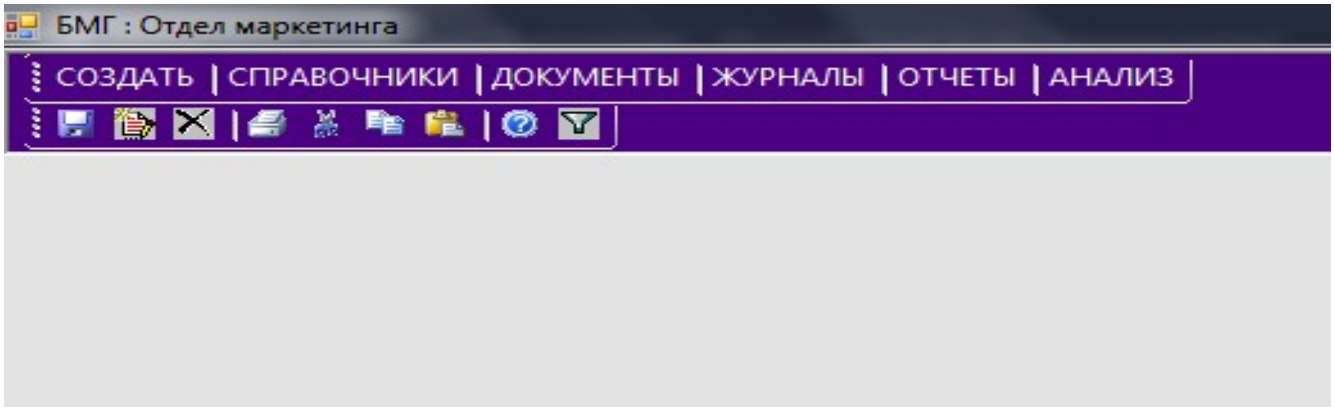


Рисунок 3.1 - Главная форма

Нажмем кнопку «СОЗДАТЬ», откроется список для ввода данных:

- «Контрагент» ;
- «Документы» :
  - «Заказ»;
  - «Счет-фактуру»;
  - «Накладную»;
  - «Заявку».

Ввод данных в поля документов максимально автоматизирован : некоторые заполняются сразу же при открытии формы документа, некоторые, при получении ими ввода, заполняются автоматически, для некоторых организован ввод-подсказка, т.е. при вводе данных идет автоматический поиск этих данных в базе данных, если они есть, то сразу записываются в поле, а если нету, то продолжаем ввод. Введенные данные, которых нету в БД, при сохранении документа, будут занесены в БД и в следующий раз будут доступны при вводе. На некоторых полях, при получении ими ввода данных, автоматически открываются вспомогательные окна-поисковики этих данных или раскрывающиеся списки. Организация такого ввода, позволяет облегчить и ускорить оформление документов.

Интерфейс окон приложения максимально приближен к реальному виду документов, что делает их более понятными и восприимчивыми пользователем.

Создание-ввод документов можно производить как по отдельности, так и через принятие заявки, а уже введя в нее данные, выбираем из окна заявки

приложения, какой дальше будем оформлять документ, выбрав его, все данные введенные в заявку, передадутся в этот документ и останется только ввести недостающие.

Далее показаны примеры и результаты работы с меню.

Нажмем кнопку «СОЗДАТЬ» - «Контрагента», в открывшемся окне вводим данные и нажимаем кнопку «Далее», результат представим на рисунках 3.2 – 3.4.

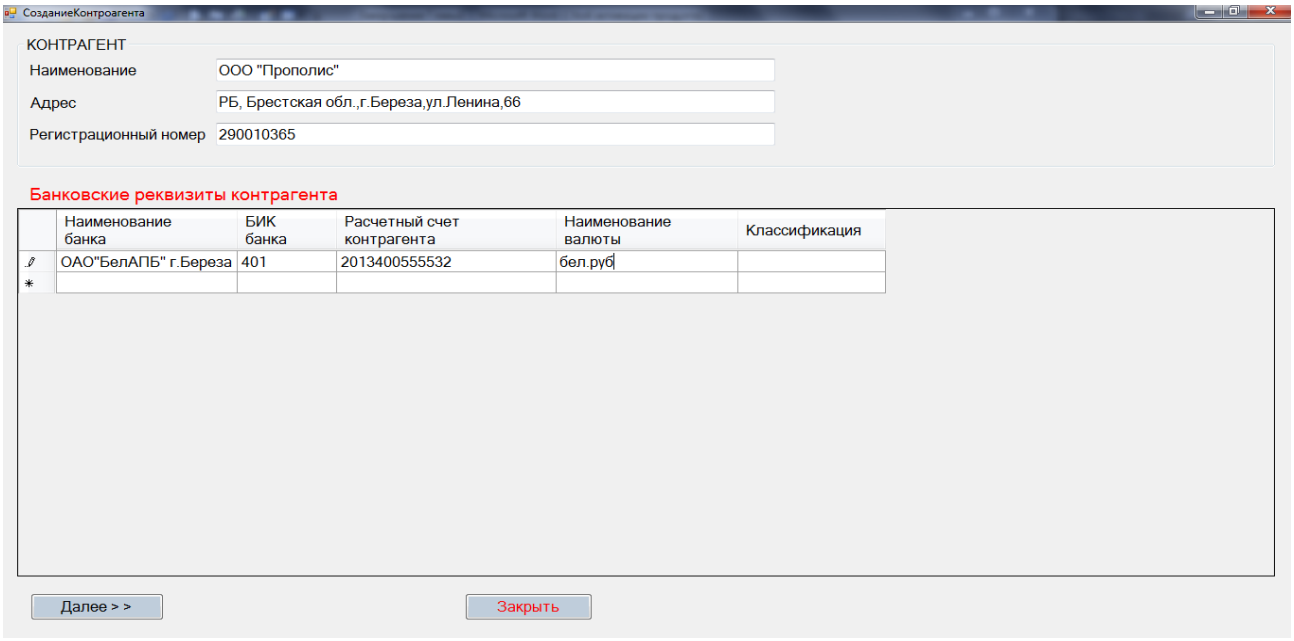


Рисунок 3.2 – Заполнение данных по Контрагенту

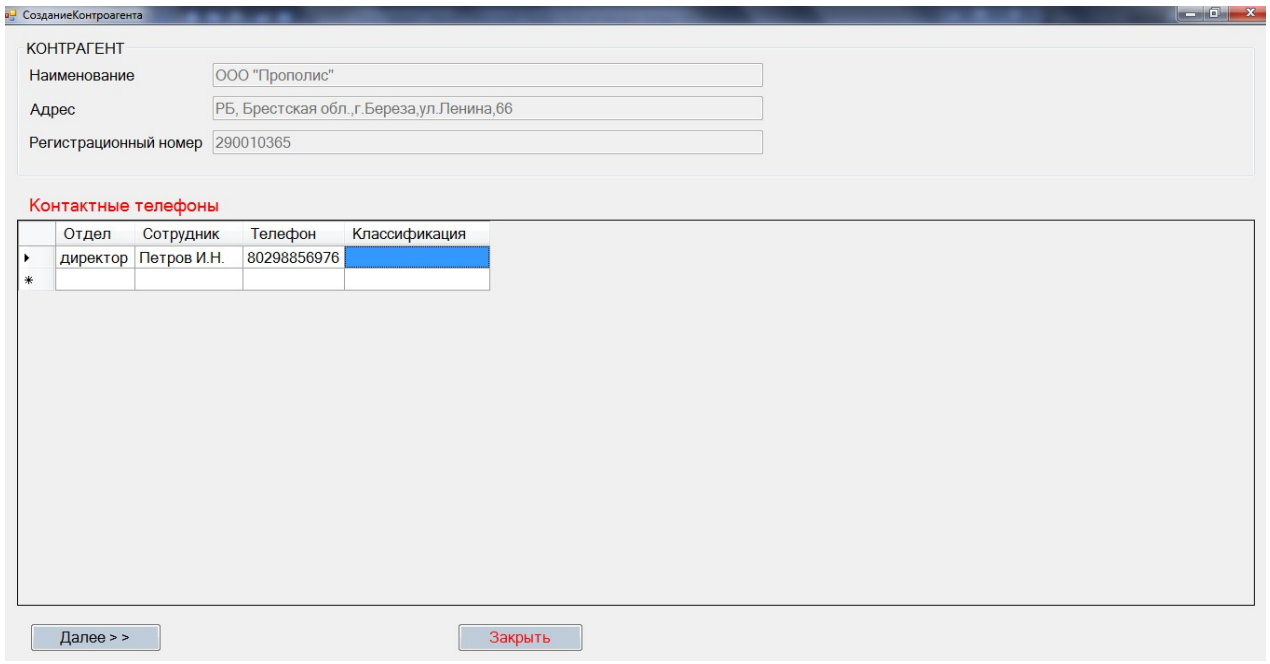


Рисунок 3.3 – Заполнение данных по Контрагенту

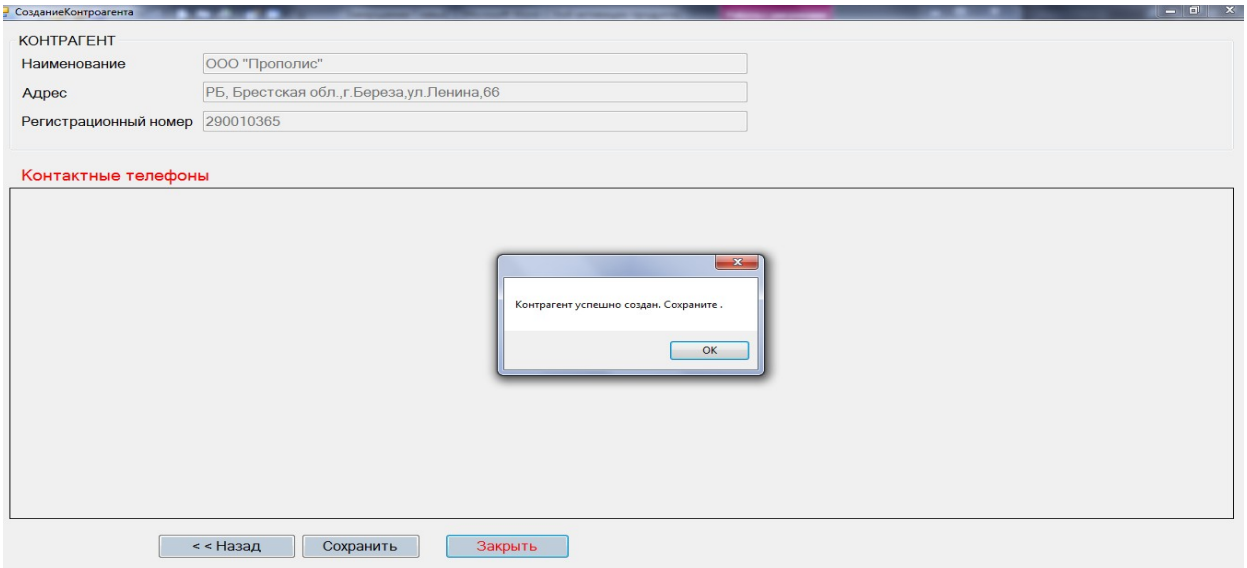


Рисунок 3.4 – Заполнение данных по Контрагенту

Нажмем кнопку «СОЗДАТЬ» - «Документы» - «Счет-фактуру», в открывшемся окне вводим данные и нажимаем кнопку «Сохранить», результаты ввода представлены на рисунках 3.5-3.8.

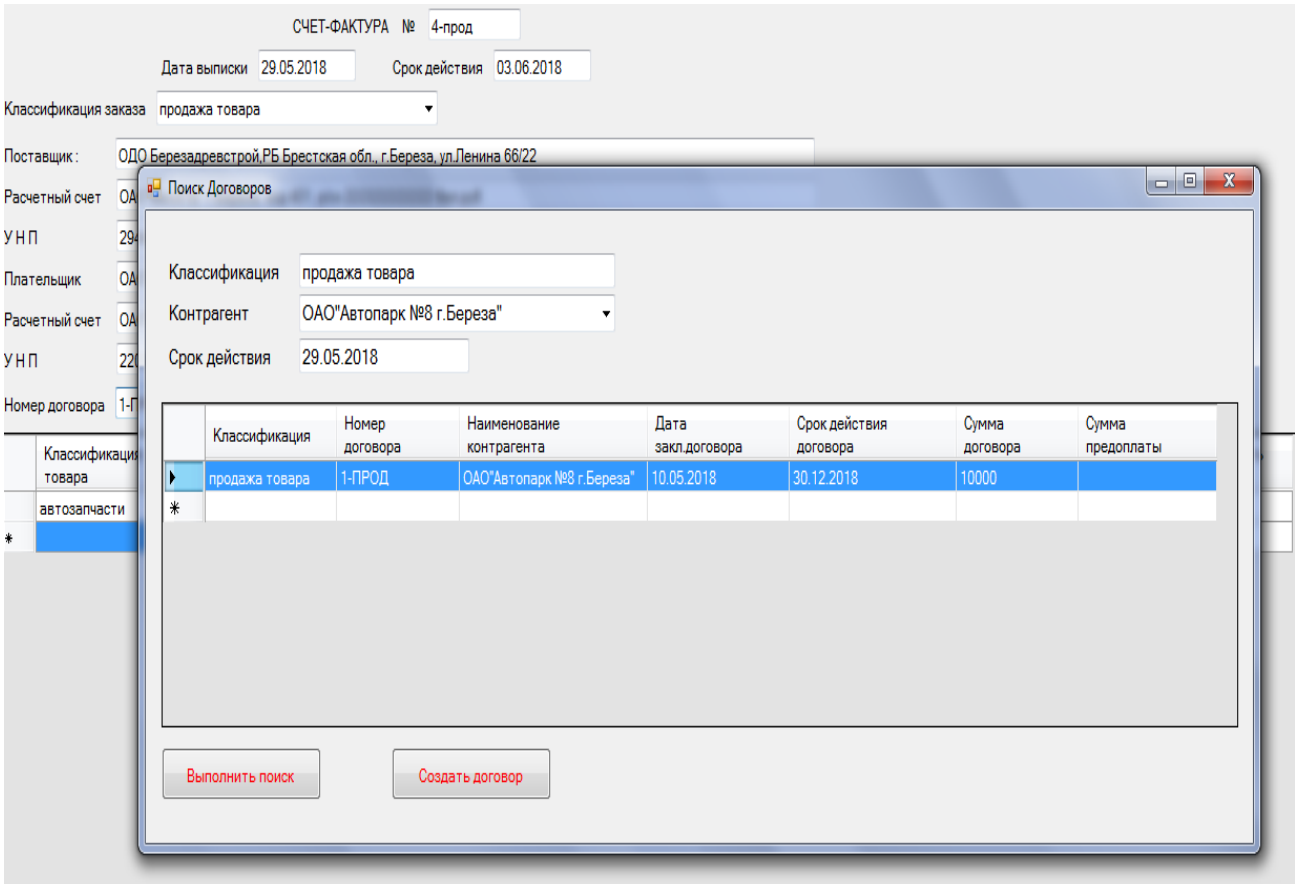


Рисунок 3.5 – Создание счет-фактуры

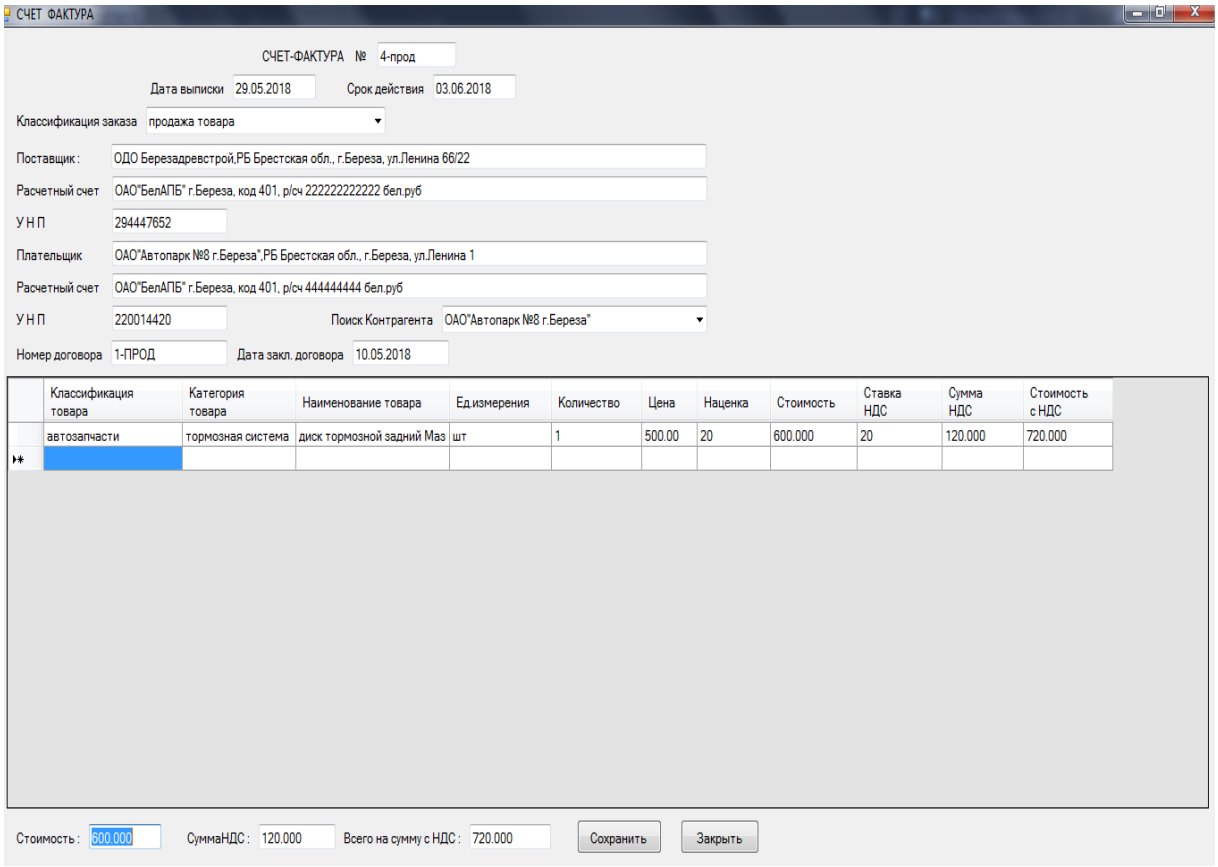


Рисунок 3.6 – Создание счет-фактуры

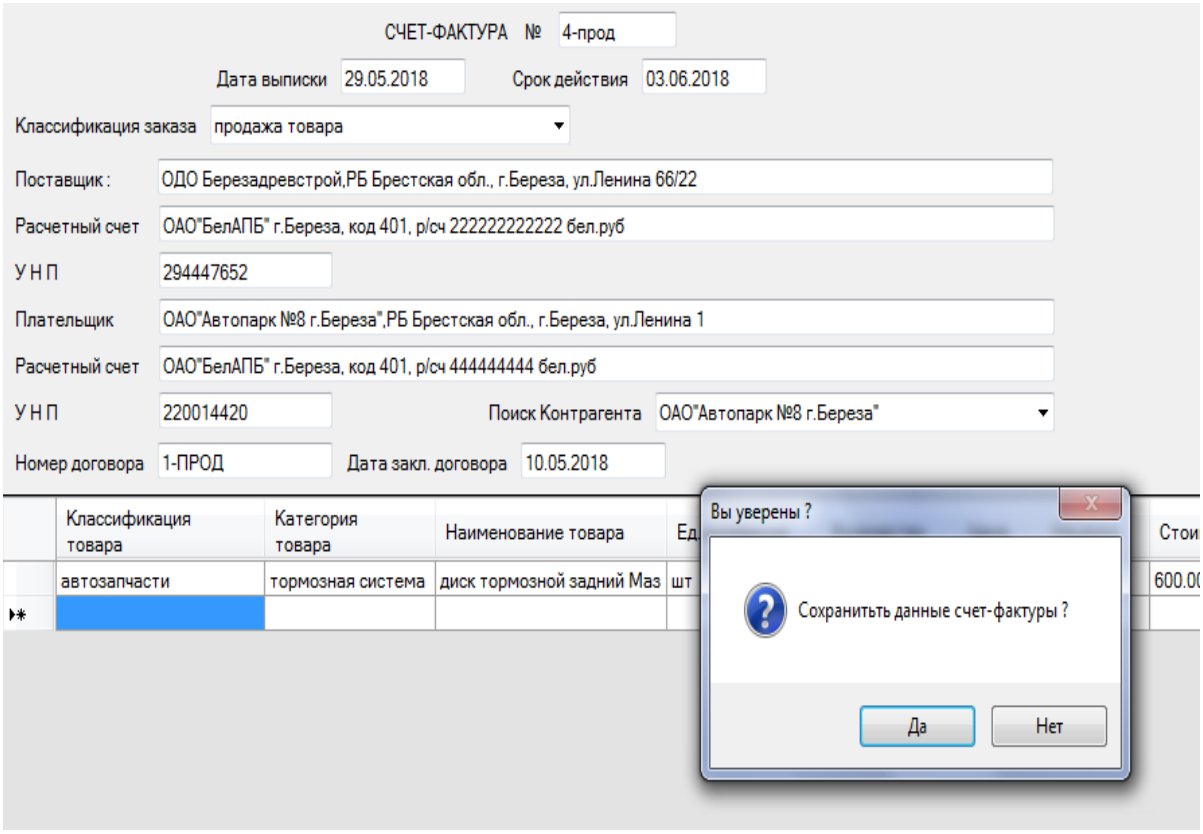


Рисунок 3.7 – Создание счет-фактуры

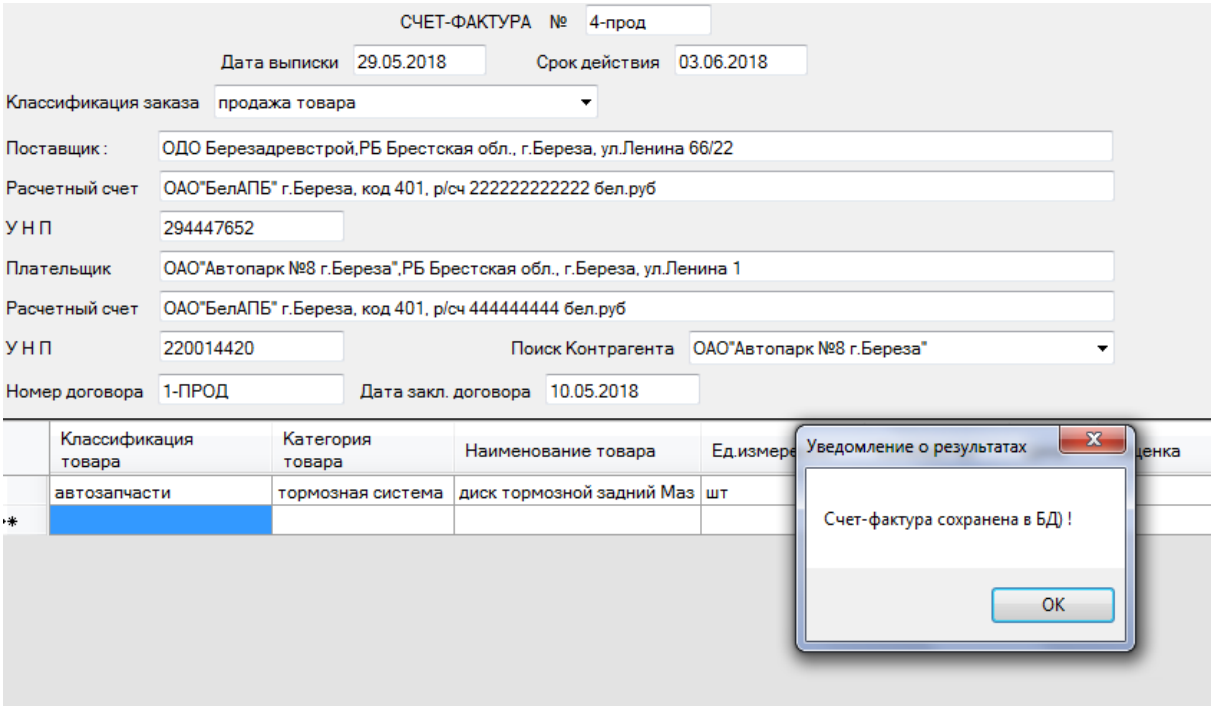


Рисунок 3.8 – Создание счет-фактуры

Нажмем кнопку «СОЗДАТЬ» - «Документы»- «Заказ», в открывшемся окне вводим данные и нажимаем кнопку «Сохранить», результат представим на рисунке 3.9.

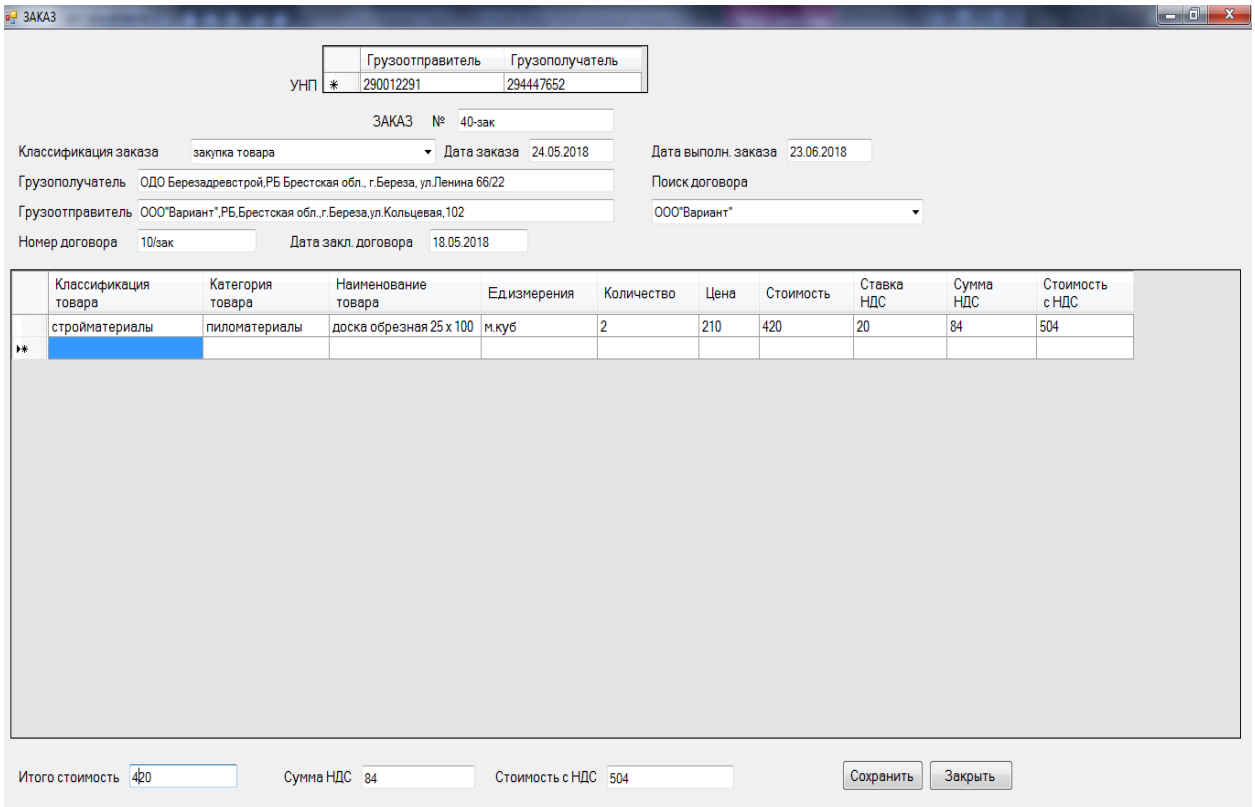


Рисунок 3.9 – Создание Заказа

Нажмем кнопку «СОЗДАТЬ» - «Документы»- «Накладную», в открывшемся окне вводим данные и нажимаем кнопку «Сохранить», результат представим на рисунке 3.10.

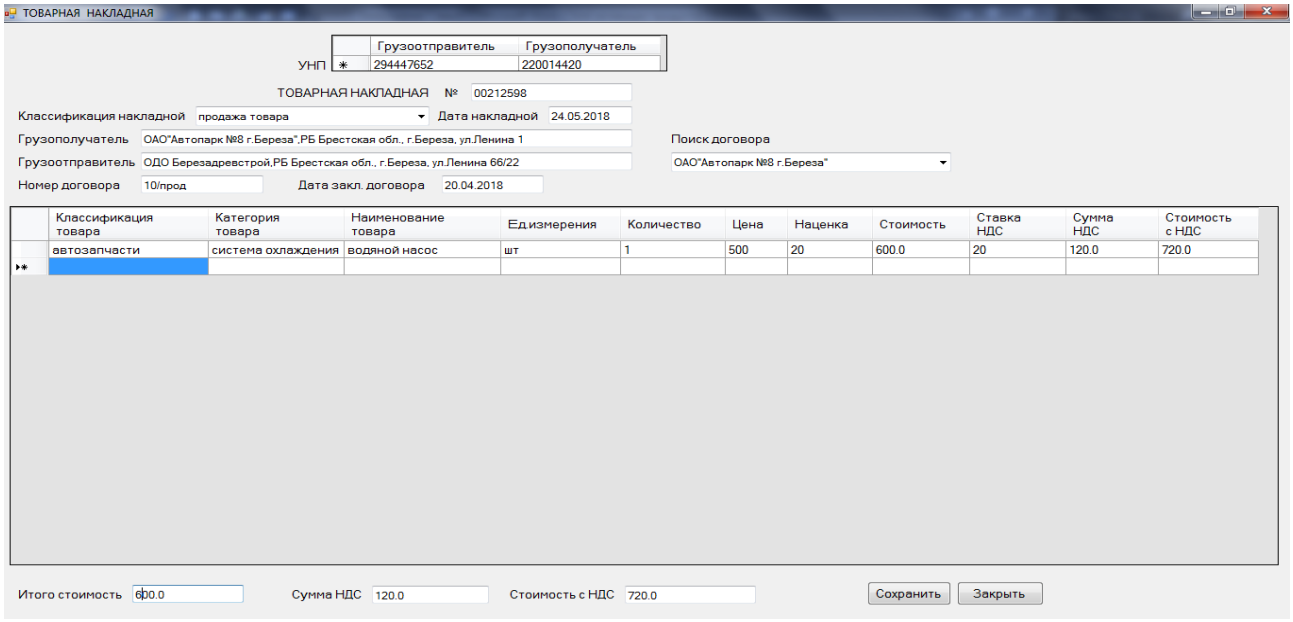


Рисунок 3.10 – Создание Накладной

Нажмем кнопку «СОЗДАТЬ» - «Документы»- «Заявку», в открывшемся окне вводим данные и нажимаем кнопку «Сохранить», результат представлен на рисунках 3.11-3.13.

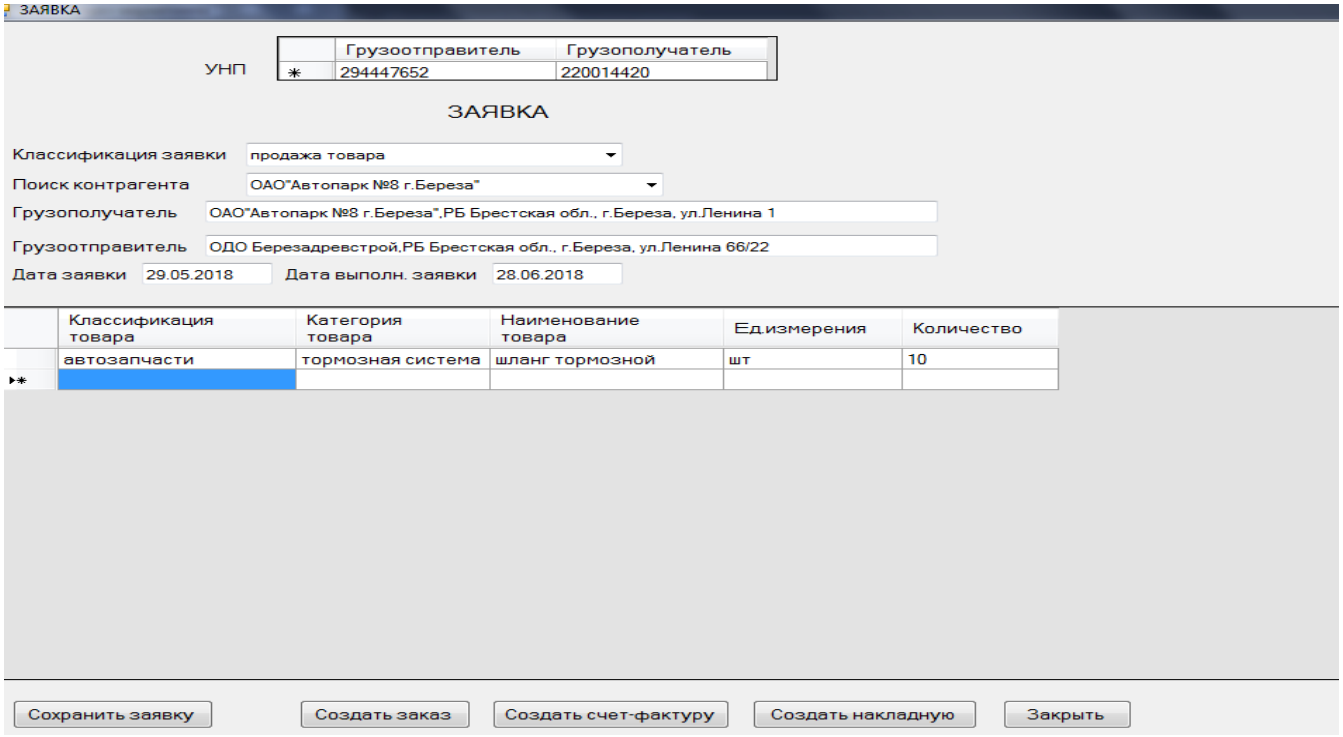


Рисунок 3.11 – Создание Заявки

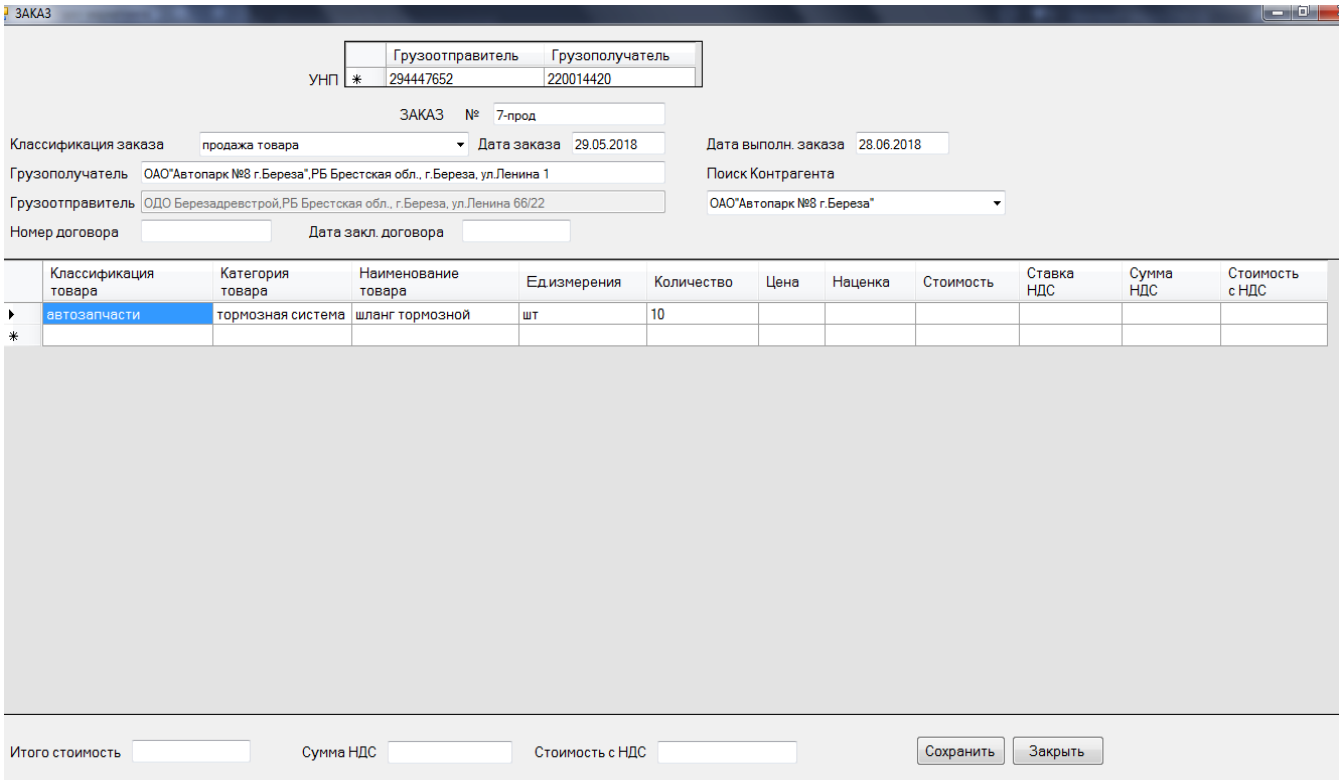


Рисунок 3.12 – Заполнение Заказа через Заявку

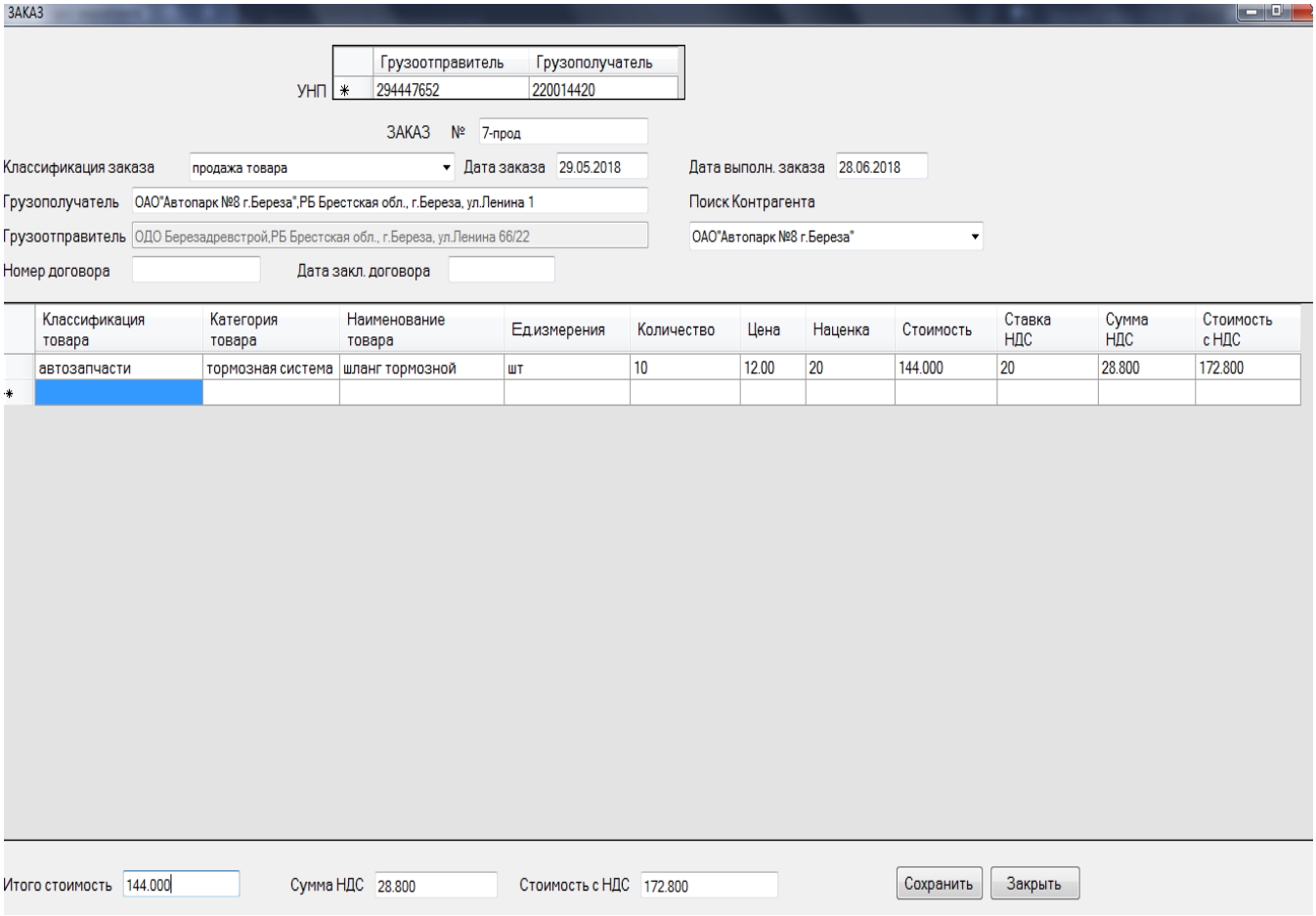
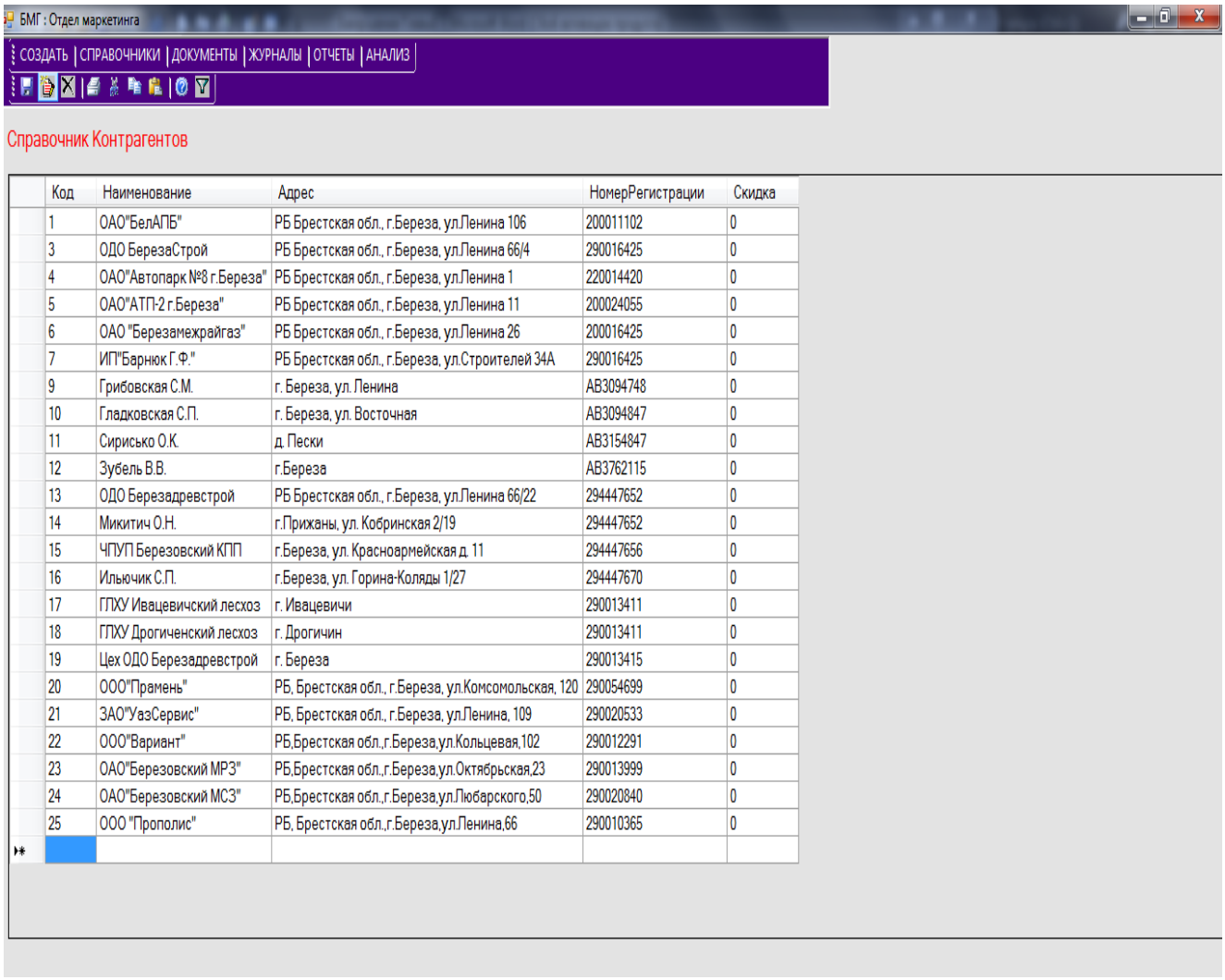


Рисунок 3.13 – Заполнение Заказа через Заявку

Нажмем кнопку «СПРАВОЧНИКИ», откроется список для просмотра и ввода данных:

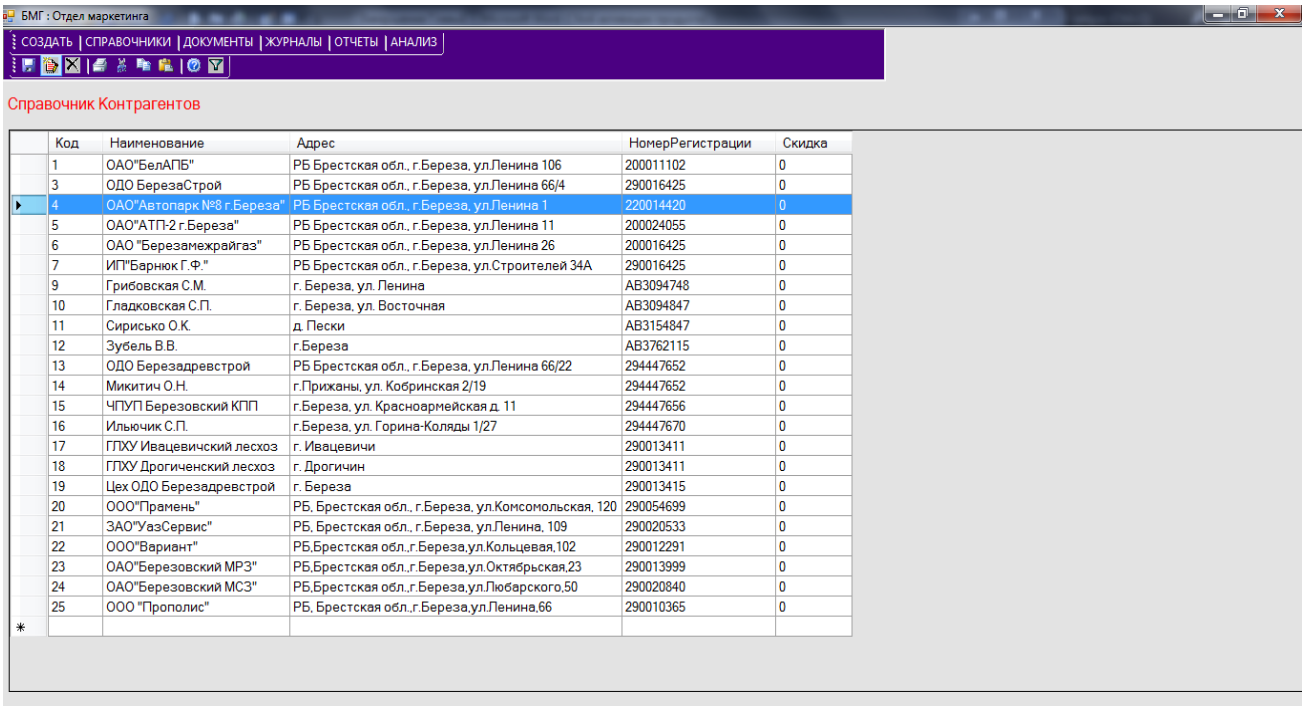
- «Контрагентов»;
- «Товаров/Услуг»;
- «Валют»;
- «Единиц измерения»;
- «Банков»;
- «Классификации договоров»;
- «Отделы».

Нажмем кнопку «СПРАВОЧНИКИ» - «Контрагентов», в открывшемся окне списка Контрагентов выделяем строку и щелкаем по ней два раза , откроется окно для просмотра данных выбранного контрагента . В открывшемся окне мы можем не только просмотреть, но и изменить, или удалить, или сохранить данные банковских реквизитов или контактных телефонов нажатием соответствующих кнопок, результаты представлены на рисунках 3.14-3.16.



Код	Наименование	Адрес	НомерРегистрации	Скидка
1	ОАО"БелАПБ"	РБ Брестская обл., г.Береза, ул.Ленина 106	200011102	0
3	ОДО БерезаСтрой	РБ Брестская обл., г.Береза, ул.Ленина 66/4	290016425	0
4	ОАО"Автопарк №8 г.Береза"	РБ Брестская обл., г.Береза, ул.Ленина 1	220014420	0
5	ОАО"АТП-2 г.Береза"	РБ Брестская обл., г.Береза, ул.Ленина 11	200024055	0
6	ОАО "Березамежрайгаз"	РБ Брестская обл., г.Береза, ул.Ленина 26	200016425	0
7	ИП"Барнюк Г.Ф."	РБ Брестская обл., г.Береза, ул.Строителей 34А	290016425	0
9	Грибовская С.М.	г. Береза, ул. Ленина	AB3094748	0
10	Гладковская С.П.	г. Береза, ул. Восточная	AB3094847	0
11	Сирисько О.К.	д. Пески	AB3154847	0
12	Зубель В.В.	г.Береза	AB3762115	0
13	ОДО Березадревстрой	РБ Брестская обл., г.Береза, ул.Ленина 66/22	294447652	0
14	Микитич О.Н.	г.Прижаны, ул. Кобринская 2/19	294447652	0
15	ЧПУП Березовский КПП	г.Береза, ул. Красноармейская д. 11	294447656	0
16	Ильючик С.П.	г.Береза, ул. Горина-Коляды 1/27	294447670	0
17	ГЛХУ Ивацевичский лесхоз	г. Ивацевичи	290013411	0
18	ГЛХУ Дрогичинский лесхоз	г. Дрогичин	290013411	0
19	Цех ОДО Березадревстрой	г. Береза	290013415	0
20	ООО"Прамень"	РБ, Брестская обл., г.Береза, ул.Комсомольская, 120	290054699	0
21	ЗАО"УазСервис"	РБ, Брестская обл., г.Береза, ул.Ленина, 109	290020533	0
22	ООО"Вариант"	РБ,Брестская обл.,г.Береза,ул.Кольцевая,102	290012291	0
23	ОАО"Березовский МРЗ"	РБ,Брестская обл.,г.Береза,ул.Октябрьская,23	290013999	0
24	ОАО"Березовский МСЗ"	РБ,Брестская обл.,г.Береза,ул.Любарского,50	290020840	0
25	ООО "Прополис"	РБ, Брестская обл.,г.Береза,ул.Ленина,66	290010365	0

Рисунок 3.14 – Работа со справочником Контрагентов



Код	Наименование	Адрес	НомерРегистрации	Скидка
1	ОАО"БелАПБ"	РБ Брестская обл., г.Береза, ул.Ленина 106	200011102	0
3	ОДО БерезаСтрой	РБ Брестская обл., г.Береза, ул.Ленина 66/4	290016425	0
4	ОАО"Автопарк №8 г.Береза"	РБ Брестская обл., г.Береза, ул.Ленина 1	220014420	0
5	ОАО"АТП-2 г.Береза"	РБ Брестская обл., г.Береза, ул.Ленина 11	200024055	0
6	ОАО "Березамежрайгаз"	РБ Брестская обл., г.Береза, ул.Ленина 26	200016425	0
7	ИП"Барнюк Г.Ф."	РБ Брестская обл., г.Береза, ул.Строителей 34А	290016425	0
9	Грибовская С.М.	г. Береза, ул. Ленина	АВ3094748	0
10	Гладковская С.П.	г. Береза, ул. Восточная	АВ3094847	0
11	Сирисько О.К.	д. Пески	АВ3154847	0
12	Зубель В.В.	г.Береза	АВ3762115	0
13	ОДО Березадревстрой	РБ Брестская обл., г.Береза, ул.Ленина 66/22	294447652	0
14	Микитич О.Н.	г.Прижаны, ул. Кобринская 2/19	294447652	0
15	ЧПУП Березовский КПП	г.Береза, ул. Красноармейская д. 11	294447656	0
16	Ильючик С.П.	г.Береза, ул. Горина-Коляды 1/27	294447670	0
17	ГПХУ Ивацевичский лесхоз	г. Ивацевичи	290013411	0
18	ГПХУ Дрогичинский лесхоз	г. Дрогичин	290013411	0
19	Цех ОДО Березадревстрой	г. Береза	290013415	0
20	ООО"Прамень"	РБ, Брестская обл., г.Береза, ул.Комсомольская, 120	290054699	0
21	ЗАО"УазСервис"	РБ, Брестская обл., г.Береза, ул.Ленина, 109	290020533	0
22	ООО"Вариант"	РБ,Брестская обл.,г.Береза,ул.Кольцевая,102	290012291	0
23	ОАО"Березовский МРЗ"	РБ,Брестская обл.,г.Береза,ул.Октябрьская,23	290013999	0
24	ОАО"Березовский МСЗ"	РБ,Брестская обл.,г.Береза,ул.Любарского,50	290020840	0
25	ООО "Прополис"	РБ, Брестская обл.,г.Береза,ул.Ленина,66	290010365	0

Рисунок 3.15 – Работа со справочником Контрагентов

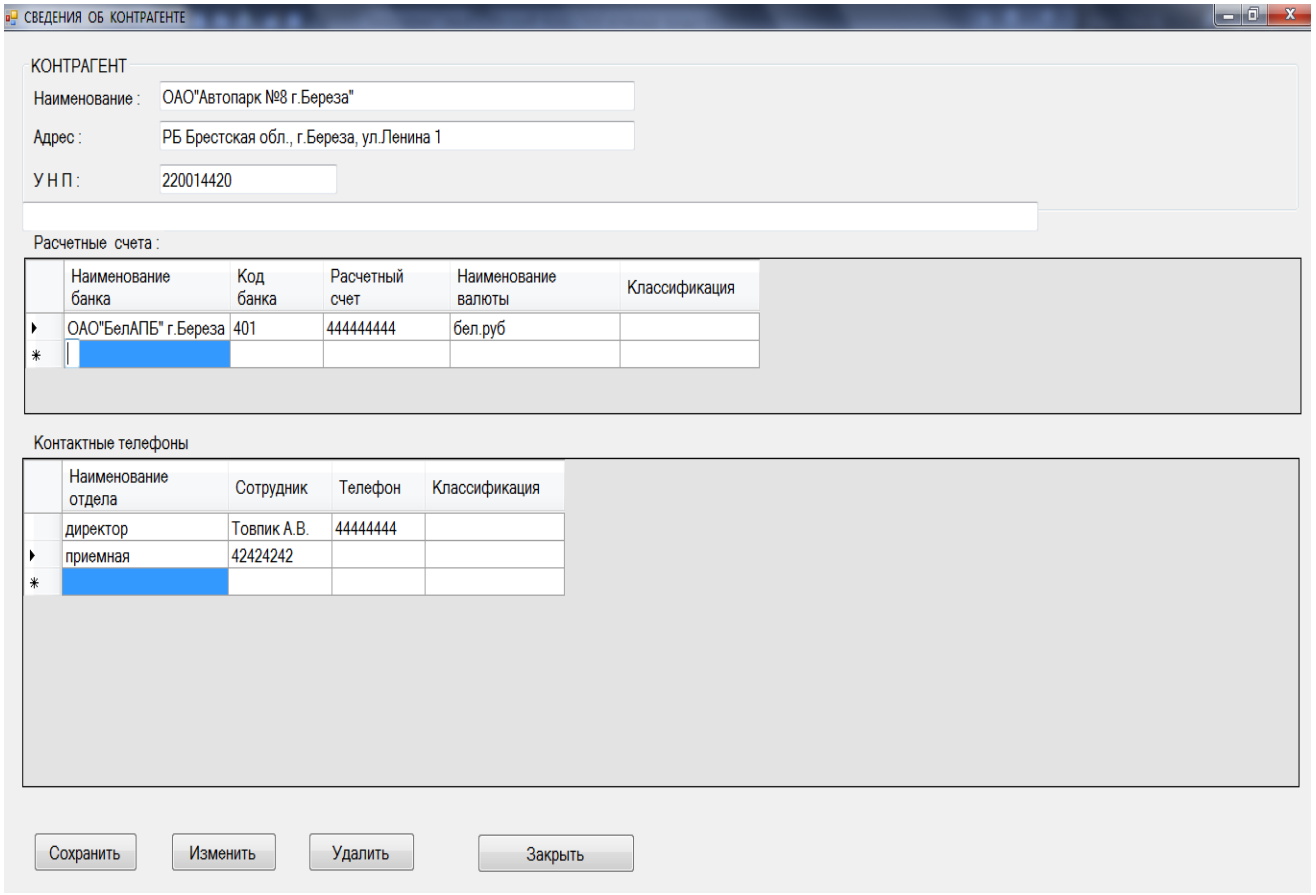


Рисунок 3.16 – Работа со справочником Контрагентов

Нажмем кнопку «СПРАВОЧНИКИ» - «Отделы», в открывшемся окне вводим данные и нажимаем кнопку «Сохранить», результат представим на рисунке 3.17.

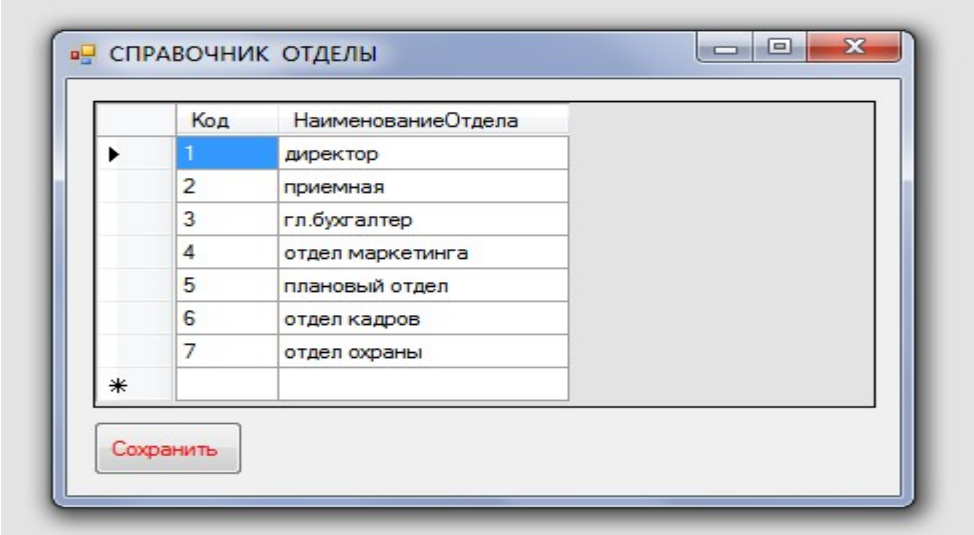


Рисунок 3.17 – Внесение данных в справочник "Отделы"

Нажмем кнопку «ДОКУМЕНТЫ» - «Список договоров» - «Договор продажи товара», в открывшемся окне списка договоров выделяем строку и щелкаем по ней два раза, откроется окно для просмотра данного договора в документе Word. В открывшемся окне списка мы можем изменить, или удалить, или сохранить данные данного договора нажатием соответствующих кнопок вверху на панели кнопок, результаты представлены на рисунках 3.18-3.19.

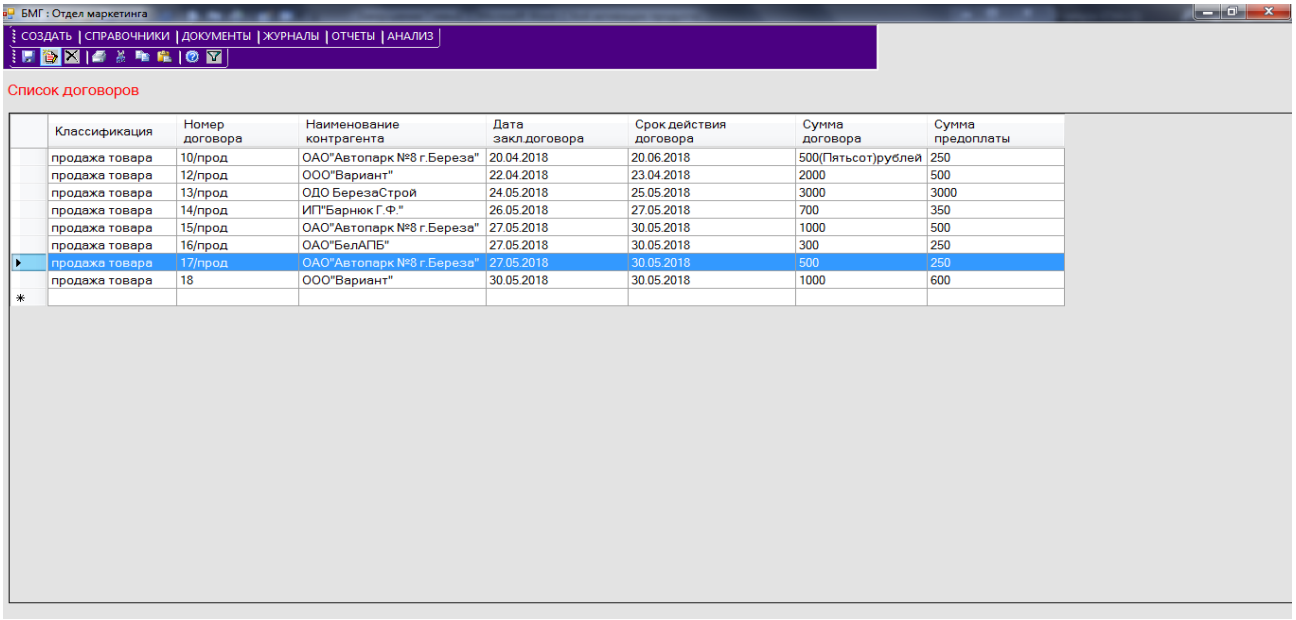


Рисунок 3.18 – Работа с Договорами

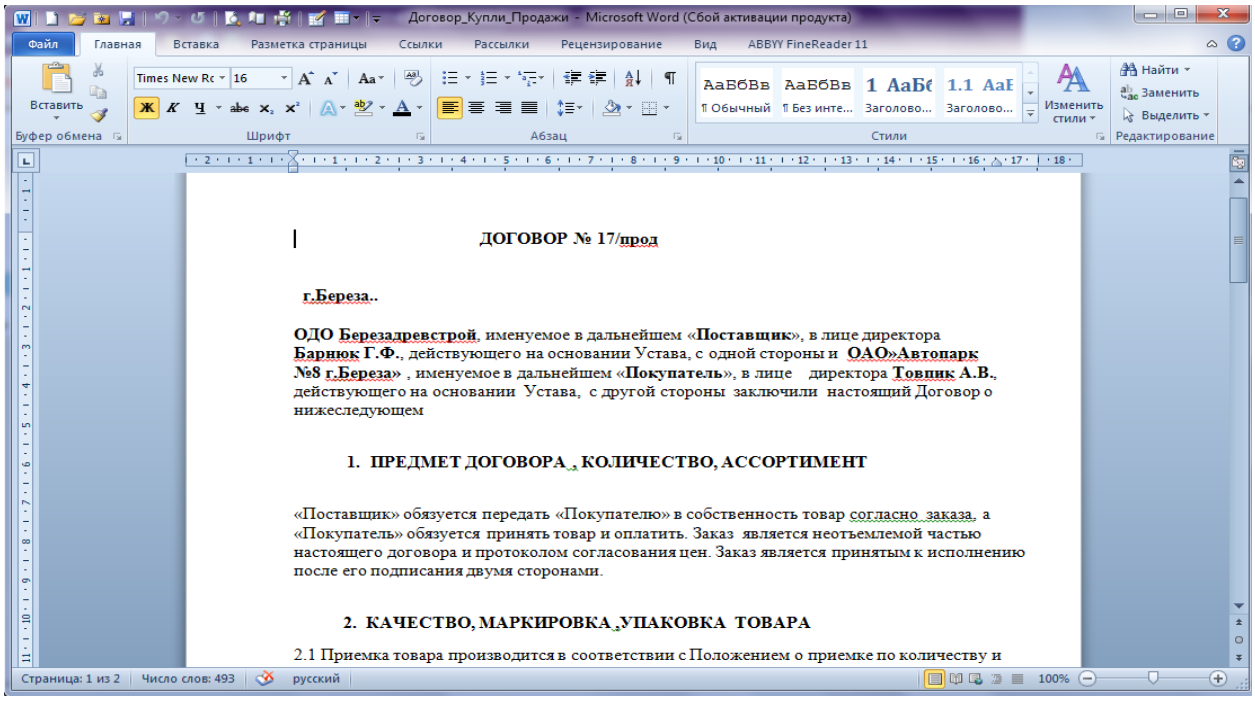
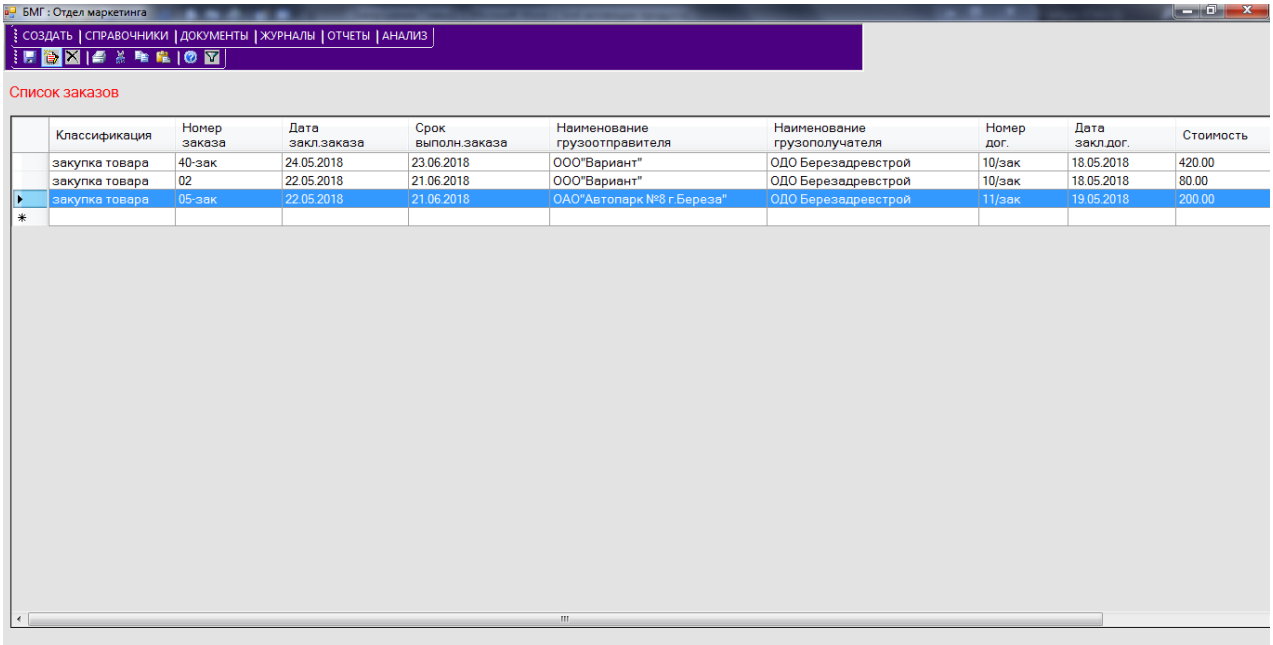


Рисунок 3.19 – Работа с Договорами

Нажмем кнопку «ДОКУМЕНТЫ» - «Список заказов» - «Заказ на продажу товара», в открывшемся окне списка заказов выделяем строку и щелкаем по ней два раза , откроется окно для просмотра данного заказа. В открывшемся окне мы можем не только просмотреть, но и изменить, или удалить, или сохранить данные данного заказа или удалить его полностью нажатием соответствующих кнопок, результаты представлены на рисунках 3.20-3.21.



Классификация	Номер заказа	Дата закл.заказа	Срок выполн.заказа	Наименование грузоотправителя	Наименование грузополучателя	Номер дог.	Дата закл.дог.	Стоимость
закупка товара	40-зак	24.05.2018	23.06.2018	ООО"Вариант"	ОДО Березадревстрой	10/зак	18.05.2018	420.00
закупка товара	02	22.05.2018	21.06.2018	ООО"Вариант"	ОДО Березадревстрой	10/зак	18.05.2018	80.00
закупка товара	05-зак	22.05.2018	21.06.2018	ОАО"Автопарк №8 г.Береза"	ОДО Березадревстрой	11/зак	19.05.2018	200.00

Рисунок 3.20 – Работа с заказами на продажу товара

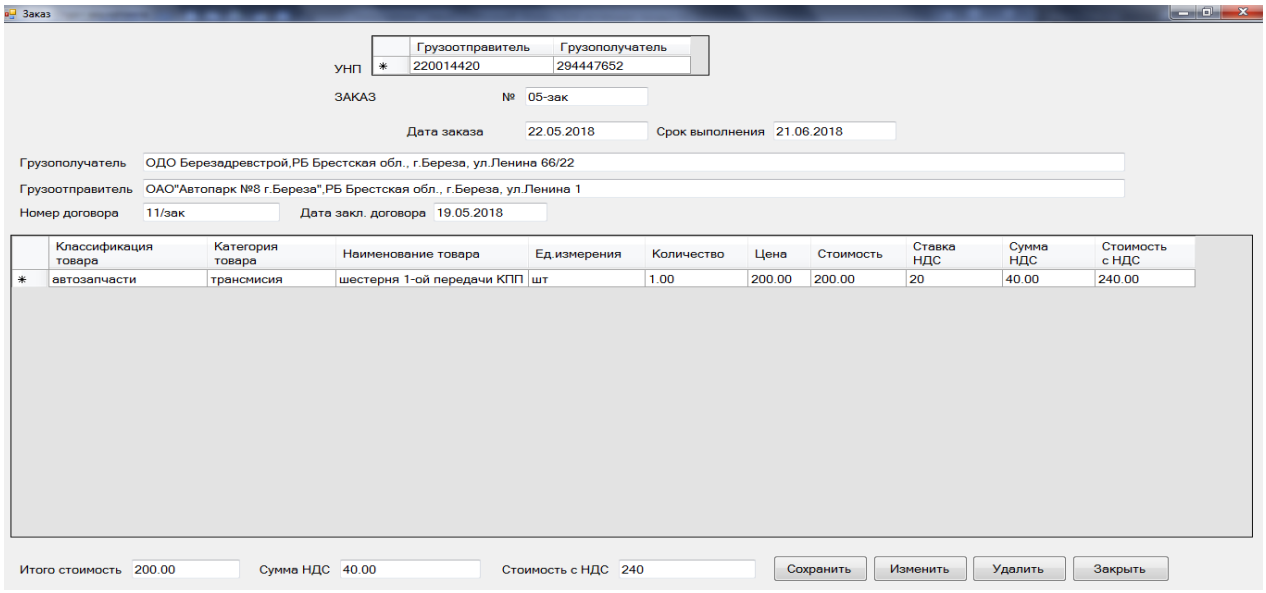


Рисунок 3.21 – Работа с заказами на продажу товара

Нажмем кнопку «ДОКУМЕНТЫ» - «Список накладных» - «Накладные продажи товара», в открывшемся окне списка товаров выделяем строку и щелкаем по ней два раза, откроется окно для просмотра данной накладной. В открывшемся окне мы можем не только просмотреть, но и изменить, или удалить, или сохранить данные данной накладной или удалить ее полностью нажатием соответствующих кнопок, результаты представлены на рисунках 3.22-3.23.

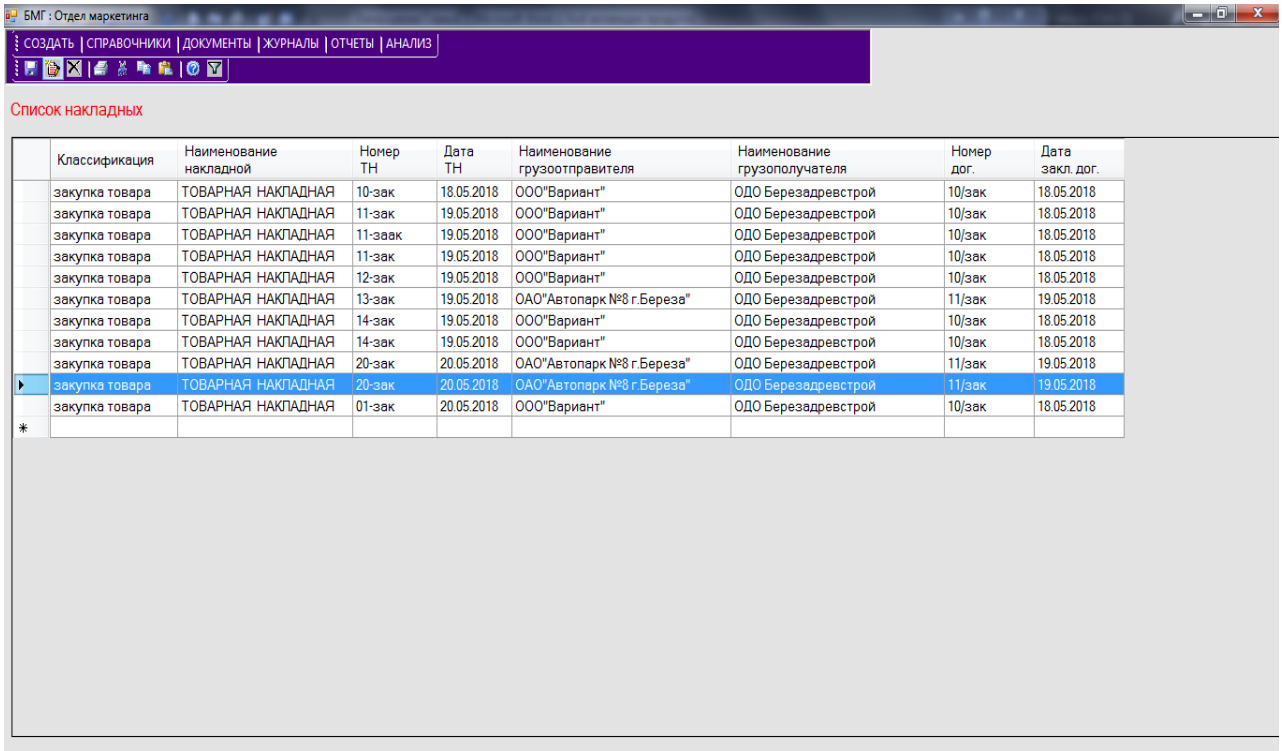


Рисунок 3.22 – Выбор Накладной

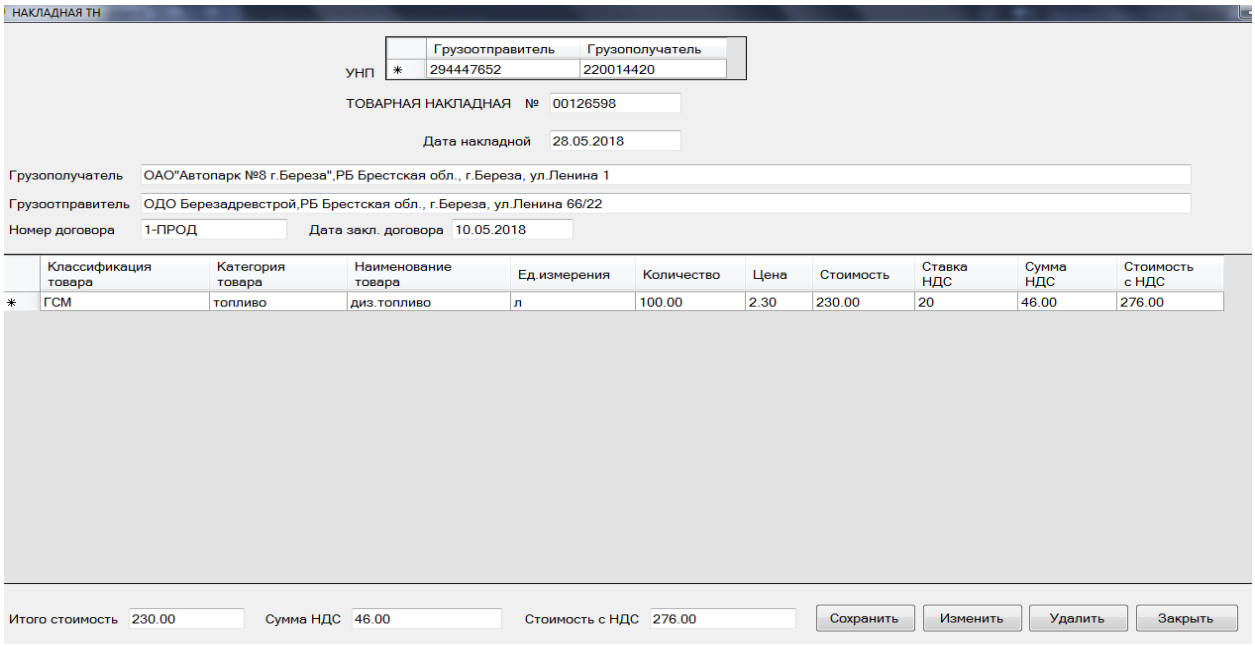


Рисунок 3.23 – Результат выбора Накладной

Нажмем кнопку «ДОКУМЕНТЫ» - «Список счет-фактур» - «Счет-фактура продажи товара», в открывшемся окне списка счет-фактур выделяем строку и щелкаем по ней два раза, откроется окно для просмотра данной счет-фактуры. В открывшемся окне мы можем не только просмотреть, но и изменить, или удалить, или сохранить данные данной счет-фактуры или удалить ее полностью нажатием соответствующих кнопок, результаты представлены на рисунках 3.24-3.25.

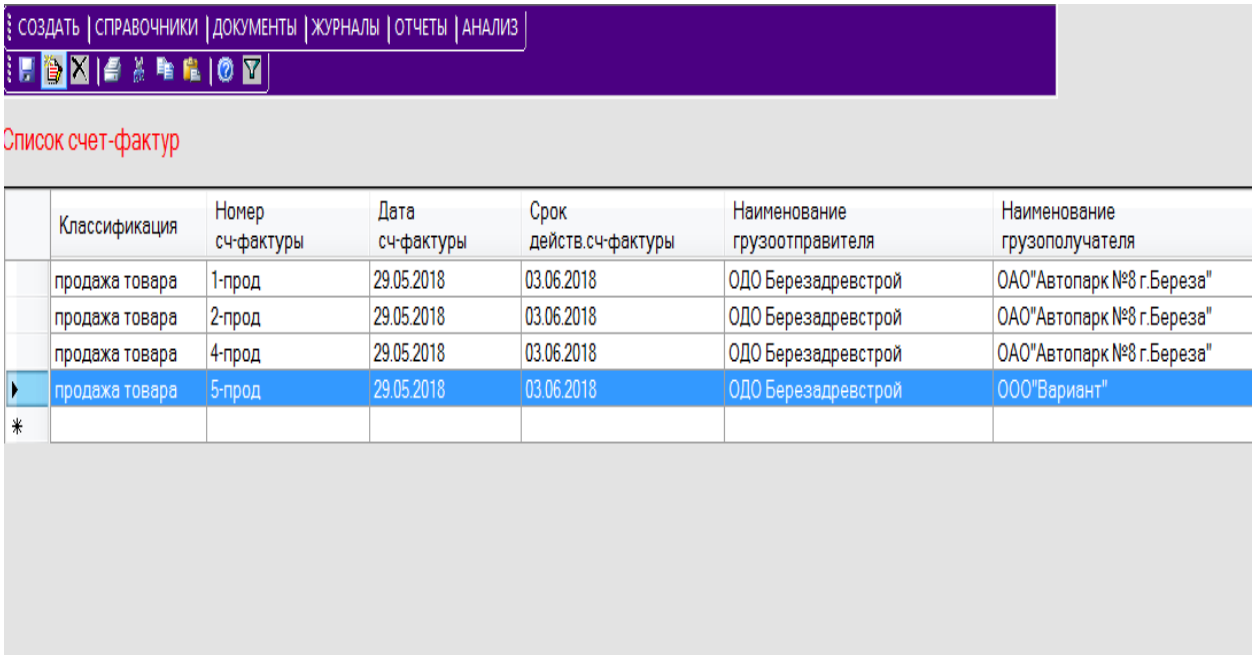


Рисунок 3.24 – Выбор счет-фактуры

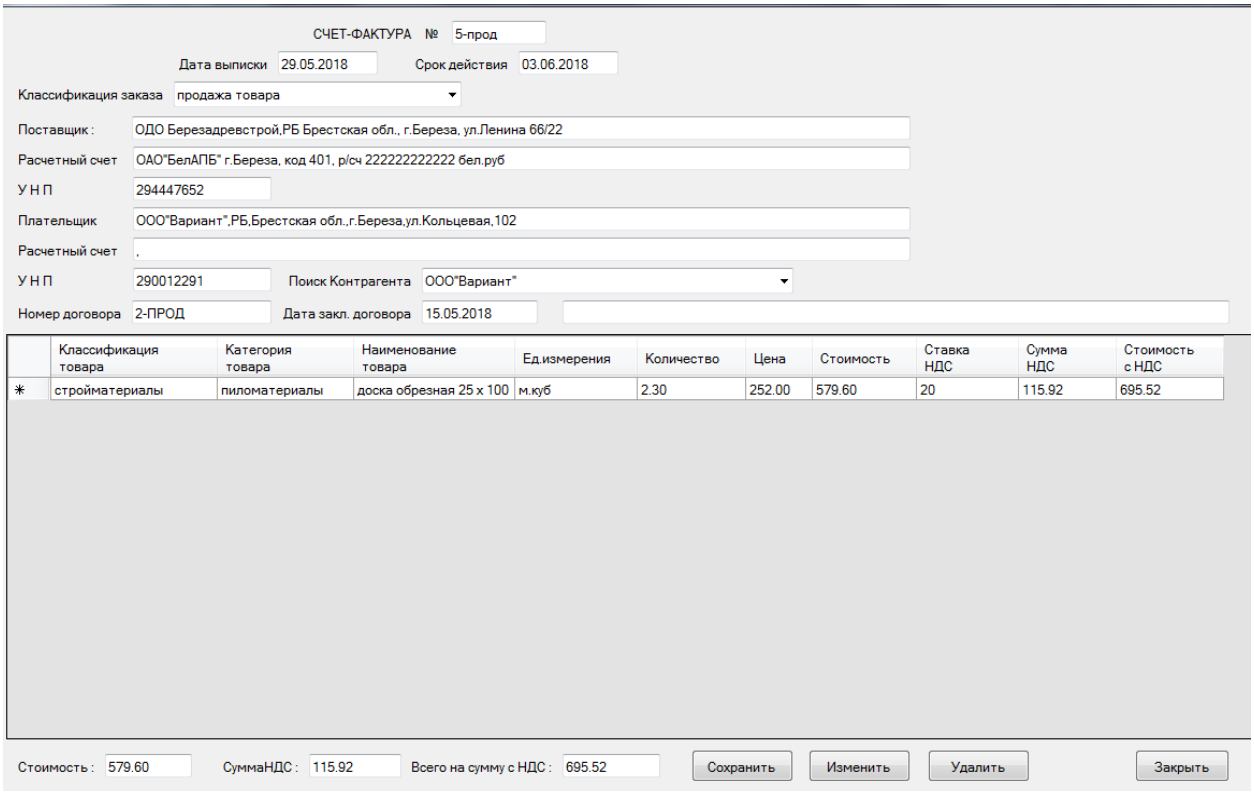


Рисунок 3.25 – Результат выбора счет-фактуры

Таким образом, протестировав основные режимы работы программы, можно сделать вывод о соответствии функционала разработанного ПО поставленным задачам.

3.6 Выводы по главе

В рамках выполнение проектной части работы были определены сущности информационной системы по учету ведения торгово-закупочной деятельности и ее анализа, установлены связи между ними, далее были определены атрибуты, необходимые для решения поставленных задач автоматизации. Было произведено проектирование логической модели данных и структура таблиц СУБД Microsoft SQL Server. На основании поставленных задач автоматизации была проведена разработка программы для автоматизации учета и анализа торгово-закупочной деятельности, в которой реализованы поставленные задачи автоматизации.

ГЛАВА 4 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

Исходные данные для осуществления расчета.

В данном дипломном проекте разрабатывается программное приложение для автоматизации документооборота, учета и анализа торгово-закупочной деятельности предприятия.

Разработка системы предусматривает проведение всех стадий проектирования (техническое задание, эскизный проект, технический проект, рабочий проект, внедрение).

Расчеты будут произведены в следующем порядке:

– Расчет объема функций программного модуля;
– Расчет полной себестоимости;
– Расчет цены и прибыли программного продукта.

Для расчета объема функций программного модуля на разработку является объем ПО. Общий объем (V_0) программного продукта определяется исходя из количества и объема функций, реализуемых программой по формуле (4.1).

$$V_O = \sum_{i=1}^{n} V_i, \quad (4.1)$$

где V_i – объем отдельной функции ПО;

n – общее число функций.

В качестве единицы измерения объема ПО будем использовать строки исходного кода (LOC). Среда разработки ПО – Visual C# (Microsoft). Система управления реляционными базами данных – SQL Server 2008 R2.

Таблица 4.1 – Перечень и объем функций программного модуля

Номер функции	Содержание функций	Объем функции, LOC	
		по каталогу ($V_{i)}$	уточненный (V_{yi})
101	Организация ввода информации	130	80
201	Генерация структуры базы данных	3500	450
202	Формирование баз данных	1980	1500
203	Обработка наборов и записей базы данных	2370	1970
207	Организация поиска и поиск в базе данных	4720	3100
506	Обработка ошибочных ситуаций	1540	940

Продолжение таблицы 4.1

507	Обеспечение интерфейса между компонентами	1680	1300
703	Расчет показателей	420	50
705	Предварительная обработка и вывод на внешние носители	470	155
Итого:		16810	9545

Таким образом, получаем общий объем ПО V_O = 16810LOC, а уточненный $V_У$ = 9545 LOC.

Расчет полной себестоимости.

Расчет основной и дополнительной заработной платы разработчиков. В создании системы будут принимать участие руководитель и программист. Заработная плата разработчика определяется как сумма основной и дополнительной заработной платы всех исполнителей.

Основная заработная плата определяется на основании разряда, тарифной ставки и отработанного времени (см. таблицу 4.2).

Основная заработная плата каждого исполнителя определяется по формуле (4.2):

$$ЗП_{ОСН} = T_{СТ1Р} \cdot T_К / 22 \cdot Ф_{РВ} \cdot K_{ПР}, \quad (4.2)$$

где $T^{СТ1Р}$ – месячная тарифная ставка 1-го разряда ($T^{М1}$ = 34,00руб.);

Тк – тарифный коэффициент, соответствующий установленному тарифному разряду;

22 – среднее количество рабочих дней в месяце;

$Ф_{РВ}$ – фонд рабочего времени исполнителя (продолжительность разработки ПП, дни);

$K_{ПР}$ – коэффициент премирования ($K_{ПР}$ = 1,4).

С 1 марта 2018 года тарифная ставка первого разряда составляет 34,00 рублей.

Количество рабочих дней в месяце – 20, коэффициент премирования – 1,4. Дополнительная заработная плата – 15% от основной заработной платы.

Дополнительная заработная плата включает в себя выплаты, предусмотренные законодательством о труде (оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью работников).

Дополнительная заработная каждого исполнителя ($Н_{доп\ зп}$) – составит 15 %: рассчитывается от основной заработной платы по формуле (4.3):

$$ЗП_{ДОП} = \frac{ЗП_{ОСН} \cdot Н_{ДОП.ЗП}}{100}, \quad (4.3)$$

Результаты вычислений занесем в таблицу 4.2

Таблица 4.2 – Расчет заработной платы

	Категории работников	Разряд	Тарифный коэффициент	Фонд рабочего времени, дни	Коэффициент премирования	Норматив дополнительной зарплаты, %	Заработная плата, руб		
							Основная	Дополнительная	Всего
1 1	Руководитель проекта	15	3,48	30	1,4	15	248,47	37,27	285,74
2 2	Программист без категории	10	2,48	60	1,4	15	354,14	53,12	407,26
Итого							602,61	90,39	693,00

Основная заработная плата руководителя:

$$34{,}00 \times \frac{3{,}48}{20} \times 30 \times 1{,}4 = 248{,}47 \quad руб.$$

Дополнительная заработная плата руководителя:

$$248{,}47 \times \frac{15}{100} = 37{,}27 \quad руб.$$

Основная заработная плата программиста:

$$34{,}00 \times \frac{2{,}48}{20} \times 60 \times 1{,}4 = 354{,}14 \quad руб.$$

Дополнительная заработная плата программиста:

$$354{,}14 \times \frac{15}{100} = 54{,}12 \ \ руб.$$

Отчисления на социальные нужды ($P^{СОЦ}$) определяются в соответствии с действующими законодательными актами по нормативу в процентном отношении к фонду основной и дополнительной заработной платы исполнителей. Сумма отчислений в фонд социальной защиты населения ($P^{СОЦ}$) определяется по формуле 4.4:

$$P_{СОЦ} = \frac{(ЗП_{ОСН} + ЗП_{ДОП}) \cdot Н_{СЗ}}{100}, \qquad (4.4)$$

где $Н_{СЗ}$ – норматив отчислений на социальные нужды (34 % - отчисления в ФСЗН +0,6 % отчисления по обязательному страхованию; $Н_{СЗ}$ = 34,6%).

Таким образом, из формулы (4.4) получаем:

$$P_{СОЦ} = \frac{(602{,}61 + 90{,}39) \cdot 34{,}6}{100} = 239{,}78 \ (руб.)$$

Расходы по статье «Спецоборудование» ($P^{С}$) отсутствуют, так как для разработки конкретного программного продукта специальное оборудование или специальные программы не приобретались, то $P^{С} = 0$.

Расходы по статье «Материалы и комплектующие изделия» ($P_М$) отсутствуют, т. к. не расходовались и не использовались никакие материалы и комплектующие, то $P_М$=0

Расходы по статье «Машинное время» ($P^{М}$) включают оплату машинного времени, необходимого для разработки и отладки ПО. Они определяются в машино-часах по нормативам на 100 строк исходного кода машинного времени в зависимости от характера решаемых задач и типа ПП и определяются по формуле 4.5:

$$P_{МВ} = \frac{Ц_{Мi} \cdot V_О \cdot Н_{МВ}}{100}, \qquad (4.5)$$

где $Ц_{Mi}$ – цена одного машино-часа. Примем $Ц_{Mi}$ = 0,50 руб.;

V_O – уточненный общий объём функций строк исходного кода (LOC);

H_{MB}– норматив расхода машинного времени на отладку 100 строк исходного кода, машино-часов. Примем H_{MB} = 0,6 машино-часов.

Рассчитаем расходы по статье «Машинное время» по формуле (4.5):

$$P_{MB} = \frac{0,50 \cdot 9545 \cdot 0,6}{100} = 28,63 (руб.)$$

Расходы на научные командировки ($P_{НК}$) отсутствуют.

$$P_{НК} = 0 руб.$$

Расходы по статье «Прочие затраты» ($P_{ПР}$) отсутствуют.

$$P_{ПР} = 0 руб.$$

Затраты по статье «Накладные расходы» ($P_{НР}$) отсутствуют.

$$P_{НР} = 0 руб.$$

Сумма вышеперечисленных расходов по статьям на ПП служит исходной базой для расчета затрат на освоение и сопровождение ПП и вычисляется формуле 4.6 :

$$C = ЗП_O + ЗП_Д + P_{СОЦ} + P_M + P_C + P_{MB} + P_{НК} + P_{ПР} + P_{НР} \quad (4.6)$$

Рассчитаем сумму затрат по вышеперечисленным статьям расходов на разработку программного продукта по формуле (4.6):

C = 602,61 + 90,39 + 239,78 + 0 + 0 + 28,63 + 0 + 0 + 0 = 961,41 (руб.)

Затраты на освоение ПО (РО) определяются по формуле 4.7:

$$P_O = \frac{C \cdot H_O}{100\%}, \quad (4.7)$$

где H_O – норматив расходов на освоение, %. Примем, H_O = 5%.

Подставив данные в (4.7) получим:

$$P_O = \frac{961,41 \cdot 5\%}{100} = 48,07 (руб.)$$

Затраты на сопровождение ПО (РСО) вычисляются по формуле 4.8:

$$P_{CO} = \frac{C \cdot H_C}{100\%}, \quad (4.8)$$

где H_C – норматив расходов на сопровождение, %. Возьмем, H_C = 5%.

$$P_{CO} = \frac{961,41 \cdot 5\%}{100} = 48,07 (руб.)$$

Полную себестоимость программного продукта по смете (СП) можно определить по формуле 4.9:

$$C_П = C + P_O + P_{CO} \quad (4.9)$$

Рассчитаем полную себестоимость ПО по формуле (4.9):

$$C_П = 8961,41 + 48,07 + 48,07 = 1057,55 (руб.)$$

Научные командировки не осуществлялись.
Прочие затраты отсутствуют.
Накладные расходы отсутствуют.
Затраты на освоение определены в размере 5% от всех затрат.

Затраты на сопровождение определены в размере 5% от всех затрат.

Приведенные выше расчеты представим в таблице 4.3.

Таблица 4.3 – Расчет полной себестоимости программного продукта

№ п/п	Наименование статей затрат	Норматив	Сумма затрат, руб.
	A	B	C
1	Заработная плата всего	-	693,00
1.1	Основная ЗП	-	602,61
1.2	Дополнительная ЗП	-	90,39
2	Отчисления на социальные нужды	34,6 %	239,78
3	Спецоборудование	-	0
4	Материалы	-	0
5	Машинное время	-	28,63
6	Научные командировки	-	0
7	Прочие затраты	-	0
8	Накладные расходы	-	0
9	Сумма расходов	-	961,41
10	Затраты на освоение и сопровождение	10 %	96,14
11	Полная себестоимость	-	1057,65

Расчет цены и прибыли программного продукта.

Для определения цены ПП необходимо рассчитать плановую прибыль.

Прибыль рассчитывается по формуле 4.10:

$$П = \frac{C_П \cdot R}{100}, \quad (4.10)$$

где $П$ – плановая прибыль от реализации ПО, руб.;

R– уровень рентабельности ПП, R = 10%.

Рентабельность и прибыль создаваемого ПП определяется исходя из результатов анализа рыночных условий, переговоров с заказчиком и согласования с ним отпускной цены.

Рассчитаем плановую прибыль по формуле(4.10):

$$П = \frac{1057,65 \cdot 10}{100} = 105,77 (руб.)$$

После расчета прибыли от реализации определяется прогнозируемая цена ПП без налогов. Прогнозируемая цена без налогов ($Ц^{П}$) рассчитывается по формуле:

$$Ц_{П} = С_{П} + П \quad (4.11)$$

Из формулы (4.11) получаем:

$$Ц_{П} = 1057,65 + 105,77 = 1163,42 (руб.)$$

Отпускная цена (цена реализации) ПП включает налог на добавленную стоимость (*НДС*) и определяется по формуле 4.12:

$$Ц_{О} = С_{П} + П + НДС \quad (4.12)$$

Налог на добавленную стоимость (*НДС*) рассчитывается по формуле 4.13:

$$НДС = \frac{Ц_{П} \cdot Н_{ДС}}{100}, \quad (4.13)$$

где $Н_{ДС}$ – норматив налога на добавленную стоимость ($Н_{ДС}$ = 20%).

$$НДС = \frac{1163,42 \cdot 20}{100} = 232,68 (руб.)$$

Таким образом, из формулы (4.12) получаем отпускную цену ПП:

$$Ц_{О} = 1057,65 + 105,77 + 232,68 = 1396,10 (руб.)$$

Прибыль от реализации ПП за вычетом налога на прибыль ($П_ч$) является чистой прибылью, остается организации разработчику и представляет собой экономический эффект от создания нового программного продукта. Чистая прибыль рассчитывается по формуле 4.14:

$$П_ч = П \cdot \left(1 - \frac{Н_П}{100}\right), \quad (4.14)$$

где $Н_П$ – ставка налога на прибыль (в настоящее время $Н_П$ = 18%).

Рассчитаем чистую прибыль по формуле (4.14):

$$П_ч = 105,77 \cdot \left(1 - \frac{18}{100}\right) = 86,73 (руб.)$$

Далее рассчитаем на сколько создание приложение было эффективным по экономическим данным.

Для подсчета итоговых данных по полной себестоимости, прибыли, цены, НДС, отпускной цены, а также чистой прибыли при проектировании приложения, создаваемого в рамках данного дипломного проекта, приведенные выше расчеты представим в таблице 4.4.

Таблица 4.4 – Отпускная цена ПП и чистая прибыль

Наименование статей затрат	Норматив	Сумма затрат, руб.
1. Полная себестоимость	-	1057,65
2. Прибыль	10%	105,77
3. Цена без НДС	-	1163,42
4. НДС	20%	232,68
5. Отпускная цена	-	1396,10
6. Чистая прибыль	-	86,73

Определены основные экономические показатели:

Полная себестоимость – 1057,65 бел, руб.

Отпускная цена – 1396,10 бел, руб.

Чистая прибыль – 86,73 бел, руб.

Таким образом рассчитанная отпускная цена на программный продукт, разрабатываемый в рамках этого дипломного проекта, является конкурентно

способной. Согласно различным источникам текущая рыночная цена на подобный программный продукт составляет порядка 1600,00 бел руб.

Таким образом программный продукт будет экономически целесообразен.

ГЛАВА 5 ОХРАНА ТРУДА

Создание благоприятных условий труда является одной из важнейших государственных задач. В понятие охрана труда входят трудовое законодательство, производственная санитария и техника безопасности.

Трудовое законодательство - правовые и организационные мероприятия, определяющие трудовые взаимоотношения на производстве, режим рабочего времени и отдыха, условия труда женщин и подростков, порядок приема на работу и увольнения [23,с.16].

Производственная санитария предусматривает осуществление санитарно-гигиенических норм и положений, оборудование бытовых помещений на предприятиях, борьбу с шумом, вибрацией, загрязненностью, запыленностью, высокой температурой на рабочих местах и в производственных помещениях.

Техника безопасности - система организационно- технических мероприятий и средств, предотвращающих воздействие на работающих опасных производственных факторов.

В прямой связи с охраной труда и техникой безопасности находится противопожарная безопасность. Противопожарная профилактика включает комплекс мероприятий, направленных на предупреждение и ликвидацию пожаров и обеспечивающих защиту людей и материальных ценностей от огня.

На всех предприятиях разрабатываются и реализуются комплексные планы улучшения условий, охраны труда и санитарно-оздоровительных мероприятий. Внедряется паспортизация санитарно-гигиенических условий труда, являющаяся основой комплексного планирования и организации работ по улучшению охраны труда на производстве.

Правительство Республики Беларусь постоянно проводит большую работу по охране и улучшению здоровья трудящихся.

В Конституции РБ, в основах законодательства РБ о труде установлено, что охрана здоровья трудящихся на производстве - одна из главных задач государства, а право трудящихся на охрану здоровья обеспечивается постоянным развитием и совершенствованием техники безопасности и производственной санитарии.

Принципиально важное значение для совершенствования организации охраны труда имеет ратификация 3 мая 1999 г. Республикой Беларусь Конвенции 155 Международной организации труда «О безопасности и гигиене труда в производственной среде». Основной целью ратификации Конвенции является признание необходимости разработки на государственном уровне, с учетом международного опыта, национальной политики в области охраны

труда и создание необходимых правовых и организационных механизмов ее реализации как на национальном уровне, так и на уровне предприятий, организаций.

В соответствии с положениями Конвенции разработка и реализация государственной политики предполагает принятие на национальном уровне государственной системы управления охраной труда на современном этапе.

Важным этапом в совершенствовании государственного управления охраной труда стало принятие Президентом Республики Беларусь Директивы от 11 марта 2004 г. № 1 «О мерах по укреплению общественной безопасности и дисциплины». В соответствии с Директивой 8 внесены изменения и дополнения в нормативные правовые акты по вопросам заключения и расторжения контрактов.

Показатели состояния трудовой и исполнительской дисциплины, безопасности труда определены в качестве важнейших критериев оценки работы руководителей всех уровней. Директива способствует повышению ответственности субъектов государственного управления, нанимателей, профсоюзов и работников за соблюдение требований охраны труда.

Министерством труда и социальной защиты Республики Беларусь с участием других заинтересованных органов разработана Концепция государственного управления охраной труда в Республике Беларусь, которая одобрена Постановлением Совета Министров Республики Беларусь от 16 августа 2005 г. № 904. Концепция определяет, что целью государственного управления охраной труда является создание условий, обеспечивающих сохранение жизни и здоровья граждан в процессе трудовой деятельности.

Основными принципами государственной политики в области охраны труда являются:

• приоритет жизни и здоровья работников по отношению к результатам производственной деятельности;

• обеспечение гарантий права работников на охрану труда;

• установление обязанностей всех субъектов правовых отношений в области охраны труда, полной ответственности нанимателей за обеспечение здоровых и безопасных условий труда;

• совершенствование правовых отношений и управления в этой сфере, включая внедрение экономического механизма обеспечения охраны труда.

Основные направления государственной политики в области охраны труда:

• разработка и принятие законодательных и иных нормативных правовых актов по охране труда, технических нормативных правовых актов, содержащих требования охраны труда;

• разработка и реализация целевых программ по улучшению условий и охраны труда;

• создание систем управления охраной труда на всех уровнях, обеспечивающих профилактическую направленность деятельности в этой сфере;

• разработка научно обоснованных методов оценок и прогнозирования рисков гибели и травмирования работников по отраслям и сферам деятельности;

• экономическое стимулирование создания безопасных условий труда, разработки и внедрения безопасных техники и технологий, производства средств индивидуальной и коллективной защиты работников;

• упорядочение предоставления компенсаций по условиям труда;

• организация научно-исследовательских работ по вопросам безопасности и гигиены труда;

• обучение и повышение квалификации работников по вопросам охраны труда, подготовка специалистов по охране труда;

• повышение ответственности работников за соблюдение требований охраны труда;

• обеспечение законных интересов потерпевших от несчастных случаев на производстве и профессиональных заболеваний, а также членов их семей на основе обязательного страхования от несчастных случаев на производстве и профессиональных заболеваний;

• финансовое обеспечение охраны труда;

• создание условий для социального партнерства в сфере охраны труда, содействие общественному контролю за соблюдением законодательства об охране труда;

• распространение передового опыта работы по улучшению условий и охраны труда;

• международное сотрудничество в области охраны труда.

Государственное управление охраной труда реализуется на республиканском, отраслевом, региональном уровнях.

На республиканском уровне государственное управление охраной труда осуществляет правительство Республики Беларусь непосредственно или уполномоченные им министерства, другие республиканские органы государственного управления, объединения (учреждения), подчиненные правительству Республики Беларусь, которым законодательством предоставлено право осуществлять соответствующие функции нормативного правового регулирования, специальные разрешительные, надзорные и контрольные полномочия в области охраны труда и связанных с ней отношений.

На отраслевом уровне государственное управление охраной труда осуществляют министерства, другие республиканские органы государственного управления, объединения (учреждения), подчиненные правительству Республики Беларусь, имеющие отраслевую направленность и подведомственные организации.

На региональном уровне государственное управление охраной труда осуществляют местные исполнительные и распорядительные органы.

В систему законодательных актов, регулирующих вопросы охраны труда, входят: Конституция Республики Беларусь, Трудовой кодекс Республики Беларусь, Законы Республики Беларусь «Об охране труда», «О техническом нормировании и стандартизации», «О промышленной безопасности опасных производственных объектов», «Об основах государственного социального страхования», «О пенсионном обеспечении», «О нормативных правовых актах Республики Беларусь», «О санитарно-эпидемическом благополучии населения», «О здравоохранении», «О радиационной безопасности населения», «О защите населения и территорий от чрезвычайных ситуаций природного и техногенного характера», «О пожарной безопасности» и др.

Конституция Республики Беларусь является правовой основой организации работ по охране труда. В ст. 41 и 45 гарантируются права граждан на здоровье и безопасные условия труда, охрану их здоровья.

Трудовой кодекс Республики Беларусь (ТК) определяет и регулирует правоотношения в сферах труда и охраны труда. Наряду с правами работника на здоровье и безопасные условия труда (ст. 11) предусмотрен механизм реализации этого права через обязанность нанимателя обеспечивать такие условия труда (ст. 54, 55, 89, 224, 226, 228–231).

Впервые ст. 227 законодательно регламентирована деятельность службы охраны труда. Для организации работы и осуществления контроля по охране труда наниматели вводят должность специалиста по охране труда или создают службу охраны труда из числа лиц, имеющих необходимую подготовку.

Закон Республики Беларусь «Об охране труда» направлен на регулирование общественных отношений в области охраны труда и реализацию установленного Конституцией Республики Беларусь права граждан на здоровые и безопасные условия труда.

Закон Республики Беларусь «О техническом нормировании и стандартизации» регулирует отношения, возникающие при разработке, утверждении и применении технических требований к продукции, процессам ее разработки, производства, эксплуатации (использования), хранения, перевозки, реализации и утилизации или к оказанию услуг, определяет правовые и

организационные основы технического нормирования и стандартизации и направлен на обеспечение единой государственной политики в этой области.

Закон Республики Беларусь « О промышленной безопасности опасных производственных объектов» определяет правовые, экономические и социальные основы обеспечения безопасной эксплуатации опасных производственных объектов и направлен на предупреждение аварий на опасных производственных объектах и обеспечение готовности организаций, эксплуатирующих опасные производственные объекты, к локализации и ликвидации последствий указанных аварий.

Законом Республики Беларусь «Об основах государственного социального страхования» в рамках общих вопросов страхования граждан предусмотрены вопросы страхования их от несчастных случаев на производстве и профессиональных заболеваний.

Закон Республики Беларусь «О нормативных правовых актах Республики Беларусь» определяет понятие и виды нормативных правовых актов Республики Беларусь, устанавливает общий порядок их подготовки, оформления, принятия (издания), опубликования, действия, толкования и систематизации.

Закон Республики Беларусь «О санитарно-эпидемическом благополучии населения» направлен на предупреждение воздействия неблагоприятных факторов среды обитания на здоровье населения и регламентирует действия по обеспечению санитарно-эпидемического благополучия, устанавливает государственный санитарный надзор за соблюдением санитарных норм и гигиенических нормативов.

Закон Республики Беларусь « О здравоохранении» направлен на обеспечение правовых, организационных, экономических и социальных основ государственного регулирования в области здравоохранения в целях сохранения, укрепления и восстановления здоровья населения.

Закон Республики Беларусь «О радиационной безопасности населения» определяет основы правового регулирования в области обеспечения радиационной безопасности населения, направлен на создание условий, обеспечивающих охрану жизни и здоровья людей от вредного воздействия ионизирующего излучения.

Закон Республики Беларусь «О защите населения и территорий от чрезвычайных ситуаций природного и техногенного характера» регулирует отношения в области защиты населения и территорий от чрезвычайных ситуаций природного и техногенного характера, являющиеся в современных условиях важнейшей частью обеспечения безопасности.

Закон Республики Беларусь «О пожарной безопасности» устанавливает государственный надзор за обеспечением пожарной безопасности

министерствами, комитетами, концернами, предприятиями, учреждениями, организациями, а также принципы деятельности пожарной службы.

Закон Республики Беларусь «О сертификации продукции, работ и услуг» направлен на обеспечение безопасности продукции для жизни, здоровья и имущества населения, а также охраны окружающей среды, определяет национальную систему сертификации.

Помимо законов деятельность по охране труда регулируется директивными документами. Это декреты, указания и распоряжения Президента страны, постановления Правительства Республики Беларусь.

Декрет Президента Республики Беларусь № 18 от 30 июля 2003 г.

«Об обязательном страховании от несчастных случаев на производстве и профессиональных заболеваний» направлен на усиление социальной защиты граждан, потерпевших в результате травматизма на производстве и профессиональных заболеваний. Он регулирует вопросы возмещения причиненного работающим вреда, стимулирует меры по предупреждению производственного травматизма и профессиональных заболеваний.

Постановление Совета Министров Республики Беларусь от 10 февраля 2003 г. № 150 направлено на реализацию государственной политики в области охраны труда и совершенствование организации работы по подготовке и принятию нормативных правовых актов, технических нормативных правовых актов, содержащих государственные нормативные требования охраны труда. Утвержден перечень видов нормативных правовых актов, технических нормативных правовых актов, содержащих требования охраны труда, и республиканских органов государственного управления, принимающих их.

Постановление Совета Министров Республики Беларусь от 16 августа 2005 г. № 905 «О Республиканской целевой программе по улучшению условий и охраны труда на 2006–2010 годы» направлено на улучшение условий и охраны труда, снижение уровня производственного травматизма и профессиональной заболеваемости в организациях республики.

Основные требования производственной санитарии и гигиены труда, техники безопасности, пожарной безопасности регламентируются:

1. Межгосударственными и государственными техническими нормативными правовыми актами (ТНПА). К ним относятся: межгосударственные стандарты (МГОСТ, МЭК, ИСО); государственные стандарты системы стандартов безопасности труда (ГОСТ ССБТ); технические кодексы установившейся практики (ТКП); государственные стандарты Беларуси (СТБ); санитарные правила и нормы (СанПиН); строительные нормы и правила (СНиП); строительные нормы Беларуси (СНБ); нормы радиационной

безопасности (НРБ); нормы пожарной безопасности (НПБ); правила пожарной безопасности (ППБ).

2. Межотраслевыми ТНПА: межотраслевые правила по охране труда (ПОТ М); гигиенические нормативы (ГН); межотраслевые типовые инструкции по охране труда (ТИОТ М); правила технической безопасности (ПТБ); правила устройства (ПУ); правила устройства и безопасной эксплуатации (ПУБЭ); инструкции по безопасности (ИБ); руководящие документы (РД); республиканские допустимые уровни (РДУ).

3. Отраслевыми ТНПА: отраслевые правила по охране труда (ПОТ О); отраслевые типовые инструкции по охране труда (ТИОТ О); руководящие документы (РД).

4. Локальными ТНПА: стандарты предприятия (СТП); соглашения по охране труда; технологические инструкции; правила техники безопасности и производственной санитарии; инструкции по охране труда по профессиям и видам работ.

Межгосударственные, государственные и межотраслевые ТНПА обязательны для исполнения всеми субъектами хозяйственной деятельности и организациями независимо от форм собственности и ведомственного подчинения.

Отраслевые ТНПА распространяются только на отдельную отрасль производства в масштабах всей страны.

Локальные ТНПА разрабатываются для конкретных предприятий и организаций и действуют только на них.

С целью создания единого комплекса взаимоувязанной нормативно-технической документации по обеспечению безопасности труда была разработана система стандартов безопасности труда (ССБТ). В настоящее время в республике применяется более 600 ГОСТов .

В настоящее время основой постоянного улучшения условий и безопасности труда является комплексная автоматизация всех видов работ.

Компьютеризация и автоматизация предприятий стала целью номер один. Использование компьютерных ресурсов способно повысить эффективность, как бизнеса, так и образования. Но вместе с тем возникает ряд проблем, на которые стоит обратить внимание. Ни для кого не секрет, что ПК оказывает вредное влияние на организм человека [23, с.65].

Многие инженеры-программисты и операторы ПК связаны с воздействием таких психофизических факторов, как умственное перенапряжение, перенапряжение зрительных и слуховых анализаторов, монотонность труда, эмоциональные перегрузки. Воздействие указанных неблагоприятных факторов приводит к снижению работоспособности, утомлению, изменениями, возникающими в центральной нервной системе и в коре головного мозга.

Поэтому организацию оптимального рабочего места программиста нужно проводить на основании данных эргономики, антропологии, правил охраны труда и целого ряда других наук. При этом необходимо учитывать требования стандартов и санитарных норм.

Основной причиной, обуславливающей ухудшение работоспособности и здоровья людей, использующих в своей работе персональный компьютер, является сидячий и малоподвижный труд. При малоподвижном образе жизни и отсутствии достаточной нагрузки на мышцы в центральную нервную систему не поступают сигналы от мышц, а значит, головной мозг не дает команды о поступлении в мышцы питательных веществ и кислорода, что сказывается на состоянии мышечных волокон: уменьшатся их объем, сократительная способность и способность длительно работать [23, с.18].

Работа сотрудника отдела маркетинга требует длительного статического напряжения мышц спины, шеи, рук и ног, что ведет к их утомлению и появлению специфических жалоб.

Организация рабочего места с ПК должна учитывать требования безопасности, удобство положения, движений и действий работника. Рабочий стол с учетом характера выполняемой работы должен иметь достаточный размер для рационального размещения монитора (дисплея), клавиатуры, другого используемого оборудования и документов, поверхность, обладающую низкой отражающей способностью. Клавиатура располагается на поверхности стола таким образом, чтобы пространство перед клавиатурой было достаточным для опоры рук работника (на расстоянии не менее чем 100 мм от края, обращенного к работнику), либо на специальной поверхности, отделённой от основной столешницы. Чтобы обеспечивалось удобство зрительного наблюдения, быстрое и точное считывание информации, плоскость экрана монитора располагается ниже уровня глаз работника предпочтительно перпендикулярно к нормальной линии взгляда работника (нормальная линия взгляда - 15 град. вниз от горизонтали). Для исключения воздействия повышенных уровней электромагнитных излучений расстояние между экраном монитора и работником должно составлять не менее 500 мм (оптимальное 600 - 700 мм). Применяемые подвижные подставки для документов (пюпитры) размещаются в одной плоскости и на одной высоте с экраном.

Рабочий стул (кресло) должен быть устойчивым, место сидения должно регулироваться по высоте, а спинка сиденья - по высоте, углам наклона, а также расстоянию спинки от переднего края сиденья. Регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию. Для тех, кому это удобно, предусматривается подставка для ног.

Рабочее место размещается таким образом, чтобы естественный свет падал сбоку (желательно слева). Для снижения яркости в поле зрения при естественном освещении применяются регулируемые жалюзи, плотные шторы.

Для создания нормальных условий освещённости и соответствующего контраста между экраном и окружающей обстановкой с учётом вида работы и требований видимости со стороны работника используются светильники общего и местного освещения. Освещенность на поверхности стола в зоне размещения рабочего документа должна составлять 300-500 люкс. Возможные мешающие отражения и отблески на экране монитора и другом оборудовании устраняются путем соответствующего размещения экрана, оборудования, расположения светильников местного освещения.

Ещё одним фактором, влияющим на самочувствие человека, является уровень фонового шума. Верхний предел фонового шума для офисных помещений составляет 55 дБ. Шум может происходить от различных источников: компьютеры, лампы освещения, уличный шум и т.д. Снижение шума достигается с помощью перегородок из звукоизоляционных и звукопоглощающих материалов. Шумящее оборудование (печатающие устройства, сканеры, серверы и тому подобные), уровни шума которого превышают нормативные, должно размещаться вне рабочих мест сотрудников.

Для обеспечения оптимальных параметров микроклимата проводятся регулярное в течение рабочего дня проветривание и ежедневная влажная уборка помещений. Для создания нормальной температуры (22-24 °С в холодное время года и 23-25 °С в тёплое) помещения оборудованы кондиционерами, для поддержания необходимого уровня влажности используются увлажнители воздуха.

Для обеспечения безопасности работников на соседних рабочих местах расстояние между рабочими столами с мониторами (в направлении тыла поверхности одного монитора и экрана другого монитора) должно быть не менее 2,0 м, а расстояние между боковыми поверхностями мониторов - не менее 1,2 м.

Работу за экраном монитора следует периодически прерывать на регламентированные перерывы, которые устанавливаются для обеспечения работоспособности и сохранения здоровья, или заменять другой работой с целью сокращения рабочей нагрузки у экрана. Время регламентированных перерывов в течение рабочего дня (смены) устанавливается в зависимости от его (ее) продолжительности, вида и категории трудовой деятельности. При этом время непрерывной работы за компьютером между перерывами не должно превышать 2 часов. Во время перерыва с целью снижения нервно-эмоционального напряжения и утомления зрения, предотвращения развития утомления целесообразно выполнять комплексы специальных упражнений.

Каждое помещение в компании оборудовано стендом с информацией по охране труда. Содержание стенда приведено на рисунке 5.1.



Рисунок 5.1– Стенд «Уголок по охране труда», размещаемый в рабочих помещениях

Таким образом, изложенные выше предложения обеспечат безопасные условия труда сотрудника отдела маркетинга нашего предприятия.

ЗАКЛЮЧЕНИЕ

Изучена предметная область деятельности отдела маркетинга. Произведен анализ и описание предметной области, определены необходимые требования к приложению. Произведено технико-экономическое обоснование дипломного проекта.

В результате дипломного проектирования было создано приложение «Отдел маркетинга». Выполнены все необходимые требования согласно целям и задачам. По результатам проектирования системы был разработан интерфейс приложения и реализована логика работы приложения. После реализации приложения было проведено тестирование на его работоспособность.

Приложение «Отдел маркетинга» было реализовано в среде разработки ПО-Microsoft Visual Studio.

Разработано максимально удобное и доступное в понимании и использовании приложение для оформления, учета и анализа совершаемых сделок по продаже и закупке товаров. Все документы имеют общую классификацию, товары разбиты по классификациям и категориям, все это позволило разделить сделки на продажу товаров и закупку товаров. Такое разбитие дало возможность отслеживать выполнение заказов, проводить их анализ и принимать соответствующие решения.

Таким образом, приложение послужит хорошим инструментом для продажи товаров. Система имеет удобный и интуитивно понятный интерфейс, тем самым минимизирует время на освоение программного продукта. Все формы имеют понятный, приближенный к реальным документам, ввод данных. При заполнении таблиц и полей ввода данных, одни заполняются автоматически, другие имеют функцию ввода-подсказки, а для некоторых данных, открываются формы поиска. Все операции сохранения, изменения и удаления сопровождаются диалогом с пользователем, чтобы он не сделал ошибки.

Приложение обладает возможностью расширения его функционала, то есть могут быть добавлены новые функции, новые объекты учета и анализа.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Абдикеев Н.М. Корпоративные информационные системы управления: учебник / Под науч. ред. Н.М. Абдикеева, О.В. Китовой – М.: ИНФРА-М, 2015. – 464 с.

2. Акулич И.Л. Маркетинг: учебник / И.Л. Акулич. – 8-е изд., перераб. и доп. – Минск: Вышэйшая школа, 2014. – 543 с.: ил.

3. Андерсен Б. Бизнес-процессы. Инструменты совершенствования /Пер. с англ. С.В. Ариничева - М.: РИА «Стандарты и качество», 2013. – 272 с.

4.Бен Форта «Освой самостоятельно SQL. 10 минут на урок», “И.Д.Вильямс” 2010.

5. Баркан, Д.И. Управление продажами: Учебник / Д.И. Баркан. – СПб.: Издат. дом С.-Петерб. гос. ун-та, 2013. – 908 с.

6.Браст Эндрю Дж., Форте С Разработка приложений на основе Microsoft SQL Server 2005.,Microsoft Press, 2012.

7.Белая Т.Р. Автоматизированная система документационного обеспечения управления: организация создания АС ДОУ / Т. Р. Белая – 2007. - №3. – 40-47 с.

Василенко Ж. В. Основы языка C#. Общие концеции синтаксиса./ Ж. В. Василенко. – Минск : БГУ,2008 – 50 с.

8.Вендров А. М. Практикум по проектированию программного обеспечения экономических информационных систем / А. М. Вендров. – М.: Финансы и статистика, 2010.-192с.

9.Вендров, А.М. Проектирование программного обеспечения экономических информационных систем / А.М. Вендров. – М.: Финансы и статистика, 2012. – 264 с.

10. Герберт Шмидт. Полное руководство C# 3.0. М.; Изд. дом «Вильяме», 2010, - 992.

11.Гома Х. UML. Проектирование систем реального времени, параллельных и распределенных приложений / Х. Гома. – М.: ДМК Пресс, 2016. – 700 с.

12. Ефимов В. В. Управление процессами : учебное пособие / В. В. Ефимов, М. В. Самсонова. – Ульяновск : УлГТУ, 2008. – 222 с.

13.Каменева Н.Г., Поляков В.А. Маркетинговые исследования: Учебное пособие. – 2-е изд., доп. – М.: Вузовский учебник: ИНФРА-М, 2013. – 368 с.

14.Ковалев В., Ковалев С. Секреты успешных предприятий: бизнес-процессы и организационная структура. – М.: БИТЕК, 2012. – 513 с.

15.Кузнецов С.Д. Проектирование и разработка корпоративных информационных систем. – [Электронный ресурс] – Режим доступа: http://zeus.sai.msu.ru:7000/cfin/prcorpsys/index.shtml.– Дата доступа: 10.05.2018.

16.Об информации, информатизации и защите информации: Закон Респ. Беларусь от 10 ноября 2008 года № 455-З // Нац. реестр правовых актов Респ. Беларусь. – 2008. – № 279. – 2/1552.

17. Об электронном документе и электронной цифровой подписи: Закон Респ. Беларусь от 28 декабря 2009 г. № 113-З: текст по состоянию на 25 мая 2012 г.// Нац. реестр правовых актов Респ. Беларусь. – 2010. – № 15. – 2/1665

18. Охрана труда в Беларуси. СанПиН 9-131 РБ 2000 [Электронный ресурс] – Режим доступа: http://www.otb.by- Дата доступа: 20.03.2018.

19. Павловская Т. А. C# Программирование на языке высокого уровня. ПИТЕР, 2007, - 432 с.

20.Парамонова Т.Н. Маркетинг торгового предприятия: Учебник для бакалавров / Т.Н. Парамонова, И.Н. Красюк, В.В. Лукашевич; под ред. д.э.н., проф. Т.Н. Парамоновой. – М.: Издательско-торговая корпорация «Дашков и К˚», 2016. – 284 с.

21.Программное обеспечение для автоматизации и управления предприятиями и государственными учреждениями [Электронный ресурс] – Режим доступа: http://www.parus.ua- Дата доступа: 20.04.2018.

22.Разработка алгоритмов с использованием принципов ООП на языке C# : учебно-методическое пособие / А.А. Калинин ; МГУП имени Ивана Федорова, 2014. — 106 с.

23. Семич, В. П. Охрана труда при работе на персональных электрон- новычислительных машинах и другой офисной технике: Практ. пособие. / В. П. Семич. – Мн.: ЦОТЖ 2003. – 86 с.

24. Халянов Г.Н. Моделирование, анализ, реорганизация и автоматизация бизнес процессов. —М.: 2006г. —309с.

25.Фаулер, М. Архитектура корпоративных программных приложений.: Пер. с англ. – М.: Издательский дом «Вилбямс», 2008. – 544 с.

26.Флейтман, Д.В. Жизненный цикл и живучесть корпоративных информационных систем/ Д.В. Флейтман – 2004. – Т. 6. – 74 – 84 с.

27. Ясенев, В.Н. Информационные системы и технологии в экономике: учебное пособие/ В.Н. Ясенев. – М.: Юнити-Дана, 2012. – 561с.

28.http://msdn.microsoft.com/ru-ru/library/cc879262.aspx - справочник по SQL Server

29.http://msdn.microsoft.com/ru-ru/library/dd642420.aspx - справочник по C#.NET

30.http://www.sql.ru/forum/actualtopics.aspx?bid=1 – форум по SQL Server.

ПРИЛОЖЕНИЕ А

Код класс документов

```
#region   class Договора
 public class Договор
 {
    // Поля
    string номерДоговора;
    string классификацияДоговора;
    string наименованиеКонтрагента;
    DateTime датаЗаключения;
    DateTime срокДействия;
    string суммаДоговора;
    string суммаПредоплаты;

    #region  // Свойства
    public string НомерДоговора
    {
        get { return номерДоговора; }
       set { номерДоговора = value; }
    }
    public string СуммаДоговора
    {
       get { return суммаДоговора; }
       set { суммаДоговора = value; }
    }
    public string СуммаПредоплаты
    {
       get { return суммаПредоплаты; }
       set { суммаПредоплаты = value; }
    }
    public string НаименованиеКонтрагента
    {
       get { return наименованиеКонтрагента; }
       set { наименованиеКонтрагента = value; }
    }
    public string КлассификацияДоговора
    {
```

```
            get { return классификацияДоговора; }
            set { классификацияДоговора = value; }
        }
        public DateTime ДатаЗаключения
        {
            get { return датаЗаключения; }
            set { датаЗаключения = value; }
        }
        public DateTime СрокДействия
        {
            get { return срокДействия; }
            set { срокДействия = value; }
        }
        #endregion

    #region  // Конструкторы

public Договор(string номерДоговора, string наименованиеКонтрагента,
    DateTime датаЗаключения, DateTime срокДействия,
      string суммаДоговора, string суммаПредоплаты)
        {
          this.номерДоговора = номерДоговора;
          this.наименованиеКонтрагента = наименованиеКонтрагента;
          this.датаЗаключения = датаЗаключения;
          this.срокДействия = срокДействия;
          this.суммаДоговора = суммаДоговора;
          this.суммаПредоплаты = суммаПредоплаты;
        }

      public Договор(string классификацияДоговора, string номерДоговора,
            string наименованиеКонтрагента, DateTime датаЗаключения,
    DateTime срокДействия, string суммаДоговора, string  суммаПредоплаты)
        {
          this.классификацияДоговора = классификацияДоговора;
          this.номерДоговора = номерДоговора;
          this.наименованиеКонтрагента = наименованиеКонтрагента;
          this.датаЗаключения = датаЗаключения;
          this.срокДействия = срокДействия;
          this.суммаДоговора = суммаДоговора;
```

```
            this.суммаПредоплаты = суммаПредоплаты;
        }
        public static string ToString(Договор договор)
        {
            string информация = "";
            if (договор != null)
            {
                if (договор.КлассификацияДоговора == "")
                {
    string s2 = договор.НомерДоговора;
    string s3 = договор.НаименованиеКонтрагента;
    string s4 = Convert.ToString(договор.ДатаЗаключения);
    string s5 = Convert.ToString(договор.СрокДействия);
    string s6 = договор.СуммаДоговора;
    string s7 = договор.СуммаПредоплаты;
информация = s2+";"+s3+";"+s4+";"+s5+";"+s6+";"+ s7
                }
                else
                {
    string s1 = договор.КлассификацияДоговора;
    string s2 = договор.НомерДоговора;
    string s3 = договор.НаименованиеКонтрагента;
    string s4 = Convert.ToString(договор.ДатаЗаключения);
    string s5 = Convert.ToString(договор.СрокДействия);
    string s6 = договор.СуммаДоговора;
    string s7 = договор.СуммаПредоплаты;
информация = s1+";"+s2+";"+s3+";"+s4+";" +s5+";"+s6+";"+s7;
                }
            }
            return информация;
        }
        public static Договор StringToДоговор(string информация)
        {
            Договор договор = null;
          string s = "", классификацияДоговора1 = "", номерДоговора1 = "",
контрагент1 = "", датаЗаключения1 = "", срокДействия1 = "";
        string суммаДоговора1 = "", суммаПредоплаты1 = "";
        string[] str;
        DateTime датаЗаключения;
```

```
                DateTime срокДействия;
                try
                {
            if (информация != null)
            {
               s = информация.Trim();
               str = s.Split(';');

               if (str.Length < 7)
               {
                  for (int j = 0; j < str.Length; j++)
                  {
   if (номерДоговора1 == "") { номерДоговора1 = str[j]; continue; }
   if (контрагент1 == "") { контрагент1 = str[j]; continue; }
   if (датаЗаключения1 == "") { датаЗаключения1 = str[j]; continue; }
   if (срокДействия1 == "") { срокДействия1 = str[j]; continue; }
   if (суммаДоговора1 == "") { суммаДоговора1 = str[j]; continue; }
   if (суммаПредоплаты1 == "") { суммаПредоплаты1 = str[j]; continue; }
                  }

   if (датаЗаключения1 != "")
   датаЗаключения = DateTime.Parse(датаЗаключения1);
                        else
      датаЗаключения = DateTime.Parse("1000.01.01");

      if (срокДействия1 != "")
      срокДействия = DateTime.Parse(срокДействия1);
                        else
      срокДействия = DateTime.Parse("1000.01.01");

                              договор = new Договор(номерДоговора1, контрагент1,
датаЗаключения, срокДействия, суммаДоговора1, суммаПредоплаты1);
                        }

      if (str.Length == 7)
                        {
      for (int j = 0; j < str.Length; j++)
                           {
```

```
        if (классификацияДоговора1 == "") { классификацияДоговора1 = str[j]; continue; }
        if (номерДоговора1 == "") { номерДоговора1 = str[j]; continue; }
        if (контрагент1 == "") { контрагент1 = str[j]; continue; }
        if (датаЗаключения1 == "") { датаЗаключения1 = str[j]; continue; }
        if (срокДействия1 == "") { срокДействия1 = str[j]; continue; }
        if (суммаДоговора1 == "") { суммаДоговора1 = str[j]; continue; }
        if (суммаПредоплаты1 == "") { суммаПредоплаты1 = str[j]; continue; }
                        }

        датаЗаключения = DateTime.Parse(датаЗаключения1);
        срокДействия = DateTime.Parse(срокДействия1);
        договор = new Договор(классификацияДоговора1, номерДоговора1, контрагент1, датаЗаключения, срокДействия, суммаДоговора1, суммаПредоплаты1);
                        }

        if (str.Length > 7)
                        {
        for (int j = 0; j < str.Length; j++)
                        {
        if (классификацияДоговора1 == "") { классификацияДоговора1 = str[j]; continue; }
        if (номерДоговора1 == "") { номерДоговора1 = str[j]; continue; }
        if (контрагент1 == "") { контрагент1 = str[j]; continue; }
        if (датаЗаключения1 == "") { датаЗаключения1 = str[j]; continue; }
        if (срокДействия1 == "") { срокДействия1 = str[j]; continue; }
        if (суммаДоговора1 == "") { суммаДоговора1 = str[j]; continue; }
        if (суммаПредоплаты1 == "") { суммаПредоплаты1 = str[j]; continue; }
                        }
        if (датаЗаключения1 != "")
        датаЗаключения = DateTime.Parse(датаЗаключения1);
                        else
        датаЗаключения = DateTime.Parse("1000.01.01");

        if (срокДействия1 != "")
        срокДействия = DateTime.Parse(срокДействия1);
                        else
        срокДействия = DateTime.Parse("1000.01.01");
```

```
оговор = new Договор(классификацияДоговора1, номерДоговора1, контрагент1, датаЗаключения, срокДействия, суммаДоговора1, суммаПредоплаты1);
                    }
                  }
                }
                catch { }

        return договор;  // возвращает экземпляр (договор) класса Договора
            }
            #endregion

        #region  // Перегружаем оператор ' == ' и ' != ' , результат bool.

            public override bool Equals(object input)
            {
              if (input is Договор && input != null)
              {
                Договор договор = (Договор)input;
        if (договор.номерДоговора == this.номерДоговора &&
        договор.классификацияДоговора == this.классификацияДоговора &&
        договор.наименованиеКонтрагента == this.наименованиеКонтрагента &&
        договор.датаЗаключения == this.датаЗаключения &&
        договор.срокДействия == this.срокДействия &&
        договор.суммаДоговора == this.суммаДоговора &&
        договор.суммаПредоплаты == this.суммаПредоплаты)
                    return true;
              }
              return false;
            }
            public override int GetHashCode()
            {
              return this.ToString().GetHashCode();
            }
         public static bool operator ==(Договор договора1, Договор договора2)
            {
              return договора1.Equals(договора2);
            }
        public static bool operator !=(Договор договора1, Договор договора2)
```

```
        {
            return !договора1.Equals(договора2);
        }
        #endregion
        #region
                                                    public static Договор
ВыделеннаяСтрока_Списка_Договоров_из_DataGridView(DataGridView
dataGridView1)
        {
            Договор договор = null;

                                                        string str =
Таблица.СтрокаТаблПриложенияВыделеннаяКурсором(dataGridView1);
            договор = Договор.StringToДоговор(str);
            return договор;
        }
                public int Код_Договора_в_БД(Договор договор, SqlConnection
подклБД)
        {
    if (договор.НаименованиеКонтрагента == "")
                return 0;
    int                                 кодКонтрагента                  =
Справочники.Контрагенты.КодКонтрагента_в_БД(договор.НаименованиеКонтра
гента, подклБД);
     int код_Договора_БД = 0;
      string запросБД_Код_Договора = "Select Код from Договора where
Классификация =  '" + договор.КлассификацияДоговора + "' and НомерДоговора =
'" +
                                                    договор.НомерДоговора +
"' and КодКонтрагента = " +
                                                      кодКонтрагента +  " and
ДатаЗаключения = '" + договор.ДатаЗаключения +
                                                     "'  and СрокДействия = '" +
договор.СрокДействия +
                                                     "' and СуммаДоговора = '"
+ договор.СуммаДоговора +
                                                    "' and СуммаПредоплаты =
'" + договор.СуммаПредоплаты + "'";
            подклБД.Open();
```

```
        string значениеКод_Договора_БД = Таблица.СтрокаСтолбцаТаблицыБД(запросБД_Код_Договора, подклБД);
                подклБД.Close();

        if (значениеКод_Договора_БД != "" )
        код_Договора_БД = int.Parse(значениеКод_Договора_БД);
                return код_Договора_БД;
            }
        private static void Список_Вводимых_Договоров_из_DataGridView(ref int колЗаполненныхСтрок_до_ВводаДанных_в_DataGridView,
                                                                       ref List<Договор> список_Договоров, DataGridView dataGridView1)
            {
        List<string> список_Вводимых_СтрокDataGridView = new List<string>();
        Таблица.СписокВводимыхСтрок_в_ТаблицуDataGridView(ref список_Вводимых_СтрокDataGridView, ref колЗаполненныхСтрок_до_ВводаДанных_в_DataGridView, dataGridView1);
        Договор.Создание_Списка_Договоров_из_СпискаСтрок(ref список_Вводимых_СтрокDataGridView, ref список_Договоров);
            }

             private static void Создание_Списка_Договоров_из_СпискаСтрок(ref List<string> списокСтрок, ref List<Договор> список_Договоров)
            {
              Договор договор;
              список_Договоров.Clear();

              if (списокСтрок != null)
              {
                for (int i = 0; i < списокСтрок.Count; i++)
                {
                  договор = Договор.StringToДоговор(списокСтрок[i]);
                  список_Договоров.Add(договор);
                }
              }
              списокСтрок.Clear();
            }
```

```
        public static void СохранениеДанных_об_Договоре_в_БД_из_DataGridView(ref int колЗаполненныхСтрок_до_ВводаДанных_в_DataGridView,
                                                                              DataGridView dataGridView1, SqlConnection подклБД)
                {
                   List<Договор> список_Договор = new List<Договор>();
                   // Окно с кнопками Да и Нет
                            DialogResult result = MessageBox.Show("Сохранить данные договора ?", "Вы уверены ?",
                                     MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                   if (result == DialogResult.Yes)
                   {
                      try
                      {
        Список_Вводимых_Договоров_из_DataGridView(ref колЗаполненныхСтрок_до_ВводаДанных_в_DataGridView, ref список_Договор, dataGridView1);

        string запросБД_Наименование;
        string наименованиеКонтрагента_из_БД;

        int код_Контрагента_из_БД = 0;
        int код_Договора_БД = 0;

        if (список_Договор != null)
                         {
        for (int i = 0; i < список_Договор.Count; i++)
                           {
                                                                код_Договора_БД = список_Договор[i].Код_Договора_в_БД(список_Договор[i], подклБД);

                                  if (код_Договора_БД != 0)
                                     continue;

                                  запросБД_Наименование = "Select Наименование from Контрагенты where Наименование = '" +
                                                          (список_Договор[i]).НаименованиеКонтрагента + "'";
```

```
                подклБД.Open();
                наименованиеКонтрагента_из_БД = Таблица.СтрокаСтолбцаТаблицыБД(запросБД_Наименование, подклБД);
                подклБД.Close();

                код_Контрагента_из_БД = Справочники.Контрагенты.КодКонтрагента_в_БД(наименованиеКонтрагента_из_БД, подклБД);

                string запросБД_Договора = " INSERT INTO Договора([Классификация],[НомерДоговора],[КодКонтрагента]," +
                "[ДатаЗаключения],[СрокДействия],[СуммаДоговора],[СуммаПредоплаты])" +
                " VALUES ( '" + (список_Договор[i]).КлассификацияДоговора +
                "','" + (список_Договор[i]).НомерДоговора +
                "'," + код_Контрагента_из_БД +
                ",'" + (список_Договор[i]).ДатаЗаключения +
                "','" + (список_Договор[i]).СрокДействия +
                "','" + (список_Договор[i]).СуммаДоговора +
                "','" + (список_Договор[i]).СуммаПредоплаты + "')";
                подклБД.Open();
                Таблица.СохранениеИзменениеУдалениеДанных_в_БД(запросБД_Договора, подклБД);
                подклБД.Close();
            }
        }
    }
    catch
    {
        MessageBox.Show("Произошла ошибка сохранения данных.");
    }
}
if (result == DialogResult.No)
```

```
                return;
            }

                                                    public static void
ИзменениеДанных_об_Договоре_в_БД_из_DataGridView(ref List<int>
списокИндексовИзмененныхСтрок_в_Таблице_DataGridView,
                                                    ref List<int>
списокКодовИзмененныхСтрок_в_Таблице_БД,
                                                    SqlConnection подклБД,
DataGridView dataGridView1)
            {
              // Окно с кнопками Да и Нет
                    DialogResult result = MessageBox.Show("Изменить данные
договора ?", "Вы уверены ?",
                        MessageBoxButtons.YesNo, MessageBoxIcon.Question);
              if (result == DialogResult.Yes)
              {
                try
                {
                  Таблица.РазрешениеНовойПустойСтрокиДляВводаДанных_в_
DataGridView(dataGridView1);
                Таблица.ПеремещениеАктивнойЯчейки_в_Конец_Таблицы_Dat
aGridView(dataGridView1);

                if (списокИндексовИзмененныхСтрок_в_Таблице_DataGridView
!= null && списокКодовИзмененныхСтрок_в_Таблице_БД != null)
                {
                                                    for (int i = 0; i <
списокИндексовИзмененныхСтрок_в_Таблице_DataGridView.Count; i++)
                  {
                                                    string измененнаяСтрокаТабл_DataGridView =
Таблица.ИзменяемаяСтрока_в_Таблице_DataGridView(списокИндексовИзмененн
ыхСтрок_в_Таблице_DataGridView[i], dataGridView1);

                                                    Договор договор =
Договор.StringToДоговор(измененнаяСтрокаТабл_DataGridView);
```

```
                                                    int кодКонтрагента_БД = Справочники.Контрагенты.КодКонтрагента_в_БД(договор.НаименованиеКонтрагента, подклБД);

                                подклБД.Open();

                                string измененияВТаблице_БД = " UPDATE Договора SET Классификация = '" + договор.КлассификацияДоговора + "' , НомерДоговора = '" +
                                                                                    договор.НомерДоговора + "' , КодКонтрагента = " + кодКонтрагента_БД + "," +
                                                                                    " ДатаЗаключения = '" + договор.ДатаЗаключения + "', СрокДействия = '" +
                                                                                    договор.СрокДействия + "', СуммаДоговора = '" + договор.СуммаДоговора +
                                                                                    "', СуммаПредоплаты = '" + договор.СуммаПредоплаты +
                                                                                    "' WHERE Код= " + списокКодовИзмененныхСтрок_в_Таблице_БД[i];
                                подклБД.Close();
                                подклБД.Open();
                                Таблица.СохранениеИзменениеУдалениеДанных_в_БД(измененияВТаблице_БД, подклБД);
                                подклБД.Close();
                            }
                        }
                        Таблица.ПеремещениеАктивнойЯчейки_в_Конец_Таблицы_DataGridView(dataGridView1);
                    }
                    catch
                    {
                        MessageBox.Show("Произошла ошибка изменения данных.");
                    }
                    списокИндексовИзмененныхСтрок_в_Таблице_DataGridView.Clear();
                    списокКодовИзмененныхСтрок_в_Таблице_БД.Clear();
                }
                if (result == DialogResult.No)
                    return;
            }
```

```
                                                    public static void УдалениеДанных_об_Договоре_в_БД_из_DataGridView(ref int колЗаполненныхСтрок_в_dataGridView, SqlConnection подклБД, DataGridView dataGridView1)
        {
            // коллекцияСтрок - для удаления в БД.
            List<string> списокСтрок_DataGridView = new List<string>();
            List<Договор> списокУдаляемых_Договоров = new List<Договор>();

            Таблица.Список_Строк_Табл_Приложения_ВыделенныхКурсором(ref списокСтрок_DataGridView, dataGridView1);
                                        Создание_Списка_Договоров_из_СпискаСтрок(ref списокСтрок_DataGridView, ref списокУдаляемых_Договоров);

            // Окно с кнопками Да и Нет
                DialogResult result = MessageBox.Show("Удалить договор ?", "Вы уверены ?",
                                MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (result == DialogResult.Yes)
            {
              try
              {
                string запросБД = "";
                int кодДоговора = 0;

                for (int i = 0; i < списокУдаляемых_Договоров.Count; i++)
                {
                                                        кодДоговора = списокУдаляемых_Договоров[i].Код_Договора_в_БД(списокУдаляемых_Договоров[i], подклБД);

                                    запросБД = "delete from Договора  where Код = " + кодДоговора;

                    подклБД.Close();
                    подклБД.Open();
                     Таблица.СохранениеИзменениеУдалениеДанных_в_БД(запросБД, подклБД);
```

```
                подклБД.Close();
            }
        }
        catch
        {
            MessageBox.Show("Произошла ошибка удаления данных.");
        }
        Таблица.УдалениеСтрокТаблПриложенияВыделенныхКурсором_в_DataGridView(ref колЗаполненныхСтрок_в_dataGridView, dataGridView1);
      }
      if (result == DialogResult.No)
        return;
    }

        public static string Открыть_Список_Договоров_из_БД(string классификация, SqlConnection имяПодключенияБД, DataGridView dataGridView1)
    {
        string запрос = "Select Д.Классификация, Д.НомерДоговора as 'Номер договора', К.Наименование as 'Наименование контрагента' , Д.ДатаЗаключения as 'Дата закл.договора'," +
                        " Д.СрокДействия as 'Срок действия договора', Д.СуммаДоговора as 'Сумма договора', Д.СуммаПредоплаты as 'Сумма предоплаты'" +
                        " from Контрагенты as К , Договора as Д " +
                        " where Д.Классификация like '%" + классификация + "%' and Д.КодКонтрагента = К.Код";
      имяПодключенияБД.Close();
      имяПодключенияБД.Open();
            Таблица.ЗапросДанныхТаблБД_в_DataGridView(запрос, имяПодключенияБД, dataGridView1);
      имяПодключенияБД.Close();

      запрос = "Список договоров ";
      return запрос;
    }

        public static void Поиск_Договоров(string классификация, string наименование_Контрагента, string срокДействия, DataGridView dataGridView1, SqlConnection подклБД)
```

```
            {
                // Фильтр по наименованию контрагента
                if (наименование_Контрагента != "" && срокДействия == "")
                {
                        string запросБД_Наименование = "Select Наименование from Контрагенты where Наименование =  '" + наименование_Контрагента + "'";

                    подклБД.Close();
                    подклБД.Open();
                                                string наименованиеКонтрагента_из_БД = Таблица.СтрокаСтолбцаТаблицыБД(запросБД_Наименование, подклБД);
                    подклБД.Close();

                                                        int код_Контрагента_из_БД = Справочники.Контрагенты.КодКонтрагента_в_БД(наименованиеКонтрагента_из_БД, подклБД);

                                string запрос = "Select Д.Классификация, Д.НомерДоговора as 'Номер договора', К.Наименование as  'Наименование контрагента' ," +
                                                    " Д.ДатаЗаключения  as 'Дата закл.договора', Д.СрокДействия  as 'Срок действия договора'," +
                                                        " Д.СуммаДоговора  as 'Сумма договора', Д.СуммаПредоплаты  as 'Сумма предоплаты'" +
                                    " from  Контрагенты as К , Договора  as Д " +
                                     " where Д.Классификация like '%" + классификация + "%' and Д.КодКонтрагента = К.Код and Д.КодКонтрагента = " + код_Контрагента_из_БД;
                    подклБД.Open();
                     Таблица.ЗапросДанныхТаблБД_в_DataGridView(запрос, подклБД, dataGridView1);
                    подклБД.Close();

                }

                // Фильтр по наименованию контрагента и по сроку действия
                if (наименование_Контрагента != "" && срокДействия != "")
                {
                        string запросБД_Наименование = "Select Наименование from Контрагенты where Наименование =  '" + наименование_Контрагента + "'";
```

```
                подклБД.Close();
                подклБД.Open();
                                        string наименованиеКонтрагента_из_БД = Таблица.СтрокаСтолбцаТаблицыБД(запросБД_Наименование, подклБД);
                подклБД.Close();

                                                int код_Контрагента_из_БД = Справочники.Контрагенты.КодКонтрагента_в_БД(наименованиеКонтрагента_из_БД, подклБД);

                        string запрос = "Select Д.Классификация, Д.НомерДоговора as 'Номер договора', К.Наименование as  'Наименование контрагента' ," +
                                                " Д.ДатаЗаключения  as 'Дата закл.договора', Д.СрокДействия  as 'Срок действия договора'," +
                                                " Д.СуммаДоговора  as 'Сумма договора', Д.СуммаПредоплаты  as 'Сумма предоплаты'" +
                                " from  Контрагенты as К , Договора  as Д " +
                                 " where Д.Классификация like '%" + классификация + "%' and Д.КодКонтрагента = К.Код and Д.КодКонтрагента = " + код_Контрагента_из_БД +
                                        " and Д.СрокДействия >= '" + срокДействия + "'";
                подклБД.Open();
                 Таблица.ЗапросДанныхТаблБД_в_DataGridView(запрос, подклБД, dataGridView1);
                подклБД.Close();
            }
        }

                public static void Фильтр_Договоров(string классификация, string наименование_Контрагента, string номер_Договора, string датаЗаключения, string срокДействия,
                                                DataGridView dataGridView1, SqlConnection подклБД)
        {
            // Фильтр по номеру договора.
                                        if ((наименование_Контрагента == null || наименование_Контрагента == "") && (номер_Договора != "" || номер_Договора != null) &&
```

```
                (датаЗаключения == "" || датаЗаключения == null) && (срокДействия == "" || срокДействия == null))
            {
                string запрос = "Select Д.Классификация, Д.НомерДоговора as 'Номер договора', К.Наименование as  'Наименование контрагента' ," +
                                " Д.ДатаЗаключения  as 'Дата закл.договора', Д.СрокДействия  as 'Срок действия договора'," +
                                " Д.СуммаДоговора  as 'Сумма договора', Д.СуммаПредоплаты  as 'Сумма предоплаты'" +
                               " from  Контрагенты as К , Договора  as Д " +
                               " where  Д.Классификация like '%" + классификация + "%' and Д.НомерДоговора = '" + номер_Договора + "' and Д.КодКонтрагента = К.Код";
                подклБД.Open();
                Таблица.ЗапросДанныхТаблБД_в_DataGridView(запрос, подклБД, dataGridView1);
                подклБД.Close();
            }

            // Фильтр по дате заключения
            if (наименование_Контрагента == "" && номер_Договора == "" && датаЗаключения != "" && срокДействия == "")
            {
                string запрос = "Select Д.Классификация, Д.НомерДоговора as 'Номер договора', К.Наименование as  'Наименование контрагента' ," +
                                " Д.ДатаЗаключения  as 'Дата закл.договора', Д.СрокДействия  as 'Срок действия договора'," +
                                " Д.СуммаДоговора  as 'Сумма договора', Д.СуммаПредоплаты  as 'Сумма предоплаты'" +
                               " from  Контрагенты as К , Договора  as Д " +
                                " where Д.Классификация like '%" + классификация + "%' and Д.ДатаЗаключения >= '" + датаЗаключения +
                                    "' and Д.КодКонтрагента = К.Код";
                подклБД.Open();
                Таблица.ЗапросДанныхТаблБД_в_DataGridView(запрос, подклБД, dataGridView1);
                подклБД.Close();
            }

            // Фильтр по сроку действия
```

```
                if (наименование_Контрагента == "" && номер_Договора == "" &&
датаЗаключения == "" && срокДействия != "")
                {
                    string запрос = "Select Д.Классификация, Д.НомерДоговора as
'Номер договора', К.Наименование as  'Наименование контрагента' ," +
                                    " Д.ДатаЗаключения  as 'Дата закл.договора',
Д.СрокДействия  as 'Срок действия договора'," +
                                    " Д.СуммаДоговора  as 'Сумма договора',
Д.СуммаПредоплаты  as 'Сумма предоплаты'" +
                                " from  Контрагенты as К , Договора  as Д " +
                                 " where Д.Классификация like '%" + классификация + "%'
and Д.СрокДействия <= '" + срокДействия + "' and Д.КодКонтрагента = К.Код";
                подклБД.Open();
                 Таблица.ЗапросДанныхТаблБД_в_DataGridView(запрос, подклБД,
dataGridView1);
                подклБД.Close();
                }

                // Фильтр по дате заключения и сроку действия ( за определенный
интервал времени )
                if (наименование_Контрагента == "" && номер_Договора == "" &&
датаЗаключения != "" && срокДействия != "" )
                {
                    string запрос = "Select Д.Классификация, Д.НомерДоговора as
'Номер договора', К.Наименование as  'Наименование контрагента' ," +
                                    " Д.ДатаЗаключения  as 'Дата закл.договора',
Д.СрокДействия  as 'Срок действия договора'," +
                                    " Д.СуммаДоговора  as 'Сумма договора',
Д.СуммаПредоплаты  as 'Сумма предоплаты'" +
                                " from  Контрагенты as К , Договора  as Д " +
                                 " where Д.Классификация like '%" + классификация + "%'
and Д.ДатаЗаключения >= '" + датаЗаключения +
                                    "' and Д.СрокДействия <= '" + срокДействия + "' and
Д.КодКонтрагента = К.Код";
                подклБД.Open();
                 Таблица.ЗапросДанныхТаблБД_в_DataGridView(запрос, подклБД,
dataGridView1);
                подклБД.Close();
                }
```

```
                // Фильтр по наименованию контрагента
                  if ((наименование_Контрагента != "") && ((номер_Договора == "" )
&& (датаЗаключения == "") && (срокДействия == "")))
                {
                        string запросБД_Наименование = "Select Наименование from
Контрагенты where Наименование =  '" + наименование_Контрагента + "'";

                  подклБД.Close();
                  подклБД.Open();
                                                string  наименованиеКонтрагента_из_БД  =
Таблица.СтрокаСтолбцаТаблицыБД(запросБД_Наименование, подклБД);
                  подклБД.Close();

                                                    int  код_Контрагента_из_БД  =
Справочники.Контрагенты.КодКонтрагента_в_БД(наименованиеКонтрагента_из
_БД, подклБД);

                        string запрос = "Select Д.Классификация, Д.НомерДоговора as
'Номер договора', К.Наименование as  'Наименование контрагента' ," +
                                              " Д.ДатаЗаключения  as 'Дата закл.договора',
Д.СрокДействия  as 'Срок действия договора'," +
                                                 " Д.СуммаДоговора  as 'Сумма договора',
Д.СуммаПредоплаты  as 'Сумма предоплаты'" +
                                  " from  Контрагенты as К , Договора  as Д " +
                                   " where Д.Классификация like '%" + классификация + "%'
and  Д.КодКонтрагента  =  К.Код  and  Д.КодКонтрагента  =  " +
код_Контрагента_из_БД ;
                  подклБД.Open();
                   Таблица.ЗапросДанныхТаблБД_в_DataGridView(запрос, подклБД,
dataGridView1);
                  подклБД.Close();
                }

                // Фильтр по наименованию контрагента и по дате заключения
                  if (( наименование_Контрагента != "" && датаЗаключения != ""  &&
срокДействия == "" ) && (номер_Договора == "" ))
                {
```

```
                string запросБД_Наименование = "Select Наименование from Контрагенты where Наименование =  '" + наименование_Контрагента + "'";

                подклБД.Close();
                подклБД.Open();
                string наименованиеКонтрагента_из_БД = Таблица.СтрокаСтолбцаТаблицыБД(запросБД_Наименование, подклБД);
                подклБД.Close();

                int код_Контрагента_из_БД = Справочники.Контрагенты.КодКонтрагента_в_БД(наименованиеКонтрагента_из_БД, подклБД);

                string запрос = "Select Д.Классификация, Д.НомерДоговора as 'Номер договора', К.Наименование as  'Наименование контрагента' ," +
                " Д.ДатаЗаключения  as 'Дата закл.договора', Д.СрокДействия  as 'Срок действия договора'," +
                " Д.СуммаДоговора  as 'Сумма договора', Д.СуммаПредоплаты  as 'Сумма предоплаты'" +
                " from  Контрагенты as К , Договора  as Д " +
                " where Д.Классификация like '%" + классификация + "%' and Д.КодКонтрагента = К.Код and Д.КодКонтрагента = " + код_Контрагента_из_БД +
                " and Д.ДатаЗаключения >= '" + датаЗаключения + "'";
                подклБД.Open();
                Таблица.ЗапросДанныхТаблБД_в_DataGridView(запрос, подклБД, dataGridView1);
                подклБД.Close();
            }

            // Фильтр по наименованию контрагента и по сроку действия
            if ((наименование_Контрагента != "" && датаЗаключения == "" && срокДействия != "") && (номер_Договора == ""))
            {
                string запросБД_Наименование = "Select Наименование from Контрагенты where Наименование =  '" + наименование_Контрагента + "'";

                подклБД.Close();
                подклБД.Open();
```

```
                                        string наименованиеКонтрагента_из_БД = Таблица.СтрокаСтолбцаТаблицыБД(запросБД_Наименование, подклБД);
                    подклБД.Close();

                                                int код_Контрагента_из_БД = Справочники.Контрагенты.КодКонтрагента_в_БД(наименованиеКонтрагента_из_БД, подклБД);

                    string запрос = "Select Д.Классификация, Д.НомерДоговора as 'Номер договора', К.Наименование as 'Наименование контрагента' ," +
                                                " Д.ДатаЗаключения as 'Дата закл.договора', Д.СрокДействия as 'Срок действия договора'," +
                                                " Д.СуммаДоговора as 'Сумма договора', Д.СуммаПредоплаты as 'Сумма предоплаты'" +
                                " from Контрагенты as К , Договора as Д " +
                                 " where Д.Классификация like '%" + классификация + "%' and Д.КодКонтрагента = К.Код and Д.КодКонтрагента = " + код_Контрагента_из_БД +
                                        " and Д.СрокДействия <= '" + срокДействия + "'";
                    подклБД.Open();
                     Таблица.ЗапросДанныхТаблБД_в_DataGridView(запрос, подклБД, dataGridView1);
                    подклБД.Close();
                }

                // Фильтр по наименованию контрагента и по дате заключения и сроку действия ( за определенный интервал времени )
                if ((наименование_Контрагента != "" && датаЗаключения != "" && срокДействия != "") && (номер_Договора == ""))
                {
                    string запросБД_Наименование = "Select Наименование from Контрагенты where Наименование = '" + наименование_Контрагента + "'";

                    подклБД.Close();
                    подклБД.Open();
                                        string наименованиеКонтрагента_из_БД = Таблица.СтрокаСтолбцаТаблицыБД(запросБД_Наименование, подклБД);
                    подклБД.Close();
```

```
                                                int код_Контрагента_из_БД = Справочники.Контрагенты.КодКонтрагента_в_БД(наименованиеКонтрагента_из_БД, подклБД);

                                string запрос = "Select Д.Классификация, Д.НомерДоговора as 'Номер договора', К.Наименование as 'Наименование контрагента' ," +
                                                " Д.ДатаЗаключения as 'Дата закл.договора', Д.СрокДействия as 'Срок действия договора'," +
                                                " Д.СуммаДоговора as 'Сумма договора', Д.СуммаПредоплаты as 'Сумма предоплаты'" +
                                " from Контрагенты as К , Договора as Д " +
                                 " where Д.Классификация like '%" + классификация + "%' and Д.КодКонтрагента = К.Код and Д.КодКонтрагента = " + код_Контрагента_из_БД +
                                                " and Д.ДатаЗаключения >= '" + датаЗаключения + "' and Д.СрокДействия <= '" + срокДействия + "'";
                        подклБД.Open();
                         Таблица.ЗапросДанныхТаблБД_в_DataGridView(запрос, подклБД, dataGridView1);
                        подклБД.Close();
                    }
                }
                #endregion
            }

            #region public class Договор_Купли_Продажи

            public class Договор_Купли_Продажи : Договор
            {
                #region // Конструкторы

                        public Договор_Купли_Продажи(string номерДоговора, string наименованиеКонтрагента, DateTime датаЗаключения, DateTime срокДействия,
                                                string суммаДоговора, string суммаПредоплаты) :
                    base(номерДоговора, наименованиеКонтрагента, датаЗаключения, срокДействия, суммаДоговора, суммаПредоплаты) { }
```

```
        public Договор_Купли_Продажи(string классификацияДоговора, string номерДоговора, string наименованиеКонтрагента, DateTime датаЗаключения,
                                     DateTime срокДействия, string суммаДоговора, string суммаПредоплаты) :
                                         base(классификацияДоговора, номерДоговора, наименованиеКонтрагента, датаЗаключения, срокДействия, суммаДоговора, суммаПредоплаты) { }

        public static string ToString(Договор_Купли_Продажи договор)
        {
            string информация = "";

            if (договор != null)
            {
                if (договор.КлассификацияДоговора == "")
                {
                    string s2 = договор.НомерДоговора;
                    string s3 = договор.НаименованиеКонтрагента;
                    string s4 = Convert.ToString(договор.ДатаЗаключения);
                    string s5 = Convert.ToString(договор.СрокДействия);
                    string s6 = договор.СуммаДоговора;
                    string s7 = договор.СуммаПредоплаты;

                    информация = s2 + ";" + s3 + ";" + s4 + ";" + s5 + ";" + s6 + ";" + s7;
                }
                else
                {
                    string s1 = договор.КлассификацияДоговора;
                    string s2 = договор.НомерДоговора;
                    string s3 = договор.НаименованиеКонтрагента;
                    string s4 = Convert.ToString(договор.ДатаЗаключения);
                    string s5 = Convert.ToString(договор.СрокДействия);
                    string s6 = договор.СуммаДоговора;
                    string s7 = договор.СуммаПредоплаты;

                    информация = s1 + ";" + s2 + ";" + s3 + ";" + s4 + ";" + s5 + ";" + s6 + ";" + s7;
                }
            }
```

```
            return информация;
        }

                                                public static Договор_Купли_Продажи
StringToДоговор_Купли_Продажи(string информация)
        {
            Договор_Купли_Продажи договор = null;
                string s, классификацияДоговора1 = "", номерДоговора1 = "",
контрагент1 = "", датаЗаключения1 = "", срокДействия1 = "";
            string суммаДоговора1 = "", суммаПредоплаты1 = "";
            string[] str;
            DateTime датаЗаключения;
            DateTime срокДействия;

            try
            {
              if (информация != null)
              {
                 s = информация.Trim();
                str = s.Split(';');

                 if (str.Length < 7)
                 {
                    for (int j = 0; j < str.Length; j++)
                    {
                                if (номерДоговора1 == "") { номерДоговора1 = str[j];
continue; }
                       if (контрагент1 == "") { контрагент1 = str[j]; continue; }
                             if (датаЗаключения1 == "") { датаЗаключения1 = str[j];
continue; }
                       if (срокДействия1 == "") { срокДействия1 = str[j]; continue; }
                                if (суммаДоговора1 == "") { суммаДоговора1 = str[j];
continue; }
                         if (суммаПредоплаты1 == "") { суммаПредоплаты1 = str[j];
continue; }
                    }

                    if (датаЗаключения1 != "")
                       датаЗаключения = DateTime.Parse(датаЗаключения1);
```

```
                    else
                        датаЗаключения = DateTime.Parse("1000.01.01");

                    if (срокДействия1 != "")
                        срокДействия = DateTime.Parse(срокДействия1);
                    else
                        срокДействия = DateTime.Parse("1000.01.01");

                        договор = new Договор_Купли_Продажи(номерДоговора1,
контрагент1, датаЗаключения, срокДействия,
                                                суммаДоговора1, суммаПредоплаты1);
                }

                if (str.Length == 7)
                {
                    for (int j = 0; j < str.Length; j++)
                    {
                                                    if (классификацияДоговора1 == "")
{ классификацияДоговора1 = str[j]; continue; }
                                    if (номерДоговора1 == "") { номерДоговора1 = str[j];
continue; }
                        if (контрагент1 == "") { контрагент1 = str[j]; continue; }
                                if (датаЗаключения1 == "") { датаЗаключения1 = str[j];
continue; }
                        if (срокДействия1 == "") { срокДействия1 = str[j]; continue; }
                                    if (суммаДоговора1 == "") { суммаДоговора1 = str[j];
continue; }
                                if (суммаПредоплаты1 == "") { суммаПредоплаты1 = str[j];
continue; }
                    }

                    датаЗаключения = DateTime.Parse(датаЗаключения1);
                    срокДействия = DateTime.Parse(срокДействия1);
                                                        договор = new
Договор_Купли_Продажи(классификацияДоговора1, номерДоговора1,
контрагент1, датаЗаключения, срокДействия,
                                                суммаДоговора1, суммаПредоплаты1);
                }
```

```
                    if (str.Length > 7)
                    {
                        for (int j = 0; j < str.Length; j++)
                        {
                                                    if (классификацияДоговора1 == "")
{ классификацияДоговора1 = str[j]; continue; }
                                        if (номерДоговора1 == "") { номерДоговора1 = str[j];
continue; }
                            if (контрагент1 == "") { контрагент1 = str[j]; continue; }
                                    if (датаЗаключения1 == "") { датаЗаключения1 = str[j];
continue; }
                            if (срокДействия1 == "") { срокДействия1 = str[j]; continue; }
                                        if (суммаДоговора1 == "") { суммаДоговора1 = str[j];
continue; }
                              if (суммаПредоплаты1 == "") { суммаПредоплаты1 = str[j];
continue; }
                        }
                        if (датаЗаключения1 != "")
                            датаЗаключения = DateTime.Parse(датаЗаключения1);
                        else
                            датаЗаключения = DateTime.Parse("1000.01.01");

                        if (срокДействия1 != "")
                            срокДействия = DateTime.Parse(срокДействия1);
                        else
                            срокДействия = DateTime.Parse("1000.01.01");

                                                                договор = new
Договор_Купли_Продажи(классификацияДоговора1,                номерДоговора1,
контрагент1, датаЗаключения, срокДействия,
                                                    суммаДоговора1, суммаПредоплаты1);
                    }
                  }
                }
                catch { }

                return договор;
        }       #endregion
```

```
        #region  // Перегружаем оператор ' == ' и ' != ' , результат bool.

        public override bool Equals(object input)
        {
          if (input is Договор_Купли_Продажи && input != null)
          {
                                                Договор_Купли_Продажи  договор  =
(Договор_Купли_Продажи)input;
            if (договор.НомерДоговора == this.НомерДоговора &&
                                                 договор.КлассификацияДоговора  ==
this.КлассификацияДоговора &&
                                               договор.НаименованиеКонтрагента  ==
this.НаименованиеКонтрагента &&
              договор.ДатаЗаключения == this.ДатаЗаключения &&
              договор.СрокДействия == this.СрокДействия &&
              договор.СуммаДоговора == this.СуммаДоговора &&
              договор.СуммаПредоплаты == this.СуммаПредоплаты) // &&

              return true;
          }
          return false;
        }
        public override int GetHashCode()
        {
          return this.ToString().GetHashCode();
        }
           public static bool operator ==(Договор_Купли_Продажи договора1,
Договор_Купли_Продажи договора2)
        {
          return договора1.Equals(договора2);
        }
           public static bool operator !=(Договор_Купли_Продажи договора1,
Договор_Купли_Продажи договора2)
        {
          return !договора1.Equals(договора2);
        }
        #endregion
```

```
                public new static Договор_Купли_Продажи
ВыделеннаяСтрока_Списка_Договоров_из_DataGridView(DataGridView
dataGridView1)
        {
            Договор_Купли_Продажи договор = null;

                                                string str =
Таблица.СтрокаТаблПриложенияВыделеннаяКурсором(dataGridView1);
                                                    договор =
Договор_Купли_Продажи.StringToДоговор_Купли_Продажи(str);

            return договор;
        }

                                                public void
ОткрытьДоговор_Продажи_в_Word(Договор_Купли_Продажи договор,
Справочники.Контрагенты нашаФирма, SqlConnection подклБД,
                                                string
путь_к_Файлу_Шаблону_Word_Документа,
                                                string
путь_к_Файлу_для_Сохранения_Word_Документа)
        {
            bool флаг_Откр_Word_Документа = true;

            string юр_Адрес_Поставщика = "";
            string банк_Счет_Поставщика = "";
            string контактные_Телефоны_Поставщика = "";

            string юр_Адрес_Покупателя = "";
            string банк_Счет_Покупателя = "";
            string контактные_Телефоны_Покупателя = "";
            string[] списокНаименованияВалют = { "бел.руб" };
            string[] списокОтделов = { "Директор" };

                string[] массив_СтрокЗаглушек_в_Шаблоне_Word_Документа = {
"{номерДоговора}", "{сторонаДоговора_Поставщик}", " {фио_Поставщик}",
                                                "
{сторонаДоговора_Покупатель}", " {фио_Покупатель}",
```

"{суммаДоговора}", "{суммаПредоплаты}", "{датаЗаключенияДоговора}", "{срокДействияДоговора}"};

string фио_Поставщика = нашаФирма.ФИО_Руководителя_Контрагента(нашаФирма, подклБД);

флаг_Откр_Word_Документа = нашаФирма.ЮрАдрес_БанкСчета_КонтактныеТелефоны_Контрагента(нашаФирма, ref юр_Адрес_Поставщика, ref банк_Счет_Поставщика, ref контактные_Телефоны_Поставщика, списокНаименованияВалют, списокОтделов, подклБД);

string строка_Подписи_ФИО_Руководителя_Поставщика = нашаФирма.Строка_Подписи_ФИО_Руководителя_Контрагента(нашаФирма, подклБД);

string[] массив_Заголовков_Табл = { "Поставщик :", "Покупатель :" };

string реквизиты_Поставщика = юр_Адрес_Поставщика + банк_Счет_Поставщика + контактные_Телефоны_Поставщика + строка_Подписи_ФИО_Руководителя_Поставщика;

Справочники.Контрагенты покупатель = Справочники.Контрагенты.Данные_об_Контрагенте_из_БД(договор.НаименованиеКонтрагента, подклБД);

string фио_Покупателя = покупатель.ФИО_Руководителя_Контрагента(покупатель, подклБД);

флаг_Откр_Word_Документа = покупатель.ЮрАдрес_БанкСчета_КонтактныеТелефоны_Контрагента(покупатель, ref юр_Адрес_Покупателя, ref банк_Счет_Покупателя, ref контактные_Телефоны_Покупателя, списокНаименованияВалют, списокОтделов, подклБД);

```
                string строка_Подписи_ФИО_Руководителя_Покупателя = покупатель.Строка_Подписи_ФИО_Руководителя_Контрагента(покупатель, подклБД);


                string реквизиты_Покупателя = юр_Адрес_Покупателя + банк_Счет_Покупателя + контактные_Телефоны_Покупателя + строка_Подписи_ФИО_Руководителя_Покупателя;

                string[] массив_Текстов_Вставляемых_в_Шаблон_Word_Документа = { договор.НомерДоговора,
                                                                                нашаФирма.Наименование, фио_Поставщика,
                                                                                договор.НаименованиеКонтрагента, фио_Покупателя,
                                                                                договор.СуммаДоговора, договор.СуммаПредоплаты,
                        String.Format("{0:D}", договор.ДатаЗаключения), String.Format("{0:D}", договор.СрокДействия)};

                string[] массив_Строк_Табл = { реквизиты_Поставщика, реквизиты_Покупателя };

                if (флаг_Откр_Word_Документа == true)
                    Работа_с_Word.Вставка_Текста_в_Шаблон_Word_Документа_и_Табл_в_конец(путь_к_Файлу_Шаблону_Word_Документа,
                    путь_к_Файлу_для_Сохранения_Word_Документа,
                                                                                массив_СтрокЗаглушек_в_Шаблоне_Word_Документа,
                                                                                массив_Текстов_Вставляемых_в_Шаблон_Word_Документа,
                                                                                массив_Заголовков_Табл, массив_Строк_Табл);
                }

                public void ОткрытьДоговор_Закупки_в_Word(string путь_к_Файлу, string путь_к_Файлу_для_Сохран_Word_Документа, Договор_Купли_Продажи договор, string расширФайла)
                {
```

```
                try
                {
                    путь_к_Файлу = путь_к_Файлу + договор.НомерДоговора.Trim() +
расширФайла;
                        Работа_с_Word.Просмотр_Word_Документа(путь_к_Файлу,
путь_к_Файлу_для_Сохран_Word_Документа);
                }
                catch
                {
                    MessageBox.Show("Такого договора нет. Сохраните ее в папке :
Договор_Купли_Продажи данного приложения.");
                }
            }

        }
        #endregion
```