


Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Лесосибирский педагогический институт –
филиал Сибирского федерального университета

Кафедра высшей математики, информатики и естествознания

УТВЕРЖДАЮ

Заведующий кафедрой

 Н.Ф. Романцова

подпись

« 8 » июня 2018 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 Информационные системы и технологии

**Разработка информационной системы учета товаров на оптовом складе
ООО «Современные смазочные материалы»**

Руководитель


подпись, дата

доцент, канд.экон.наук

А.В.Рубцов

Выпускник


подпись, дата

Т.А.Гуреева

Лесосибирск 2018

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка информационной системы учета товаров на оптовом складе ООО «Современные смазочные материалы» содержит 142 страницы текстового документа, три приложения, 40 использованных источников.

АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО, СИСТЕМА УПРАВЛЕНИЯ, ПРОГРАММНЫЙ ПРОДУКТ, АВТОМАТИЗАЦИЯ, БАЗА ДАННЫХ, СУБД

Цель выпускной квалификационной работы состоит в совершенствовании складского учета организации «Современные смазочные материалы» посредством разработки и внедрения автоматизированной информационной системы.

Объектом исследования настоящей дипломной работы выступает организация ООО «Современные смазочные материалы», основная деятельность которой связана с оптовой продажей смазочных материалов компании Газпром-нефть-СМ.

Основные задачи:

- провести общий анализ организации с целью выявления процессов учета, требующих автоматизации;
- изучить организационную структуру организации и выявить подразделения, участвующие в процессах складского учета;
- изучить существующую технологию учета складских операций в организации;
- описать функциональные требования к проектируемой системе;
- провести анализ существующих ИС от известных разработчиков для выбранной предметной области;
- описать процесс выполнения учета складских операций по предлагаемой технологии, т.е. после внедрения ИС;
- описать требования к видам обеспечения проектируемой системы;

- описать входные и выходные потоки информации проектируемой системы;
- провести анализ средств разработки приложения и базы данных;
- построить модель данных на логическом и физическом уровнях;
- разработать автоматизированную информационную систему и описать принцип ее работы;
- провести тестирование созданной ИС;
- оценить экономический эффект от внедрения разработанной системы на объекте исследования.

Цель и задачи определили структуру выпускной квалификационной работы. Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка использованных источников и приложения.

СОДЕРЖАНИЕ

Введение.....	5
1 Анализ автоматизации складского учета.....	8
1.1 Характеристика ООО «Современные смазочные материалы»	8
1.2 Существующая технология выполнения исследуемого процесса	13
1.3 Проблемный анализ ключевых процессов организации.....	19
1.4 Постановка задачи на проектирование и информатизацию исследуемого процесса	212
1.5 Анализ существующих разработок	28
1.6 Предлагаемая технология выполнения исследуемого процесса.....	31
1.7 Описание видов обеспечения исследуемого процесса	34
2 Проектная часть.....	41
2.1 Обоснование выбора средств разработки.....	40
2.2 Информационное обеспечение	45
2.3 Описание реализации базы данных ИС	49
2.4 Описание реализации клиентской части ИС.....	612
3 Обоснование экономической эффективности проекта	923
3.1 Определение экономической эффективности проекта	92
3.2 Определение показателей прямой экономической эффективности	93
Заключение	1001
Список использованных источников.....	104
Приложение А Описание обозначений базы данных.....	10708
Приложение Б Перечень форм разработанной программы.....	108109
Приложение В Листинг программы.....	107

ВВЕДЕНИЕ

Актуальность темы исследования обуславливается тем, что в процессе осуществления складских операций сотрудникам организации приходится обрабатывать большие объемы информации: ассортимент товаров, поставщики, единицы измерения, сроки хранения, адреса хранения, заказы клиентов, сроки доставки и т.п. Учет данных достаточно долгий и трудоемкий процесс, ошибка в котором может привести к просчетам во всех процедурах складского учета.

Отечественный рынок информационных систем представлен множеством разработок для автоматизации складского учета, однако их функционал не всегда удается подстроить под процессы любой организации. Такие проекты требуют дополнительных затрат на доработку и изменение заложенных в них функций. При этом, стоимость доработки, обозначенная разработчиком, порой превышает рациональные пределы, в связи с чем, заказчик вынужден использовать проект, в готовом виде изменяя существующие процессы под требования используемой ИС.

Объектом исследования настоящей выпускной квалификационной работы выступает организация ООО «Современные смазочные материалы», основная деятельность которой связана с оптовой продажей смазочных материалов компании Газпромнефть-СМ.

Предметом исследования является процесс оптовой реализации смазочных материалов со склада ООО «Современные смазочные материалы».

Цель дипломной работы состоит в совершенствовании складского учета организации «Современные смазочные материалы», средством разработки и внедрения автоматизированной информационной системы.

Для достижения цели, поставленной в работе, необходимо решить следующие задачи:

– провести общий анализ организации с целью выявления процессов учета, требующих автоматизации;

- изучить организационную структуру организации и выявить подразделения, участвующие в процессах складского учета;
- изучить существующую технологию учета складских операций в организации;
- описать функциональные требования к проектируемой системе;
- провести анализ существующих ИС от известных разработчиков для выбранной предметной области;
- описать процесс выполнения учета складских операций по предлагаемой технологии, т.е. после внедрения ИС;
- описать требования к видам обеспечения проектируемой системы;
- описать входные и выходные потоки информации проектируемой системы;
- провести анализ средств разработки приложения и базы данных;
- построить модель данных на логическом и физическом уровнях;
- разработать автоматизированную информационную систему и описать принцип ее работы;
- провести тестирование созданной ИС;
- оценить экономический эффект от внедрения разработанной системы на объекте исследования.

Структура выпускной работы представлена введением, тремя главами, заключением, списком использованной литературы и приложениями.

Для изучения теоретических основ объекта исследования, в первой главе проведено описание деятельности организации «Современные смазочные материалы», изучена его оргструктура, описана технологии обработки информации с целью выявления недостатков существующего учета, был определен перечень задач, которые должна решать проектируемая система, и описаны требования к ИС.

В первой главе также проведен обзор и анализ наиболее популярных ИС от известных разработчиков для учета складских операций и описана предлагаемая технология выполнения исследуемого процесса.

Во второй главе проведен обзор и анализ средств разработки клиентской части ИС и систем управления базами данных с целью выбора решения, способного удовлетворить требованиям, предъявленным к разрабатываемой системе и описано информационное обеспечение задачи.

Во второй главе описана логическая и физическая модель данных для исследуемой предметной области: приводится состав и структура базы данных, обозначены ключевые поля и связи между таблицами, представлено графическое отображение схемы базы данных и т.п.

Также представлен сценарий диалога пользователя с разработанной ИС: описаны процедуры заполнения справочников системы, регистрации поступивших товаров от поставщиков, регистрации заказов и их комплектования, перемещения и списания товаров на складе, формирования итоговых документов, отчетов и вывода их на печать.

Третья глава посвящена оценке экономической эффективности от внедрения проекта: производится расчет стоимостных и трудовых затрат, происходит сравнение базовых показателей и показателей, полученных после внедрения системы.

1 Анализ автоматизации складского учета

1.1 Характеристика ООО «Современные смазочные материалы»

ООО «Современные смазочные материалы» является официальным дистрибьютором ООО «Газпромнефть-СМ» на территории Красноярска и Красноярского края [25].

Организация работает на российском рынке смазочных материалов с 2001 года. В ассортименте организации - продукция семейства G-Family (G-Energy - масла для легкового и коммерческого транспорта, G-Profi - масла для коммерческих, грузовых автомобилей и спецтехники, G-Box и G-Truck – трансмиссионные масла, смазки и технические жидкости, G-Motion и G-Wave – для двухтактных двигателей) [25].

Вся продукция поставляется в фирменной опломбированной заводской упаковке с паспортами и сертификатами качества.

Клиентами организации являются как юридические, так и физические лица.

Успех компаний, основная деятельность которых связана с продажами, зависит от качества продаваемых товаров. В связи с этим, организация старается предлагать своим клиентам самые лучшие и качественные смазочные материалы.

Масла G-Energy имеют одобрения ведущих мировых производителей, таких как MB, VW, BMW, Renault, GM и др.

В 2010 г. «Газпромнефть-СМ» была сертифицирована по стандартам, принятым международной организацией по стандартизации (ISO), участниками которой являются представители 147 стран мира. В ноябре 2012 года по итогам ежегодного надзорного аудита компания вновь подтвердила соответствие требованиям стандартов ISO 9001, ISO 14001, OHSAS 18001. На сегодняшний день в России процедуры сертификации по ISO не являются обязательными, однако

подтверждение наличия результативной системы менеджмента является значимым для заключения договоров с крупными международными автопроизводителями.

С 2012 года в компании внедряются требования технической спецификации ISO TS 16949:2009 с целью развития системы менеджмента качества, обеспечивающей постоянное улучшение, предупреждение дефектов и потерь в цепи поставок. В 2013 г. компания планирует пройти сертификационный аудит по соответствию требованиям ISO TS 16949:2009 – стандарта, отвечающего международным стандартам качества продукции и управления на производстве в автомобильной промышленности. Также в 2013 г. пройти сертификацию по всем 4 стандартам готовится Московский завод смазочных материалов в г. Фрязино.

Проведение данных мероприятий необходимо для дальнейшего развития взаимоотношений с крупнейшими автомобильными концернами. Помимо производственных активов в России, принадлежащий компании завод в г. Бари (Италия), регулярно проходит аудиты со стороны органа по сертификации и имеет все необходимые сертификаты систем менеджмента – ISO 9001, ISO 14001, OHSAS 18001. Соответствие требованиям всех вышеперечисленных стандартов подтверждает результативный менеджмент качества (ISO 9001:2008), экологический менеджмент (ISO 14001:2004), менеджмент охраны труда и безопасности на производстве (OHSAS 18001:2007) ООО «Газпромнефть-СМ» и Gazpromneft Lubricants Italia S.p.A. в г. Бари. [25]

Процедура по контролю качества продукции начинается с момента ее поставки на склад организации и повторно реализуется в момент комплектования заказа для доставки клиенту. Такой, двойной метод учета позволяет избежать случаев продажи недоброкачественных товаров и снижения репутации организации.

Учитывая масштабы организации и ассортимент реализуемых товаров, контроль качества товаров является довольно сложной и трудоемкой процеду-

рой, особенно если он проводится без использования автоматизированных систем учета.

Так как реализация производится непосредственно со склада первоочередной задачей, стоящей в данной работе, является разработка автоматизированной информационной системы учета складских операций в ООО «Современные смазочные материалы».

Без учета товаров на складах невозможно обеспечить контроль количества оставшихся материалов каждого вида и сроков их хранения, в связи с чем, данная область является важной на пути к автоматизации и минимализации издержек области исследования.

Для определения списка автоматизируемых задач, необходимо подробнее ознакомиться с организационной структурой организации и выявить перечень подразделений, задействованных в операции «складской учет» (начиная от поставки товаров на склад и заканчивая упаковкой и доставкой товаров покупателем). На рисунке 1 представлена организационная структура ООО «Современные смазочные материалы».

Руководство организацией осуществляет генеральный директор, который: обеспечивает исполнение всех обязательств организации перед государственными бюджетами; решает финансово-экономические и производственно-хозяйственные вопросы организации; обеспечивает соблюдение всех, установленных законодательством, условий и требований по охране труда и здоровья сотрудников и т.п.

В отсутствие генерального директора его обязанности исполняет заместитель директора.

В круг вопросов решаемых «Отделом внешних связей» входит:

- определение спроса потребителей на товары и составление, на основе полученных данных, прогнозов на предстоящие периоды;
- выявление необходимости закупки товаров;
- анализ рынка поставщиков и заключение договоров на поставку товаров.

Задача «Отдела рекламы и PR» состоит в продвижении услуг и товаров организации за счет проведения эффективной рекламной политики.



Рисунок 1 – Организационная структура ООО «Современные смазочные материалы»

Задача «Отдела инноваций» состоит в изучении рынка ИКТ и создании проектов по автоматизации процессов организации.

«Отдел кадров» производит поиск, подбор и обучение персонала, управляет его численностью и составом, начисляет заработную плату в соответствии с действующей в организации системой оплаты труда.

В задачи «Бухгалтерии» входит: учет всех хозяйственных операций, осуществляемых в организации; предоставление подразделениям организации оперативной информации о хозяйственной деятельности организации; своевременное составление официальной налоговой и бухгалтерской отчетности; проведение денежных расчетов с контрагентами и т.п.

Задача работников «Склада» состоит в приемке, размещении, хранении поступающих на склад смазочных материалов и в доставке их покупателям.

Для контроля деятельности склада, в его главе ставится «начальник склада». Основные задачи начальника склада следующие:

- руководит работой склада по приему и хранению товаров, по их размещению с учетом наиболее рационального использования складских площадей, облегчения и ускорения поиска необходимых товаров и т.п.;

- проводит комплектацию заказов и подбор смазочных материалов с учетом сроков их хранения;

- контролирует своевременную подготовку товаров для доставки клиентам;

- проводит проверку отгружаемых товаров на наличие маркировок, ценников и т. п.;

- обеспечивает сохранность потребительских свойств товаров, контролируя соблюдение режимов хранения и правил транспортировки товаров.

«Управление по работе с клиентами» является важным звеном в структуре организации, посредством которого происходит взаимодействие сотрудников организации с клиентами.

В задачи менеджера «Управления по работе с клиентами» входит:

- консультация клиентов по вопросам оказываемых услуг, видам и стоимости товаров и сроках исполнения заказов;

- консультация клиентов по вопросам, связанным с наличием требуемых смазочных материалов на складе;

- оформление заказов и информирование клиентов о ходе их исполнения.

При поступлении от клиента нового заказа, менеджер регистрирует всю необходимую информацию (требуемые товары, информация о клиенте, сроки и адрес доставки и т. п.) и передает ее на «Склад», где далее исполнением заказа занимаются работники склада под контролем «Начальника склада».

В данный момент в организации нет средства, способного автоматизировать процедуру регистрации и исполнения заказов. Это приводит к образованию множества проблем и ошибок учета.

1.2 Существующая технология выполнения исследуемого процесса

Для описания существующей технологии «складского учета» в организации «Современные смазочные материалы», построим структурно-функциональную диаграмму изучаемого процесса по методологии SADT. Среди множества CASE средств, пригодных для реализации таких диаграмм, предпочтение было отдано продукту компании «Computer Associates» - «BPwin». Проектирование функциональных моделей будет производиться на основе стандарта IDEF0.

Контекстная диаграмма процесса «Складской учет» в ООО «Современные смазочные материалы» представлена на рисунке 2. Декомпозиция данного процесса представлена на рисунке 3.

Рисунок 3 показывает, что операции по складскому учету можно разбить на три блока, взаимосвязанных друг с другом:

- прием и оплата заказа;
- подготовка товаров к отправке покупателю;
- доставка товаров покупателю.

Рассмотрим каждый блок в отдельности. Для этого представим декомпозицию процесса «Прием и оплата заказа» (рисунок 4).

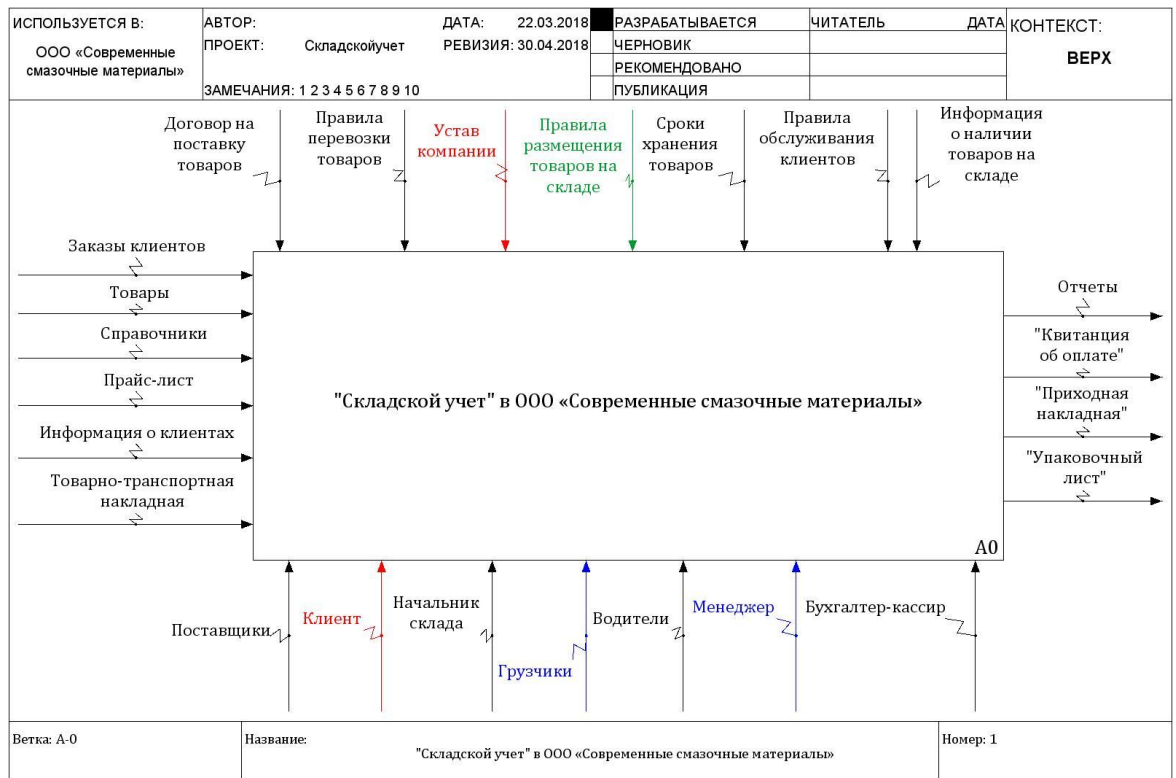


Рисунок 2 – Контекстная диаграмма процесса «Складской учет» в ООО «Современные смазочные материалы». Модель AS-IS

При обращении клиента в организацию «Современные смазочные материалы», менеджер проводит консультацию клиента о наличии товаров на складе и ценах на них, предлагает альтернативные варианты товаров (в случае, если требуемый товар отсутствует). Если клиент готов приобрести предложенные ему товары, то происходит оформление заказа и обсуждение условий доставки (дата, время, адрес, стоимость доставки) и регистрируются сведения о покупателе. На основе данной информации составляется документ «Квитанция об оплате» в двух экземплярах (по варианту для заказчика и исполнителя).

Менеджер должен зарегистрировать заказ в «Журнале заказов» и незамедлительно передать информацию о заказе на «Склад» для дальнейшего исполнения. Номер заказа должен совпадать с номером, указанным в «Квитанции об оплате». В дальнейшем, при помощи этого номера можно отслеживать состояние заказа.

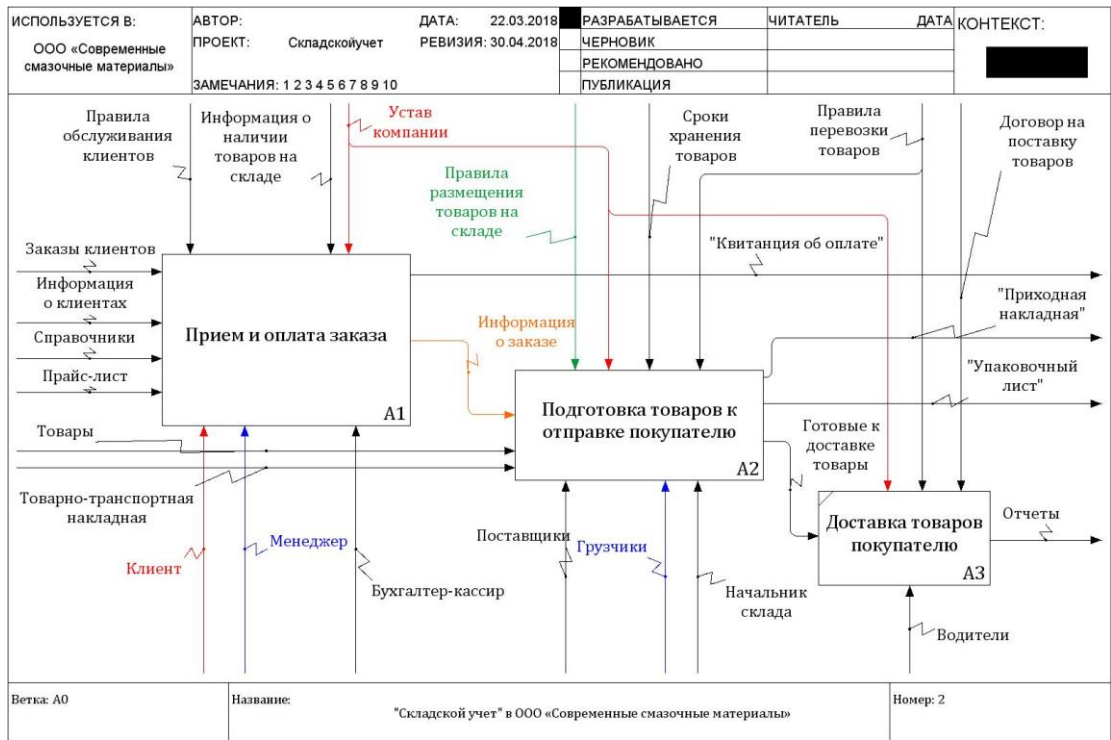


Рисунок 3 – Декомпозиция процесса «Складской учет». Модель AS-IS

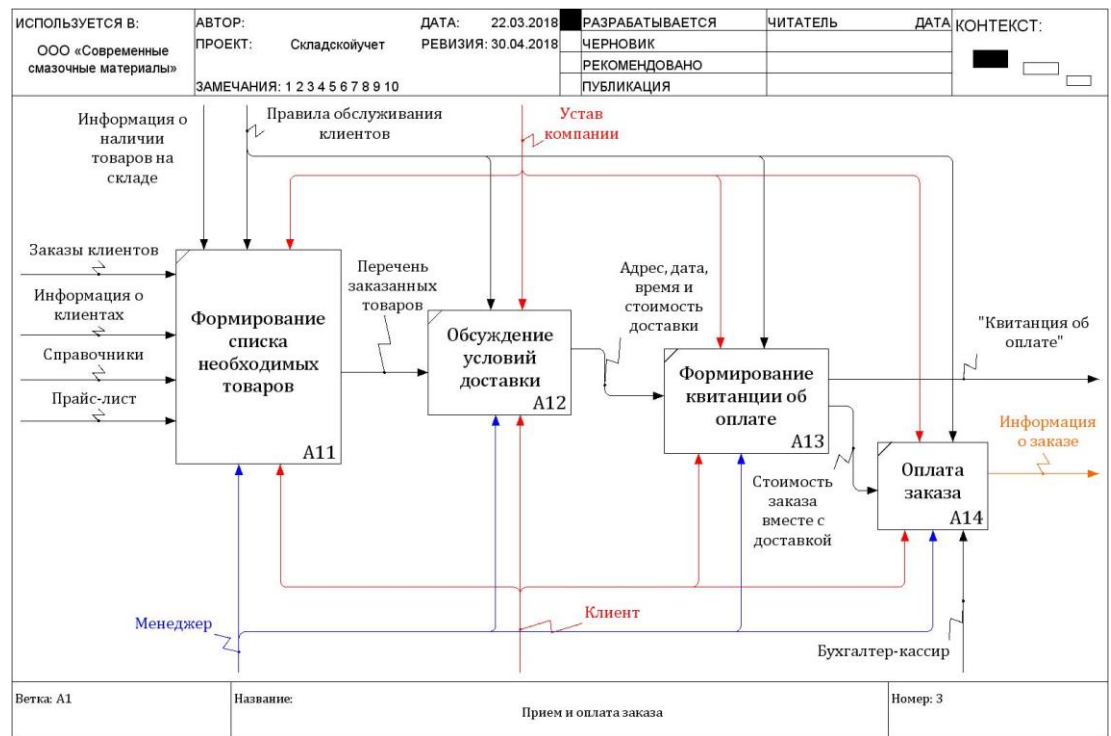


Рисунок 4 – Декомпозиция процесса «Прием и оплата заказа». Модель AS-IS

Завершается данный процесс оплатой заказа и выдачей квитанции об оплате покупателю.

Перейдем к блоку «Подготовка товаров к отправке покупателю», декомпозиция которого представлена на рисунке 5.

Процесс учета товаров на складе начинается с поставки необходимых материалов на склад. При транспортировке и размещении товаров необходимо придерживаться установленных правил перевозки и хранения.

На основе информации о составе заказа, имеющихся в наличии товаров и сроков их хранения осуществляется процесс комплектования заказа. Товары упаковываются и загружаются в транспортное средство для последующей доставки покупателю.

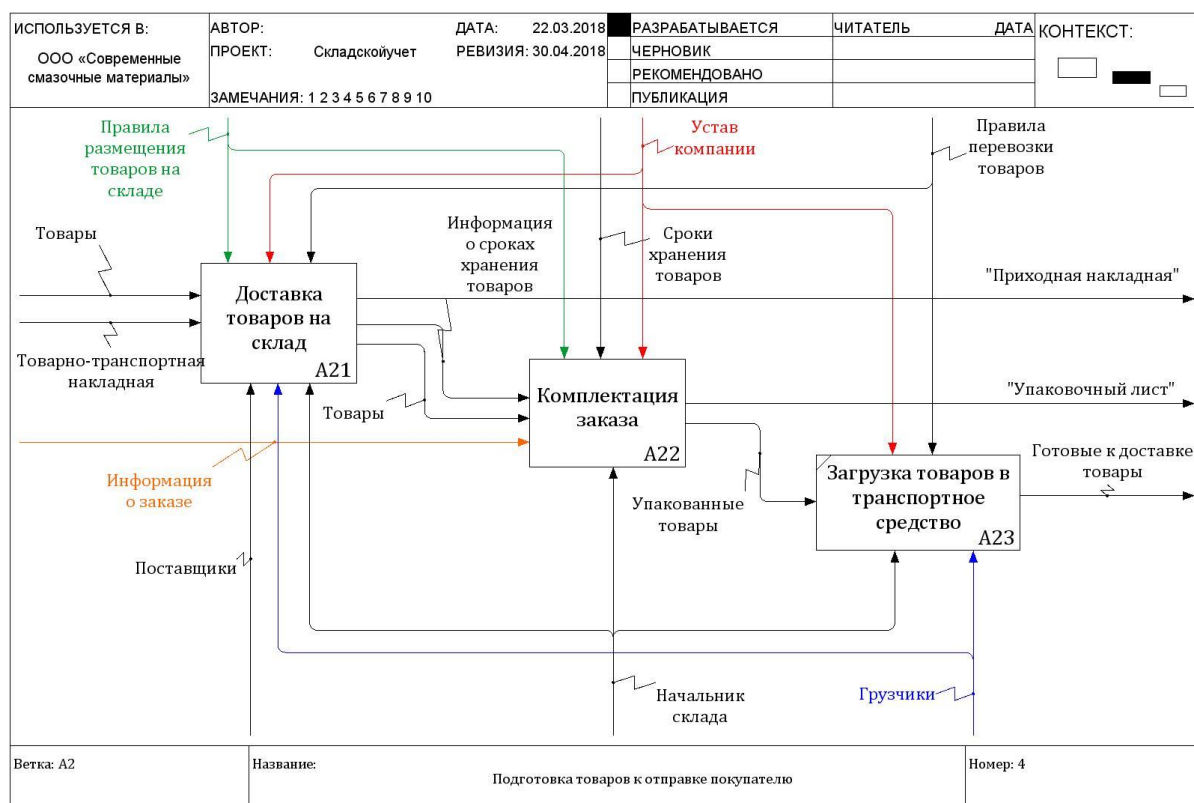


Рисунок 5 – Декомпозиция процесса «Подготовка товаров к отправке покупателю». Модель AS-IS

Начальник склада контролирует целостность заказа (т.е. ни одна единица товара, оплаченная покупателем, не должна быть упущена) и все процедуры по учету выбывшего товара.

Декомпозиция процесса «Доставка товаров на склад» представлена на рисунке 6.

Процесс поставки товаров на склад можно разбить на 3 операции:

- разгрузка товаров, осуществляемая усилиями грузчиков;
- размещение товаров (в зависимости от категории, типа и сроков хранения товара);
- регистрация данных о сроках хранения и размещения товаров (данная операция считается наиболее важной, если учитывать, что срок хранения ограничен).

На основе перечисленных сведений формируется документ «Приходная накладная».

Декомпозиция процесса «Комплектация заказа» представлена на рисунке 7.

Процедура по комплектованию заказа является очень трудоемкой, учитывая масштабы предприятия. Если не иметь под рукой удобный механизм по обеспечению «Начальника склада» актуальной информацией о сроках хранения товаров и их местоположении, поиск и комплектование заказов может затянуться надолго. Так, схема, изображенная на рисунке 7, показывает, процедура по комплектованию заказа состоит из 4 шагов, на каждом из которых требуется внимательно изучить исходную документацию. При этом важно учесть, что информация о месте хранения товаров, указанная в приходной накладной, может оказаться неактуальной (если товар был перемещен в другую ячейку).

Процесс комплектации заказа заканчивается формированием документа «Упаковочный лист». Данный документ содержит список товаров, которые необходимо укомплектовать в рамках одного заказа. Отчет необходим фасовщикам, чтобы те имели перед собой перечень товаров (код, название товара, дата производства) и место их расположения (адрес ячейки хранения).

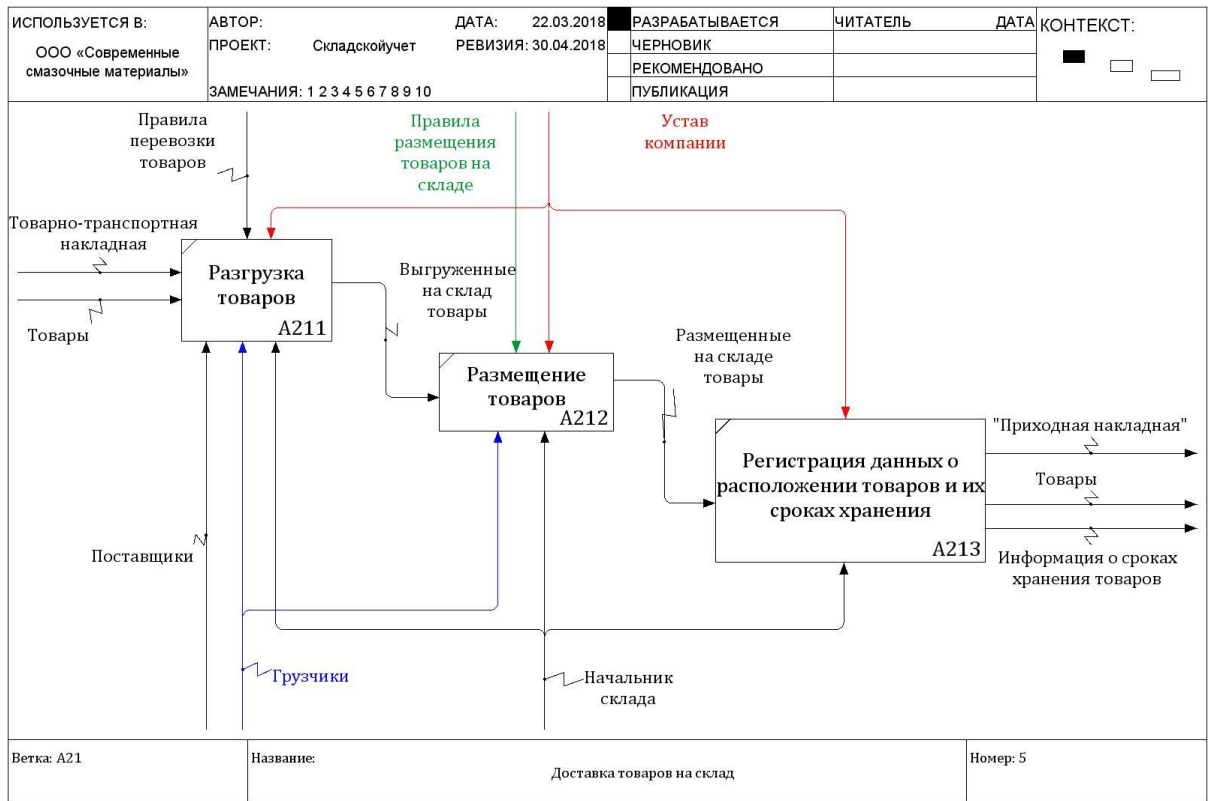


Рисунок 6 – Декомпозиция процесса «Доставка товаров на склад».

Модель AS-IS

Диаграмма потоков данных показывает, что при движении информации от менеджера к сотрудникам склада, информация неоднократно дублируется, что в несколько раз замедляет процесс исполнения заказов.

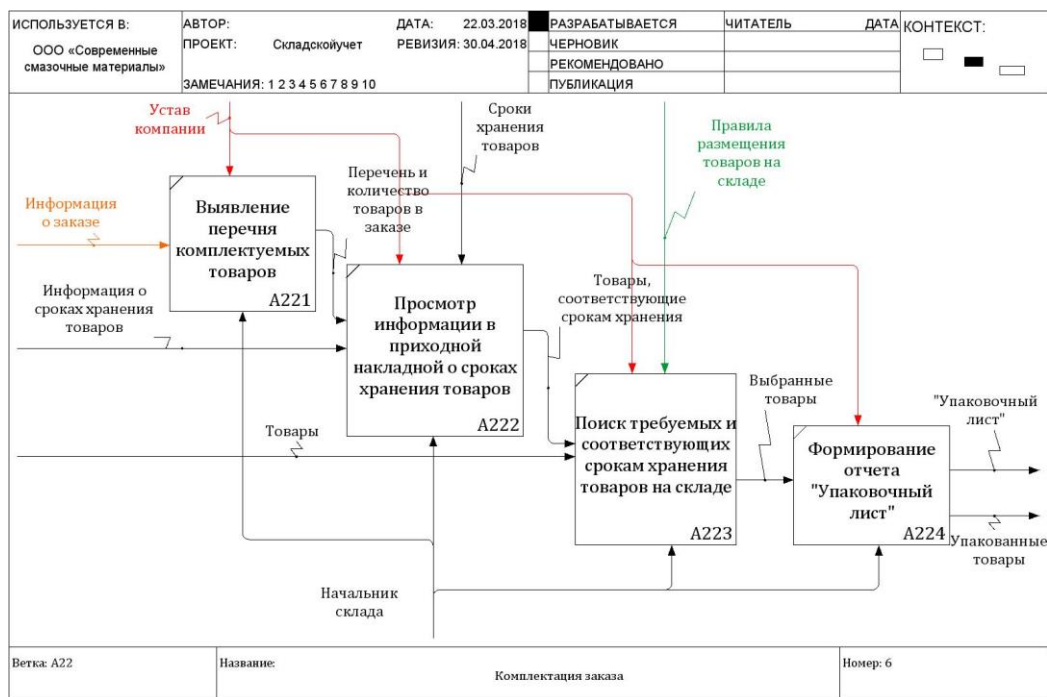


Рисунок 7 – Декомпозиция процесса «Комплектование заказа».

Модель AS-IS

1.3 Проблемный анализ ключевых процессов организации

В результате проведенного анализа на основе моделирования процессов исследуемой организации, были выявлены следующие недостатки складского учета:

- нет доступа к актуальной информации о состоянии склада;
- длительные процедуры поиска интересующей информации;
- отсутствие контроля сроков исполнения заказов;
- длительные процедуры размещения товаров на складе, связанные с поиском свободных мест и ячеек;
- высокая трудоемкость и низкая оперативность обработки информации;
- отсутствие единых стандартов сбора и регистрации информации, приводящих к дублированию или потере данных о запросах или же товарах;
- низкий уровень безопасности хранимых данных, доступ к модификации и уничтожению которых может получить любой сотрудник организации;

- большой объем бумажных носителей информации, занимающих немалую долю офисного пространства;
- отсутствие механизмов контроля работы сотрудников склада.

Важным недостатком складского учета является отсутствие механизмов, способных обеспечить всех сотрудников организации, участвующих в оказании складских услуг, актуальной информацией о состоянии склада, сроках хранения товаров и местах их расположения.

Задача менеджеров – проводить консультацию клиентов по вопросам наличия или отсутствия товаров на складе организации, принимать заказы и вести контроль сроков их исполнения. Для этого, менеджер должен иметь оперативный доступ к перечисленным видам информации, иначе операции по поиску и уточнению интересующей клиента информации занимают большую часть рабочего времени менеджеров. При этом приходится постоянно отвлекать от работы сотрудников склада (для уточнения актуальной информации о наличии интересующего товара на складе.).

Что касается работы «Склада», то здесь отсутствие информационной системы негативным образом сказывается, как на качестве, так и на количестве проводимых работниками склада работ, в связи с чем, зачастую, сроки исполнения заказов превышают желаемые.

Сложность учета состоит в следующем:

- при поступлении товаров от поставщиков, сотрудникам склада приходится искать свободные места на складе, обегая его несколько раз;
- при размещении товаров на складе, информация о местоположении каждого товара регистрируется на бумажных носителях, и, в случае перемещения товара, факт перемещения фиксируется на другом бумажном носителе, что крайне неудобно;
- при упаковке товаров, фасовщики теряются в бумажных отчетах, хранящих противоречивую информацию;

– при обращении менеджера с вопросом о наличии определенного товара на складе, сотрудникам приходится бегать по складу и подсчитывать количество запрашиваемого товара.

Отсутствие ИС также усложняет контроль и оценку работы сотрудников организации. В организации нет механизмов, способных выявить факты хищения денежных средств либо преднамеренного завышения цен на услуги с целью личной наживы.

В связи с этим, организации необходима информационная система, способная устранить существующие проблемы учета, повысить качество и скорость обработки информации на складе и решать следующие задачи:

- автоматизированный учет поставок товаров на склад;
- автоматизированное размещение товаров на складе с учетом информации о количестве свободных мест в ячейках склада;
- автоматический расчет количества свободных и занятых мест в ячейках склада;
- автоматизированный процесс упаковки товаров, согласно запросам, поступающим из магазина;
- автоматизированное формирование документов: «Приходная накладная», «Квитанция об оплате» и «Упаковочный лист»;
- автоматизированный расчет остатка товаров каждого вида на складе;
- автоматизированный процесс регистрации сроков годности товаров на основе информации о сроках хранения товаров каждой категории;
- автоматическое выявление товаров с истекшим сроком годности;
- автоматическое списание недоброкачественных товаров.

1.4 Постановка задачи на проектирование и информатизацию исследуемого процесса

Проектируемая информационная система должна быть рассчитана на автоматизацию работы, как сотрудников склада, так и менеджеров, принимающих заказы. Для этого требуется создание трех модулей программы: «Администратор», «Начальник склада» и «Менеджер». Вход в систему должны иметь только зарегистрированные пользователи, прошедшие идентификацию по логину и паролю. Логин и пароль задаются Администратором и, при необходимости, изменяются пользователем в «Личном кабинете». Такой принцип работы системы обеспечит защиту данных от недоброжелателей.

Основными функциональными возможностями модуля «Администратор» должны быть:

- учет сотрудников;
- ведение справочников;
- добавление новых пользователей в систему;
- ведение поиска и фильтрации записей;
- формирование и вывод на печать итоговых документов и отчетов;
- формирование статистики работы организации.

Функция учета сотрудников должна быть реализована через справочник «Сотрудники», где будет храниться следующая информация о сотрудниках организации: ФИО, должность, дата рождения, адрес, номер телефона, e-mail, примечания. Также в данном справочнике должны задаваться логины и пароли для входа в систему.

Справочник «Склады» будет содержать перечень складов организации, их место расположения и начальников, находящихся во главе этих складов.

Справочник «Товары» позволит вести учет ассортимента товаров организации с указанием их категории, кода товара, единицы измерения, и цены за единицу товара.

Добавление новых пользователей в систему означает, что лишь у Администратора системы должен быть доступ к этой функции, чтобы избежать несанкционированного доступа к данным путем самостоятельной регистрации в системе.

Возможность ведения поиска записей позволит сократить время пользователя на поиск необходимой информации, а функция фильтрации - избавит пользователя от избыточной информации на экране. Функция фильтрации особенно полезна при формировании отчетов.

Формирование и вывод на печать итоговых документов и отчетов позволит проводить быстрый анализ результатов деятельности организации за текущих и прошедшие периоды времени.

В системе должны быть предусмотрены шесть основных видов отчетов:

- «Отчет по завершенным заказам»;
- «Отчет по реализованным товарам за определенный период»;
- «Отчет о поступлении товаров на склад за определенный период»;
- «Отчет об отсутствующих товарах на текущую дату»;
- «Отчет о просроченных товарах».

Функция по формированию статистики работы организации должна отражать основные результаты деятельности организации за определенный период (как в графическом, так и табличном представлении).

Таковыми показателями являются:

- общее количество принятых заказов за период;
- общая стоимость принятых заказов за период (в руб.);
- среднее количество принимаемых заказов в день;
- средняя стоимость заказов в день (в руб.);
- средний рок исполнения заказов (в днях).

Формулы расчета данных показателей приведены ниже.

Общее количество принятых заказов за период (K_3) равно сумме этих заявок.

Общая стоимость принятых заказов за период (S_3) равна суммарной стоимости принятых заявок.

Среднее количество принимаемых заказов в день рассчитывается по формуле 1.

$$CpK_3 = \frac{K_3}{n}, \quad (1)$$

где n – количество дней в выбранном временном интервале.

Для расчета средней стоимости заказов в день используется формула 2.

$$CpS_3 = \frac{S_3}{n}, \quad (2)$$

где n – количество дней в выбранном временном интервале.

Средний срок исполнения заказов (T_{cp}) вычисляется по формуле 3.

$$T_{cp} = \frac{(Z1D_{заверш} - Z1D_{нач}) + (Z2D_{заверш} - Z2D_{нач}) + \dots + (ZnD_{заверш} - ZnD_{нач})}{K_3} \quad (3)$$

Где $D_{заверш}$ – дата завершения заказа;

$D_{нач}$ – дата создания заказа;

$Z1, Z2, \dots, Zn$ – перечень заказов за выбранный временной интервал.

Сотрудникам, имеющим должность «Менеджер» должны быть доступны следующие функции программы:

- доступ к информации об остатках товаров на складе;
- регистрация заказов;
- регистрация клиентов;
- отслеживание статуса заказа;
- автоматический расчет стоимости заказа;
- автоматическое формирование документа «Квитанция об оплате»;
- смена логина и пароля.

Доступ к информации об остатках товаров на складе должен, в случае обращения клиента с вопросом о наличии определенного товара, сократить время обслуживания с 10 минут до 10 секунд. Клиенту, в таком случае, не придется ждать, пока менеджер обзвонит сотрудников склада для уточнения информации, т. к. вся актуальная информация о состоянии склада будет представлена в рабочем интерфейсе менеджера.

Регистрация заказа предполагает оформление заказа в электронном «Журнале заказов», с указанием следующих данных:

- номер и дата заказа (должны проставляться автоматически);

- информация о клиенте (название, телефон, электронный адрес, имена представителей);
- условия доставки (дата, время, адрес, стоимость);
- перечень заказанных товаров;
- статус заказа;
- и т. п.

Регистрация клиентов предполагает, что менеджер может вносить изменения в справочник «Клиенты», добавляя новых, либо редактируя данные о существующих клиентах.

Все заказы, зарегистрированные в системе, должны иметь определенный статус, отражающий ход их исполнения. Целесообразно использование системы из 5 статусов: «Новый», «В работе», «Доставка», «Завершен», «Отказ».

Статус «Новый» автоматически присваивается новым заказам. Заказы, по которым уже ведется комплектация товаров, должны иметь статус «В работе». Если «Заведующий складом» фиксирует завершение процедуры по комплектации и загрузке товаров в транспортное средство, то заказу присваивается статус «Доставка». Когда заказ будет доставлен клиенту, его статус принимает значение «Завершен». Статус «Отказ» имеют заказы, отменены в процессе оформления.

Отслеживание статуса заказов означает, что менеджер в своем интерфейсе должен видеть, на каком этапе находится исполнение заказа. Это очень важная функция, позволяющая менеджеру, не отвлекая от работы сотрудников склада, предоставить клиенту всю необходимую информацию о ходе исполнения оставленного им заказа.

Автоматический расчет стоимости заказа предполагает, что система должна автоматически рассчитывать стоимость заказа на основе информации о: общей стоимости заказанных товаров (S_t) и стоимости доставки (S_d) (формула 4).

$$S = S_d + S_t, \quad (4)$$

где S – стоимость заказа;

S_d – стоимость товаров;

S_t – стоимость доставки.

Общая стоимость заказанных товаров рассчитывается следующим образом (формула 5):

$$S_t = C_{t1} * k_1 + C_{t2} * k_2 + \dots + C_{tn} * k_n, \quad (5),$$

где $C_{t1}, C_{t2} \dots C_{tn}$ – цены на заказанные товары;

$k_1, k_2 \dots k_n$ – количество заказанного товара;

n – номер заказанного товара.

Автоматическое формирование документа «Квитанция об оплате» предполагает, что система, на основе заранее заполненных справочников, и зарегистрированных менеджером данных должна автоматически формировать указанный документ.

Для сотрудников, имеющих должность «Начальник склада», должен быть настроен специальный интерфейс, который позволит ему легко и быстро проводить учет складских операций.

Основными функциональными возможностями модуля «Начальник склада» будут:

- регистрация информации о поступивших на склад товарах и формирование документа «Приходная накладная»;
- мониторинг состояния склада;
- учет списанных товаров;
- комплектование заказов и формирование документа «Упаковочный лист»;
- регистрация перемещения товаров в другую ячейку;
- регистрация товаров, выгруженных со склада;
- просмотр заказов;
- учет заказов, требующих доставки;
- учет заказов, отправленных покупателю;
- смена логина и пароля.

Регистрация информации о поступивших на склад товарах предполагает, что «Начальник склада» должен зафиксировать в системе весь перечень принятых поставщиков материалов, указывая при этом его код, количество, дату производства, место расположения и общую стоимость поступивших товаров. На основе данной информации и информации о сроке хранения каждого вида товара система должна автоматически рассчитывать срок годности товара.

Общая стоимость поступивших товаров определяется по формуле 6.

$$S_{\text{пост}} = C_{t1} * k_1 + C_{t2} * k_2 + \dots + C_{tn} * k_n, \quad (6)$$

где $C_{i1}, C_{i2} \dots C_{im}$ – цены на поступившие товары;

$k_1, k_2 \dots k_n$ – количество поступившего товара;

n – номер поступившего товара.

На основе информации о поступивших товарах формируется документ «Приходная накладная».

Мониторинг состояния склада позволяет в момент поступления нового заказа произвести подбор таких товаров, срок хранения которых максимально приближен к текущей дате (тем самым можно избежать возникновения случаев просрочки товаров).

Учет списанных товаров должен позволить изъять со склада и зарегистрировать в системе перечень товаров с истекшим сроком годности.

Комплектование заказов означает, что начальнику склада необходим интерфейс, который поможет легко и быстро составить документ «Упаковочный лист», где будет представлена следующая информация для фасовщиков (код товара, наименование, дата производства, требуемое количество, дата производства, номер ячейки хранения).

Регистрация перемещения товара в другую ячейку - известно, что каждый вид товара на складе имеет свою ячейку хранения с уникальным номером. Данная ячейка хранения указывается в карточке учета товаров в момент поставки товаров на склад. Этот метод позволяет быстро отыскать нужный товар. Однако, при перемещении товаров с одной ячейки в другую, возникает необходи-

мость обновить его адрес. Проектируемая система должна позволить легко и быстро проводить данную операцию.

Регистрация товаров, выгруженных со склада, означает, что система должна автоматически отмечать, что определенный товар, в определенном количестве, с определенным сроком годности, расположенный в определенной ячейке был отправлен клиенту.

Функция просмотра заказов позволяет просматривать новые заказы и планировать работу по ним.

Учет заказов, требующих доставки, необходим для контроля сроков исполнения заказов.

Учет заказов, отправленных покупателю, позволяет своевременно закрыть заказ после его доставки клиенту.

Функция по смене логина и пароля предполагает изменение параметров входа в систему.

1.5 Анализ существующих разработок

Современный рынок программных продуктов предлагает множество решений для автоматизации работы склада. Наибольшей популярностью среди них пользуются системы «Автоматизация склада» от компании «Abmcloud», «1С-Логистика: Управление складом 3.0» и «Мой склад» от компании «Логнекс». Данные системы находятся в свободном доступе и обладают минимальным набором функций.

Для оценки особенностей перечисленных программ и соответствия их функциональных возможностей требованиям, предъявленным организации «Современные смазочные материалы», произведем сравнительный анализ вышеуказанных систем (таблица 1).

Таблица 1 – Сравнительная характеристика программных продуктов для автоматизации работы склада

Функциональные возможности	Автоматизация склада	1С-Логистика	Мой склад
Автоматизация складских операций:			
– прием товаров на склад	+	+	+
– комплектация заказов	-+	+	+
– учет в месторасположения товаров	+	-	+
– учет внутренних перемещений товаров (в том числе между складами)	+	-	-
– отгрузка товаров	+	+	+
– контроль сроков хранения товаров	+	+	+
– учет списаний	+	+	+
Учет работников склада	+	-	+
Учет клиентов	-	-	-
Формирование итоговых документов	+	+	+
Формирование отчетов	+	+	+
Ведение справочников	+	+	+

Анализ приведенных в таблице данных, свидетельствует, что все перечисленные системы обладают минимальным набором функций, необходимых для автоматизации работы склада, но у каждой из них есть свои недостатки.

Рассмотрим, какая из рассматриваемых программ больше остальных удовлетворяет требованию «цена-качество».

В продукте компании «1С» – «1С-Логистика: Управление складом 3.0» отсутствуют базовый набор функций, необходимых для работы современного склада. Интерфейс программы имеет сложную и непонятную структуру. Для внедрения такой системы придется нанимать программиста 1С и дорабатывать проект под нужды организации. С учетом того, что базовая версия программы стоит 37800 руб., стоимость решения будет формироваться из общей стоимости программы + зарплата программисту (где-то от 5600 до 17000 руб.) [3]

«Автоматизация склада» от компании «Abmcloud» имеет удобный интерфейс, но в базовой версии нет разграничения прав доступа к определенным операциям, т.е. любой сотрудник может скорректировать данные в справочниках и т.п. Иначе говоря, внедряя такую систему, нужно полагаться на добросовестное отношение сотрудников организации. Стоимость решения для одного рабочего места составляет 3200 руб. в год [1].

«Мой склад» от компании «Логнекс» является серьезным решением с большим набором функций. Базовая версия программы выдается бесплатно и рассчитана на 1 пользователя. Полная версия программы, в зависимости от количества необходимых рабочих мест и заказов, обойдется от 1200 руб. в месяц [2].

Анализ показывает, что информационные системы «Автоматизация склада» и «Мой склад» практически удовлетворяют требованиям, предъявленным ООО «Современные смазочные материалы». Каждая из этих систем принесет немалую пользу организации после ее внедрения, однако нельзя забывать, что готовые решения не учитывают специфику работы конкретной организации, в связи с чем, их не всегда удастся подстроить под процессы любого предприятия.

Обычно есть два выхода из таких ситуаций: первый – подогнать процессы организации под требования системы; второй – доработать внедряемую систему под нужды организации.

И в первом и во втором случае есть свои минусы. Так, например, изменения в процессах организации очень часто сопровождаются негативным настроем сотрудников, не желающих менять привычную схему работы. Что касается второго варианта, то тут две проблемы, вытекающие одна из другой. Первая – компания-разработчик может отказаться от «допиливания» проекта и предложит использовать систему в готовом виде. Вторая проблема – большие финансовые затраты, необходимые для доработки системы.

Помимо этого, есть риск, что даже после доработки проекта, готовое решение не будет приносить желаемый результат в связи с растущими требованиями бизнеса – появляются новые идеи, задачи, функции, которые организация захочет видеть в используемой системе. К тому же внедрение готовых решений не гарантирует обеспечение безопасности данных, т.к. они находятся на стороннем сервере.

Итоги анализа существующих на отечественном рынке программных продуктов для автоматизации работы склада доказывают, что наиболее рацио-

нальным решением для ООО «Современные смазочные материалы» является разработка собственной информационной системы, специально ориентированной под ее процессы.

1.6 Предлагаемая технология выполнения исследуемого процесса

Предлагаемая технология выполнения процесса «складской учет» на предприятии «Современные смазочные материалы» представляет собой описание схемы работы сотрудников организации при осуществлении складских операций после внедрения проектируемой информационной системы.

Моделирование рассматриваемого процесса будет производиться при помощи программного инструментария «BPrwin» на основе стандарта IDEF0.

Как было отмечено ранее, разрабатываемая информационная система не повлечет за собой серьезных изменений в процессах организации, а лишь позволит автоматизировать и упростить ввод и исполнение многих операций.

На рисунке 8 изображена схема потоков данных операции «Прием и оплата заказа» согласно предлагаемой технологии.

Согласно предлагаемой технологии, менеджеру не придется вручную формировать документ «Квитанция об оплате». Система сформирует его автоматически на основе информации о заказе.

На рисунке 9 изображена декомпозиция процесса «Доставка товаров на склад».

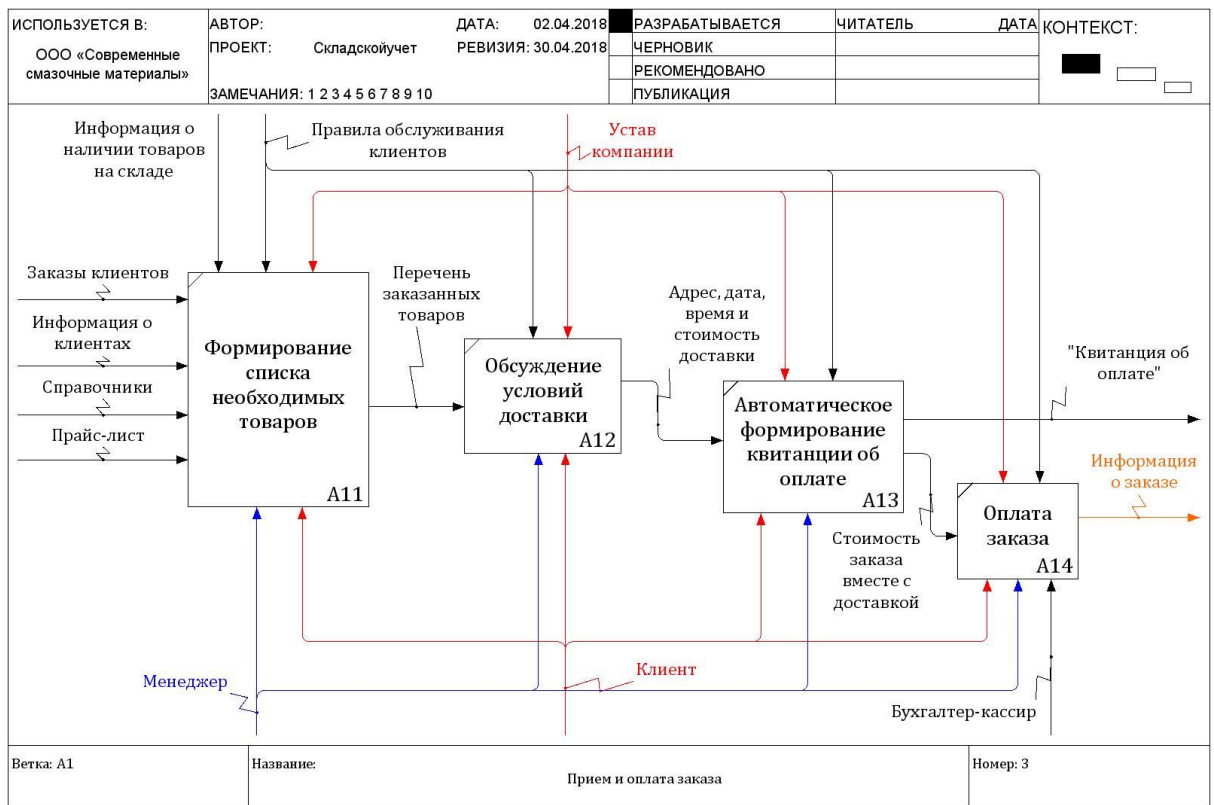


Рисунок 8 – Декомпозиция процесса «Прием и оплата заказа». Модель ТО-ВЕ

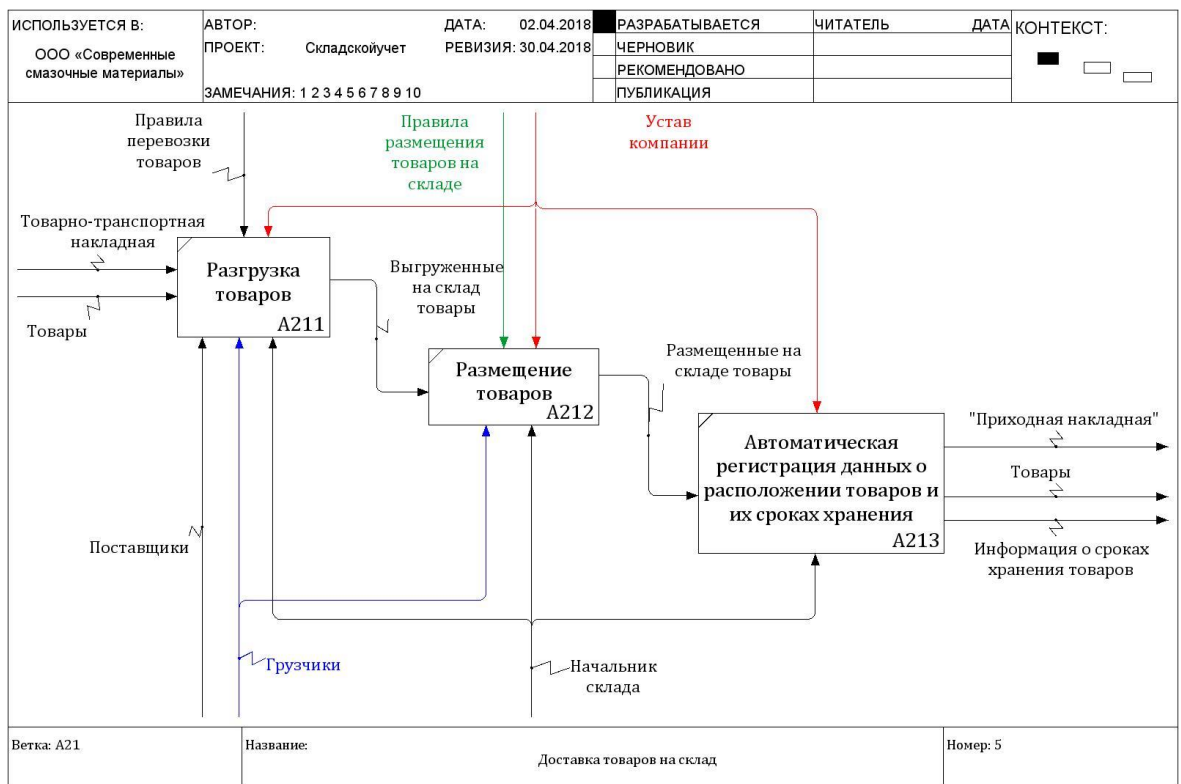


Рисунок 9 – Декомпозиция процесса «Доставка товаров на склад».

Модель ТО-ВЕ

Схема показывает, что процесс размещения товаров на складе упростился за счет автоматического подбора свободного места на складе и присвоении этого места товару в качестве адреса, избавив «Начальника склада» от необходимости искать свободные ячейки, «разгуливая по складу».

Декомпозиция процесса «Подготовка товаров к отправке покупателю» (рисунок 10) также отражает изменения в процессе поиска и упаковки запрашиваемых товаров.

Согласно новой технологии, сотрудникам склада нет необходимости изучать исходную документацию, чтобы найти информацию о месте расположения товаров. Система сама произведет подбор необходимых товаров, сформирует документ «Упаковочный лист» и выдаст результат на экран монитора или на печать (для фасовщиков), освободив пользователя от трудоемкой процедуры, отнимающей драгоценное рабочее время.

Модель потоков данных, построенная по предлагаемой технологии показывает, что реализация процесса «Складской учет» заметно проще по сравнению с существующей технологией. А именно:

- менеджеру не нужно вручную формировать документ «Квитанция об оплате»;
- менеджеру нет необходимости передавать информацию о заказе на склад, достаточно просто зафиксировать заказ в системе - сотрудники склада тут же его увидят и приступят к исполнению;
- при поступлении товаров от поставщиков, начальнику склада не нужно вручную создавать документ «Приходная накладная»;
- начальнику склада нет необходимости искать продукцию, у которой срок хранения наиболее близок к текущей дате – система сама выдаст перечень подходящих товаров и информацию об их местоположении;
- начальнику склада нет необходимости составлять список комплектующих товаров для фасовщиков – система сама подготовит документ «Упаковочный лист», начальнику склада останется лишь его распечатать.

Предложенная технология осуществления изучаемых бизнес-процессов

позволит: устранить существующие недостатки складского учета; повысить скорость и качество обработки заказов; устранить ошибки связанные с человеческим фактором; сделать работу сотрудников организации более прозрачной и т. п.

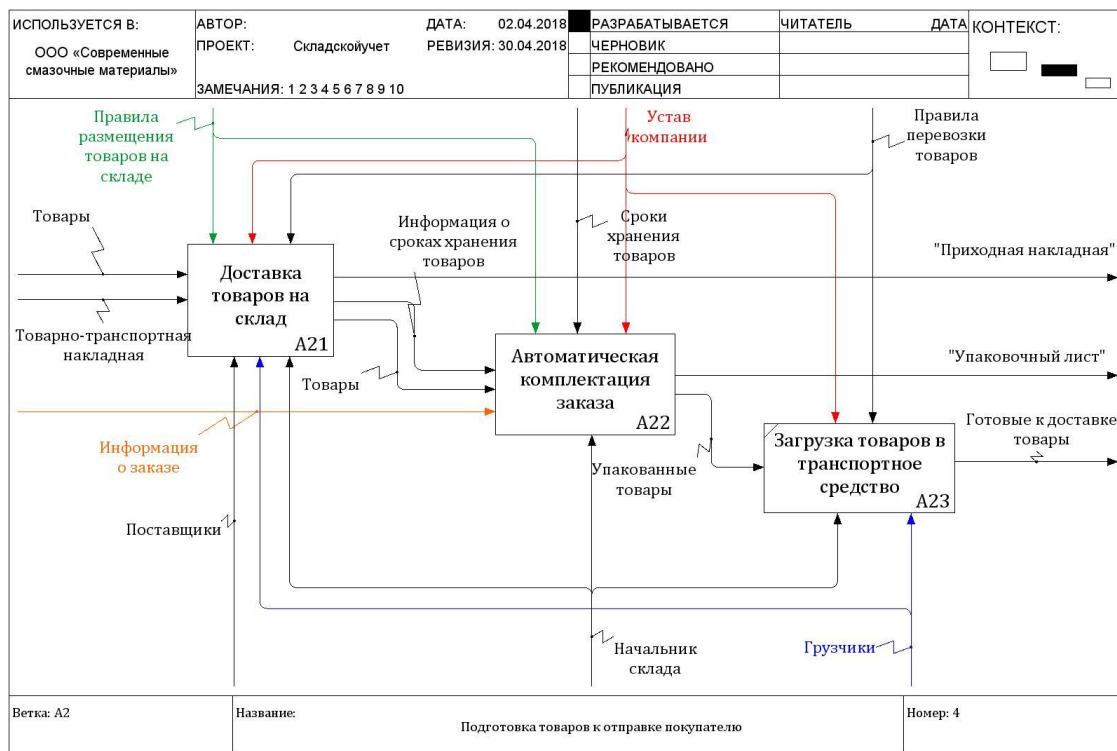


Рисунок 10 – Декомпозиция процесса «Подготовка товаров к отправке покупателю». Модель TO-VE

1.7 Описание видов обеспечения исследуемого процесса

Организационное обеспечение включает в себя методы и средства взаимодействия сотрудников организации с аппаратными средствами и между собой в процессе работы с проектируемой информационной системой.

Взаимодействие пользователя с компьютерами и периферийными устройствами будет осуществляться посредством дружественного, графического пользовательского интерфейса и элементов управления, расположенных на самих технических устройствах (кнопки включения и отключения и т.п.).

Средствами, обеспечивающими взаимодействие пользователя с техниче-

скими устройствами, могут служить экран монитора → операционная система → проектируемая ИС → окна проектируемой ИС.

Взаимодействие пользователя с проектируемой ИС может осуществляться посредством элементов управления, расположенных в окнах программы (главное меню, контекстное меню, кнопки, вкладки) и «горячих клавиш».

В качестве метода взаимодействия сотрудников с аппаратными средствами может служить совокупность действий пользователя над информационной системой для решения поставленных задач. Такими методами могут быть операции по добавлению, редактированию, удалению, обновлению, фильтрации, печати либо поиска данных и т.п.

В состав организационного обеспечения ИС входят структурные подразделения организации (пользователи), осуществляющие поддержку ее работоспособности и документы, регламентирующие порядок эксплуатации и развития системы. Такими документами могут служить «Техническое задание на разработку проекта», «Инструкции пользователей ИС» и иные документы, которые будут представлены в пункте «Правовое обеспечение» данной работы.

Для работы с системой и поддержания ее работоспособности, необходимы следующие лица:

- администратор БД – лицо, ответственное за сопровождение данных, определение уровней доступа к ним, добавление/удаление пользователей БД и т.п. Обычно администраторы БД участвуют в процессе разработки структуры БД;

- администратор ИС – лицо, обладающее правами по: редактированию справочников системы; просмотру и выводу на печать отчетов, формируемых системой; изменению параметров доступа пользователей к системе; добавлению/удалению пользователей ИС;

- конечные пользователи - лица, работающие с ИС и БД через пользовательский интерфейс для решения поставленных задач.

Программное обеспечение.

Проектируемая система будет состоять из трех модулей: «Администратор», «Начальник склада» и «Менеджер».

Минимальное количество пользователей системы должно равняться 4 сотрудникам (по одному для каждого модуля + Администратор БД). Максимальное количество сотрудников, использующих систему, не ограничено.

Администратор БД должен обладать знаниями:

- проектирования и сопровождения БД;
- обеспечения целостности и безопасности хранимых данных;
- определения уровней доступа к данным;
- добавления/удаления пользователей БД;
- резервного копирования и восстановления данных.

Остальные пользователи, работающие с системой должны обладать знаниями в исследуемой предметной области и навыками пользования компьютером на уровне продвинутого пользователя.

Надежность системы должна обеспечиваться путем:

- применения программно-аппаратных средств, соответствующих классу решаемых задач;
- соблюдения правил эксплуатации и технического обслуживания оборудования;
- предварительного обучения пользователей при работе с программой.

В случае возникновения сбоев в электроснабжении, восстановление системы должно осуществляться сразу после перезагрузки операционной системы и запуска исполняемого файла.

Все ошибки, ведущие к сбоям в программе и неточности в расчетах, выявленных на этапах тестирования и внедрения программного продукта, будут устраняться Разработчиком по мере их обнаружения.

Проектируемая система должна соответствовать следующим требованиям к эргономике и технической эстетике:

- весь текст, отображаемый на экране, должен быть представлен на русском языке;

– функции по работе с базой данных (добавить, изменить и удалить) должны быть легкодоступными;

– ИС должна иметь дружелюбный, интуитивно-понятный интерфейс;

– меню программы не должно быть перегружено (не более 3 уровней);

– интерфейс программы должен быть выполнен в едином стиле с использованием приятной цветовой гаммы, не раздражающей зрение пользователя. Допускается такое сочетание цветов, как белый, серый, голубой или белый, серый, зеленый;

– стиль и размер шрифта должен быть читаемый (желательно использовать тип шрифта «Times New Roman» либо «Cambria» размером в 12-14 пт.);

– система должна предупреждать пользователя о выполнении им некорректных действий над программой путем выдачи на экран монитора сообщения с рекомендациями по устранению возникшей проблемы.

Требование к информационной безопасности должно быть обеспечено путем:

– предоставления доступа к системе пользователям, прошедшим идентификацию по логину и паролю (логин и пароль задаются Администратором и, при необходимости, изменяются пользователем в «Личном кабинете»);

– использованием лицензионного программного обеспечения;

– регулярной проверкой ПО на наличие компьютерных вирусов.

Техническое обеспечение – это комплекс технических средств, необходимых для функционирования информационной системы.

Для обеспечения работы проектируемой информационной системы необходим IBM-совместимый персональный компьютер, включающий в себя:

– процессор «Intel Core i5-4200U» с частотой 1.6 ГГц, не менее;

– оперативную память объемом, 4 Гб, не менее;

– жесткий диск объемом 1 Тб, не менее;

– операционную систему «Windows 7», не ранее;

– манипулятор «мышь»;

– стандартная клавиатура.

Требования к программному обеспечению: сервер баз данных «MySQL» и стандартный набор офисных программ «MicrosoftOffice».

Правовое обеспечение ИС – совокупность правовых норм, регламентирующих порядок проектирования, реализации и использования информационных систем. Правовое обеспечение устанавливает порядок получения, изменения и использования данных в разрабатываемой ИС.

В состав правового обеспечения информационных систем входят:

- законы, указы, постановления и иные приказы органов государственных власти федерального и местного уровня;
- установленные законом стандарты на разработку ИС и баз данных;
- договоры, регламентирующие: отношения между разработчиком и заказчиком в процессе разработки, внедрения, эксплуатации ИС; порядок внесения поправок в систему.

В нашей стране основным законом, регулирующим отношения в сфере информации и информатизации, является Федеральный закон от 27.07.2006 N 149-ФЗ «Об информации, информационных технологиях и защите информации». Перечень иных ФЗ и стандартов представлен на рисунке 11.



Рисунок 11 – Правовое обеспечение ИС

При проектировании информационной системы и разработке проектной документации разработчик обязан руководствоваться всеми требованиями, представленными в перечисленных выше документах.

2 Проектная часть

2.1 Обоснование выбора средств разработки

Процесс проектирования информационной системы начинается с выбора программных средств ее реализации: среды разработки и системы управления базами данных (СУБД).

В наше время многообразие языков программирования и средств разработки баз данных дает возможность любому разработчику, на основе личных предпочтений и функциональных возможностей, закладываемых в проектируемую информационную систему, производить выбор той среды разработки, которая окажется наиболее приемлемой для решения поставленных задач (начиная с разработки простых приложений и заканчивая серьезными программными продуктами для крупных корпораций).

Обоснование выбора СУБД.

Для выбора системы управления базами данных необходимо провести сравнительный анализ наиболее популярных СУБД («MS Access», «Visual FoxPro», «MySQL» и «Oracle») на предмет соответствия основным функциям, которые она должна выполнять: создание БД; манипулирование данными; физическое размещение и сортировка записей; защита от сбоев, поддержку целостности данных и их восстановление; безопасность данных [28].

СУБД «MS Access» превосходит своих оппонентов по простоте и удобству использования, но значительно уступает в средствах администрирования, быстродействии (особенно при больших объемах данных) и безопасности хранимых данных. «MS Access» рекомендуется использовать при проектировании несложных программ с ограниченным объемом записей [28].

«Visual FoxPro» – СУБД, работающая на собственном языке программирования. Недостатки системы такие же, как и у «MS Access». Такую СУБД це-

лесообразно использовать при создании сложных приложений с количеством записей не более миллиона [28].

«MySQL» – СУБД, полностью удовлетворяющая требованиям безопасности, надежности, компактного хранения и скорости обработки данных с мощными средствами администрирования БД. В отличие от предыдущих СУБД, «MySQL» позволяет хранить колоссальные объемы данных, проводить архивирование и восстановление БД. Несмотря на множество доступных функций, процесс управления СУБД не отличается высокой сложностью управления и использования. «MySQL» зарекомендовал себя как незаменимое средство для создания очень больших (централизованных или распределенных) баз данных для средних и крупных предприятий, где защита и сохранность данных – первоочередная задача [28].

В СУБД «Oracle» заложены такие же функции, что и в «MySQL», но, в отличие от последней, «Oracle» имеет высокую стоимость и сложные механизмы управления. Такую СУБД целесообразно применять в очень крупных предприятиях.

На рисунке 12 приводится экспертная оценка возможностей рассматриваемых СУБД: «MS Access», «Visual FoxPro», «MySQL» и «Oracle» (по десятибалльной шкале[28]).

Итоги анализа СУБД показывают, что для разработки надежной и безопасной информационной системы целесообразно использовать «MySQL», т.к. в отличие от «Oracle», она является бесплатной и проста в освоении и использовании.

Обоснование выбора средств разработки клиентской части ИС.

В последние годы большой популярностью у программистов пользуются среды быстрой разработки приложений (RAD). К семейству таких систем программирования относятся: «Visual Basic», «Delphi», «C++», «C++ Builder», «C#», и другие [31].

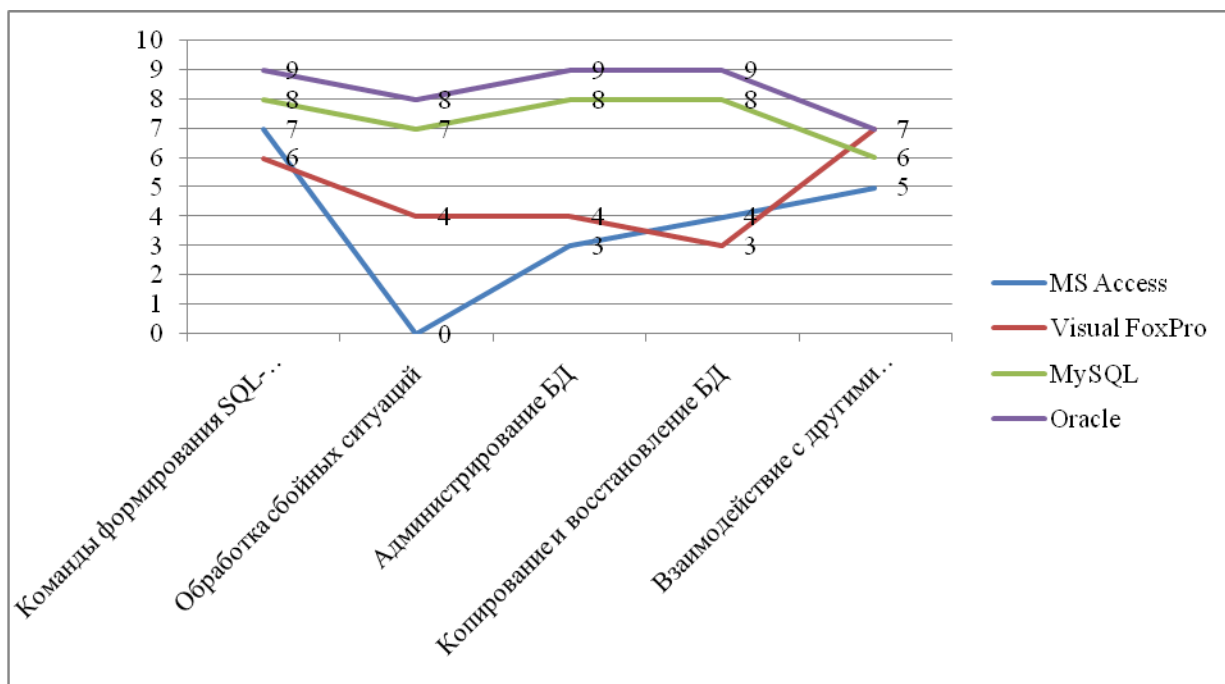


Рисунок 12 – Экспертная оценка основных СУБД

В нашей работе выбор среды разработки будет производиться на основе анализа перечисленных выше языков программирования.

«Visual Basic» хорошо подходит для быстрой разработки простых и недорогих информационных систем. Достоинством «Visual Basic» можно назвать простую и удобную среду разработки, позволяющую работать с различными типами баз данных. Программы, написанные на «Visual Basic», отличаются невысокой скоростью выполнения и требуют подключения дополнительных DLL-файлов [31].

«Delphi» является мощной и быстрой средой разработки приложений на основе технологии объектно-ориентированного программирования (ООП).

Среди преимуществ «Delphi» можно выделить:

- отсутствие необходимости в установке дополнительных библиотек;
- встроенный инструмент визуального моделирования, реализующий графическую архитектуру, которая позволяет управлять целыми бизнес-объектами;
- мощный генератор отчетов;
- быстрый доступ к различным базам данных;

– огромное количество встраиваемых компонентов.

Язык «Delphi» по возможностям значительно превосходит язык «Visual Basic» и даже язык «С++» (например, «Delphi» более надежен, стабилен и легок в изучении) [31].

Язык «С++» позволяет эффективно использовать практически все возможности современных вычислительных средств и операционных систем. Это достаточно мощный язык, позволяющий создавать кроссплатформенные приложения любого назначения. «С++» считается относительно сложным языком программирования в связи с необходимостью запоминания методов работы с каждым объектом. Еще одним важным недостатком данного языка является синтаксис, провоцирующий ошибки и плохая поддержка модульности (при подключении большого количества модулей, серьезно замедляется работа компилятора и заметно увеличивается размер конечного .exe файла) [31].

«С++ Builder» представляет собой некую интеграцию «Delphi» и «С++»: т.е. используется та же среда программирования (RAD), взятая из «С++» и библиотека визуальных компонентов (VCL), взятая из «Delphi». В этом и заключается основной недостаток этого языка - библиотека VCL, реализованная в «Delphi», существенно увеличивает размер исполняемых файлов. Преимущество «С++ Builder», по сравнению с предшественниками, - улучшенная версия объектной модели [31].

«С#» – язык, где сочетаются простота «Visual Basic» с мощностью «С++». Синтаксис языка аналогичен «С++», но более мощный, безопасный и простой. «С#» поддерживает работу с разными типами данных (а также позволяет создавать новые типы данных), статическую типизацию, перегрузку операторов и многое другое. Данный язык может применяться как при разработке приложений, так и для создания сайтов. В «С#», также как и у его оппонентов, есть свои недостатки, среди которых следует выделить моноплатформенность (разработка приложений исключительно для Windows) и низкое быстродействие [31].

На рисунке 13 приводится экспертная оценка возможностей рассматриваемых систем: «Visual Basic» (VB), «Delphi» (Del), «C++» , «C++ Builder» (CB) и C# (по десятибалльной шкале) [31].

Таким образом, подводя итоги анализа языков программирования, становится очевидным, что каждый из них может быть успешно использован для разработки, проектируемой в данной работе, информационной системы. Поэтому, выбор среды программирования будет производиться на основе опыта, уровня знаний и предпочтений разработчика, т.к. они способны существенным образом отразиться и на качестве реализуемого программного продукта.

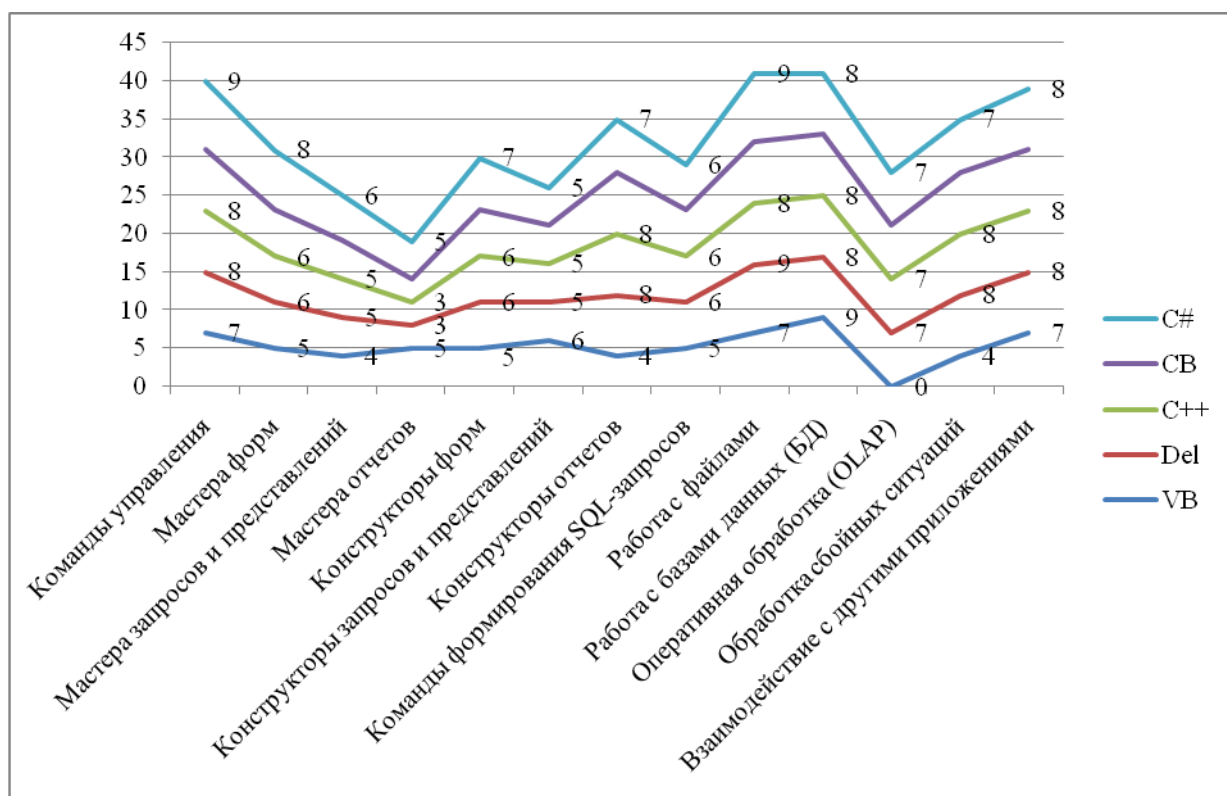


Рисунок 13 – Экспертная оценка основных средств разработки приложений

Для разработчика, наиболее понятным и удобным оказался язык программирования «Delphi». Наличие в интернете множества форумов по данной тематике, бесплатных обучающих ресурсов (книг, видео-уроков), обращение к которым поможет разобраться с возникшими в процессе программирования вопросами, только доказывают правильность принятого решения.

2.2 Информационное обеспечение

Информационная модель – это набор правил и операций, необходимых для выявления и изучения взаимосвязей между данными в конкретной предметной области.

Грамотно построенная информационная модель может служить основой для построения базы данных проектируемой ИС.

Чтобы построить информационную модель какого-либо бизнес-процесса, необходимо выявить перечень входных и выходных документов, участвующих в этом процессе. Так, для процесса «складской учет» в качестве входных документов служат справочники организации и иная информация, созданная пользователями проектируемой системы на основе этих справочников. Выходными документами системы являются: «Приходная накладная», «Упаковочный лист», «Квитанция об оплате» и различные отчеты.

На рисунке 14 представлено графическое отображение информационных потоков для изучаемой предметной области. Для построения модели использовался редактор диаграмм «DIA».

Рисунок показывает, что реквизиты любого из трех выходных документов («Приходная накладная», «Квитанция об оплате», «Упаковочный лист») имеют свои источники, которые служат входной информацией для разрабатываемой системы. Речь идет о таких справочниках как: «Клиенты», «Поставщики», «Сотрудники», «О компании», «Категории», «Товары», «Склады», «Ячейки хранения».

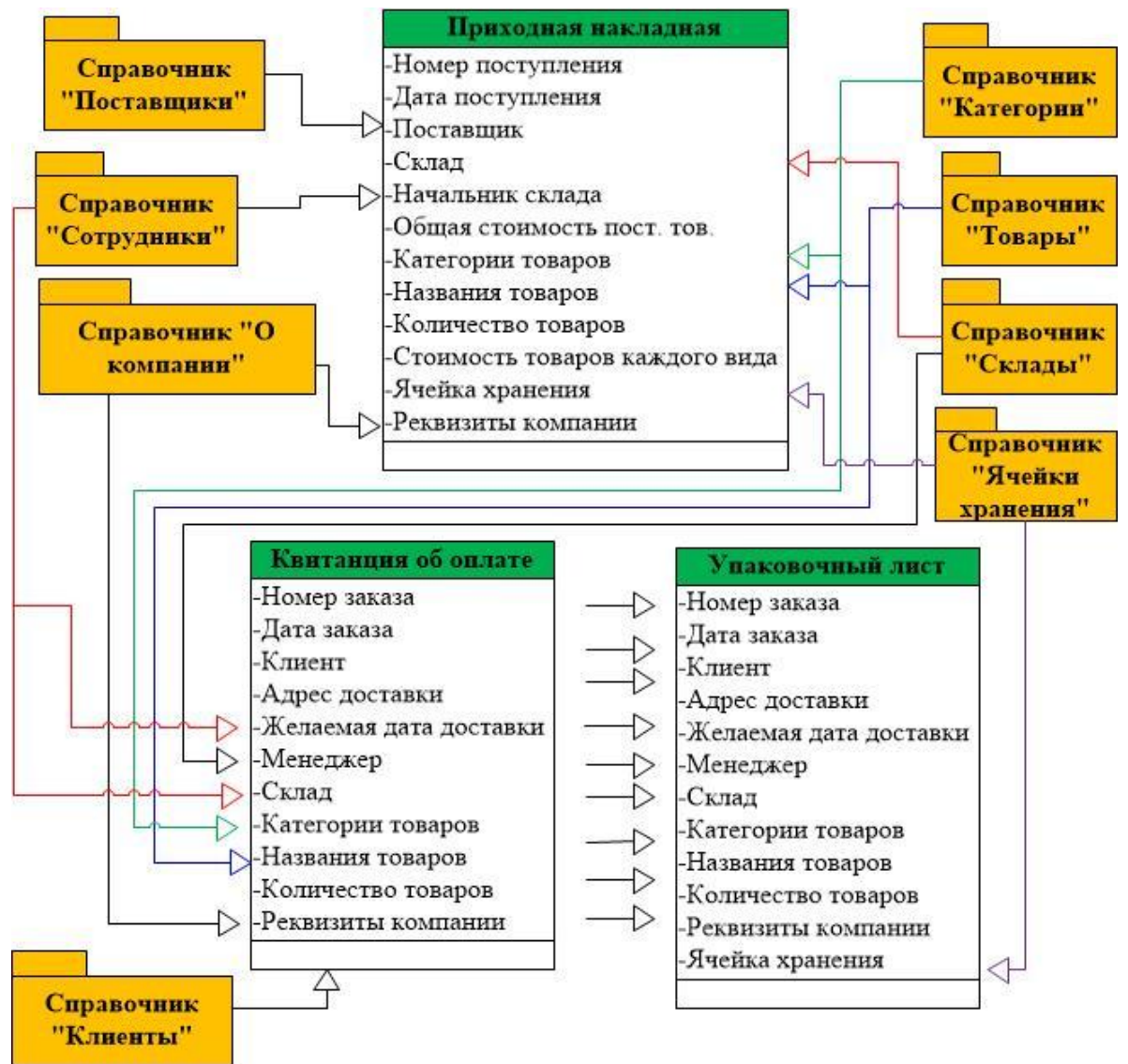


Рисунок 14 – Информационная модель проектируемой системы

Таким образом, информационная модель отображает тот минимум информации, который должен храниться в проектируемой информационной системе. Для определения полного перечня реквизитов документов, произведем описание входных и выходных источников данных.

Входная информация – это данные поступающие в систему извне. Для изучаемого бизнес-процесса такими источниками входных данных могут служить справочники системы: «Товары», «Категории», «Склады», «Ячейки хранения», «Сотрудники», «Поставщики», «Клиенты», «О компании».

На основе информации из перечисленных справочников осуществляются все процессы по учету заказов и складских операций. Регистрация таких опера-

ций производится в отдельных таблицах, также служащих в качестве входных источников информации: «Поступления», «Поступившие товары», «Списанные товары», «Заказы», «Заказанные товары», «Комплектование товаров» и «История перемещения».

Рассмотрим назначение каждого из перечисленных ранее справочников и таблиц.

Справочник «Товары» содержит ассортимент реализуемых компанией товаров, цены на них и название категории, к которым эти товары относятся.

Справочник «Категории» хранит перечень категорий товаров организации.

Справочник «Склады» содержит информацию о складах организации: их названия, адреса, имена начальников.

Справочник «Ячейки хранения» хранит сведения о ячейках склада: на сколько ячеек разбита территория склада, названия этих ячеек, количество рассчитанных мест в ячейках.

Справочник «Сотрудники» служит для хранения информации о работниках организации: ФИО, отдел, должность, адрес, телефон, E-mail и данных для входа в систему (логин и пароль).

Справочник «Поставщики» необходим для регистрации информации о поставщиках организации (названия организаций, контактная информация и банковские реквизиты).

Справочник «Клиенты» содержит ФИО, адреса и номера телефонов клиентов.

Справочник «О компании» должен содержать сведения о самой организации «Современные смазочные материалы»: название, ФИО директора, ФИО главного бухгалтера, юридический адрес, электронный адрес, телефон/факс, ИНН/КПП и иные банковские реквизиты организации.

Таблицы «Поступления» и «Поступившие товары» необходимы для регистрации операций по учету поступления товаров от поставщиков и хранения перечня поступивших товаров.

Таблица «Списанные товары» позволяет вести учета списаний товаров.

Таблицы «Заказы» и «Заказанные товары» необходимы для учета заказов, поступающих от клиентов и их состава.

Информация об укомплектованных товарах будет храниться в таблице «Комплектование товаров».

Для регистрации данных о перемещении товаров служит таблица «История перемещения».

Структура перечисленных таблиц приведена в главе «Физическая модель данных».

Результативная (выходная) информация – информация, формируемая на выходе системы.

Основными выходными документами проектируемой системы являются:

- «Приходная накладная»;
- «Квитанция об оплате»;
- «Упаковочный лист»;
- «Отчет по завершенным заказам»;
- «Отчет по реализованным товарам за определенный период»;
- «Отчет о поступлении товаров на склад за определенный период»;
- «Отчет об отсутствующих товарах на текущую дату»;
- «Отчет о просроченных товарах».

«Приходная накладная» – документ, содержащий сведения о товарах, поступивших на склад от поставщиков. В документе содержатся следующие сведения: дата поступления товаров, склад, название поставщика, общая сумма поступивших товаров, категории товаров, названия товаров, их количество и стоимость.

Документ «Квитанция об оплате» составляется менеджером и содержит: название склада, имя заказчика, адрес доставки заказа, желаемую дату доставки заказа, имя менеджера, перечень и количество заказанных товаров.

Документ «Упаковочный лист» составляется начальником склада для фасовщиков, чтобы те имели перед собой перечень комплектуемых товаров, требуемое количество товаров каждого вида и места их расположения на складе.

«Отчет по завершенным заказам» должен отражать перечень завершенных заказов и их стоимость, в общем, и за определенный период времени.

«Отчет по реализованным товарам за определенный период» должен отражать перечень проданных товаров и номер заказа, в рамках которого товар был реализован. Основываясь на показателях данного отчета, можно прогнозировать закупки на предстоящие периоды, в зависимости от того как меняется спрос на те или иные виды товаров.

«Отчет о поступивших от поставщиков товарах за определенный период» позволяет вести учет поступления товаров от каждого поставщика за определенный период.

«Отчет об отсутствующих товарах на текущую дату» отражает перечень отсутствующих товаров.

«Отчет о просроченных товарах» выдает список товаров, которые не были реализованы в рамках их срока годности. Товар может быть не продан вовремя из-за ошибок учета. Такие ошибки не самым лучшим образом отражаются на прибыли организации, в связи с чем данный отчет должен снизить риски подобного характера.

2.3 Описание реализации базы данных ИС

Логическая модель данных – графическое представление структуры базы данных. Она отображает перечень сущностей, их атрибутов, связей между ними и ограничений, налагаемые предметной областью [20].

Логическая модель данных составляется без привязки к определенной СУБД и является начальным прототипом будущей базы данных.

В нашем случае, БД будет реализована в среде «MySQL» с использованием инструментария «dbForge Studio», представленного мощными средствами

разработки и администрирования реляционных БД. Немаловажным достоинством «dbForge Studio» является возможность строить логические модели данных (схемы БД) на основе заранее созданных объектов.

Одним из подходов к построению модели базы данных является методология «IDEF 1X», основанная на концепции «сущность-связь».

Сущность на диаграмме обозначается в виде прямоугольника, разделенного на две части:

- ключевая область, представленная ключевыми атрибутами;
- область данных, состоящую из неключевых атрибутов.

Ключевыми считаются атрибуты, имеющие первичный ключ. Все остальные атрибуты относятся к неключевым (в том числе и атрибуты, составляющие внешний ключ).

Атрибуты – перечень имен, расположенных внутри блока сущности. Связи или отношения между атрибутами изображаются в виде линии, соединяющей родительский атрибут с дочерним. Отношения показывают, по какому из существующих типов связи («один-ко-многим», «один-к-одному» или «многие-ко-многим») соотносятся атрибуты между собой. В нашем случае, используется связь «один-ко-многим», в связи с чем, один из концов линии связи будет представлен точкой [17].

Логическая модель данных бизнес-процесса «Складской учет» на основе методологии «IDEF 1X» представлена на рисунке 15.

Модель базы данных, изображенная на рисунке 15, показывает, что проектируемая БД состоит из 16 таблиц, детальная информация о которых будет приведена при описании физической модели данных.

Физическая модель данных информационной системы представляет собой логическую модель данных, выраженную средствами конкретной СУБД. В качестве средства разработки БД будет использована реляционная система управления базами данных «MySQL».

При создании физической модели данных отношения, описанные в логической модели, преобразуются в таблицы, атрибуты становятся столбцами, каждому атрибуту присваивается тип данных, а для ключевых столбцов создаются уникальные индексы [20].

Как было отмечено ранее, проектируемая БД состоит из 16 таблиц (сущностей): «Товары», «Категории», «Поступления», «Поступившие товары», «Свойства товаров», «Склады», «Ячейки», «Поставщики», «Клиенты», «Сотрудники», «Заказы», «Заказанные товары», «Комплектование товаров», «История перемещения», «Списания» и «О компании».

Атрибуты сущности «Товары» представлены в таблице 3.

Атрибуты сущности «Категория» представлены в таблице 4.

Информация в таблицы «Поступления», «Поступившие товары» и «Свойства товаров» (поле «дата производства») заносится из «Приходной накладной».

Сущность «Поступления» необходима для регистрации сведений о поступлении товаров. В ней содержится следующая информация (таблица 5).

Сущность «Поступившие товары» представлена перечнем товаров, поступивших в рамках определенной приходной накладной. Атрибуты сущности «Поступившие товары» приведены в таблице 6.

Сущность «Свойства товаров» содержит данные о сроках хранения и адресе хранения товаров на складе. В таблице 7 приведены атрибуты сущности «Свойства товаров».

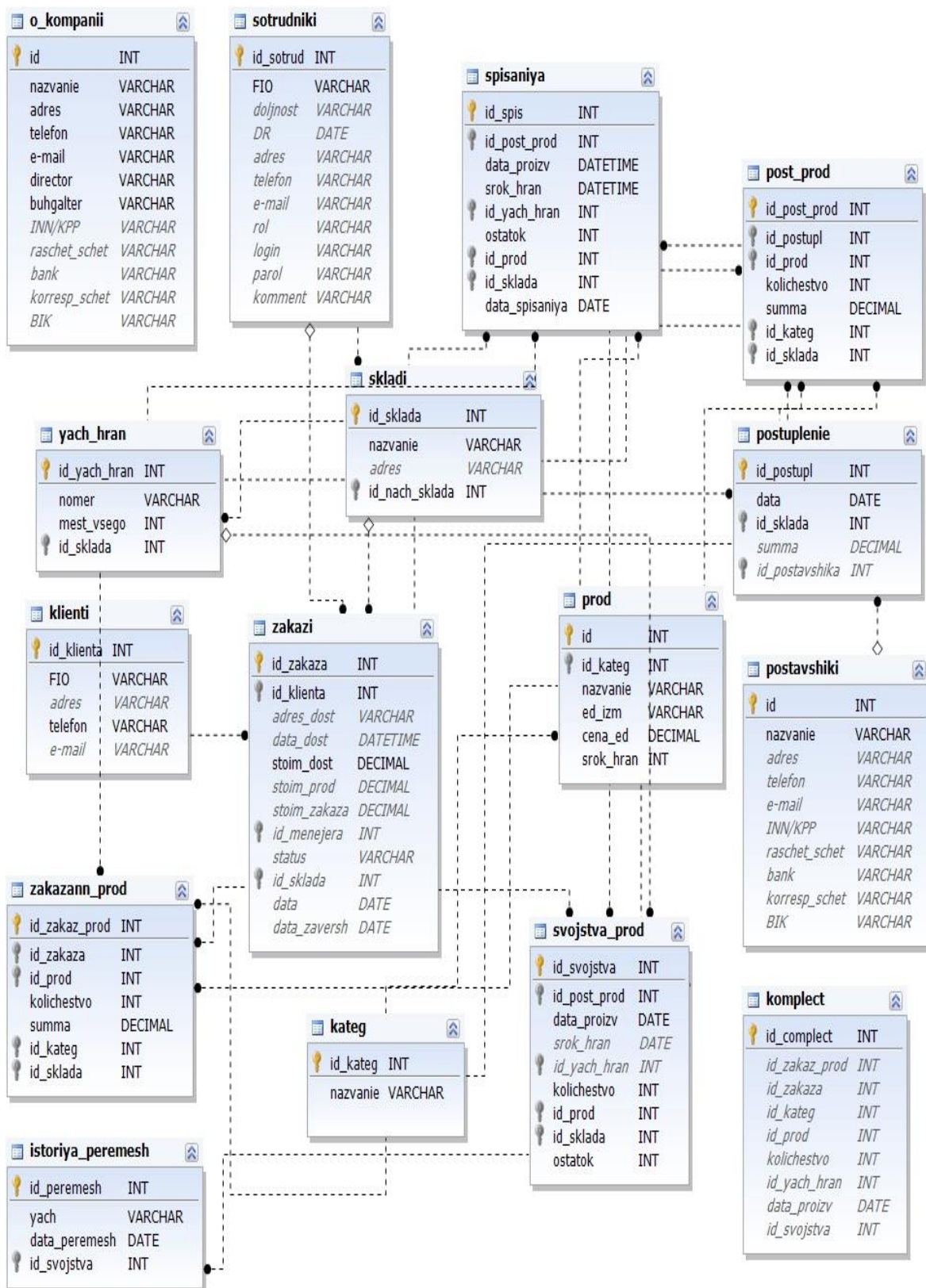


Рисунок 15 – Логическая модель данных

Таблица 3 – Атрибуты сущности «Товары»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id	int(10) unsigned	NO	PRI	auto_increm.	Код продукции
id_kateg	int(10) unsigned	NO	MUL	0	Код категории продукции
nazvanie	varchar(45)	NO			Название продукции
ed_izm	varchar(10)	NO			Единица измерения
cena_ed	decimal(8,2)	NO		0.00	Цена за единицу продукции
srok_hran	Integer	NO		0	Срок хранения (в днях)

Таблица 4 – Атрибуты сущности «Категория»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_kateg	int(10) unsigned	NO	PRI	auto_increm.	Код категории продукции
nazvanie	varchar(60)	NO			Название категории прод.

Таблица 5 – Атрибуты сущности «Поступления»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_postupl	int(10) unsigned	NO	PRI	auto_increm.	Код поступления
data	Date	NO		0000-00-00	Дата поступления
id_sklada	int(10) unsigned	NO	MUL	0	Код склада
summa	decimal(8,2)	NO		0.00	Сумма поступления
id_postavshika	int(10) unsigned	NO	MUL	0	Код поставщика

Таблица 6 – Атрибуты сущности «Поступившие товары»

	Тип	Ноль	Ключ	Знач. по ум.	Описание
--	------------	-------------	-------------	---------------------	-----------------

Имя					
id_post_prod	int(10) unsigned	NO	PRI	auto_increm.	Код поступившей прод.
id_postupl	int(10) unsigned	NO		0	Код поступления
id_prod	int(10) unsigned	NO	MUL	0	Код продукции
kolishestvo	int(10) unsigned	NO		0	Количество продукции
summa	decimal(8,2)	NO		0.00	Сумма
id_sklada	int(10) unsigned	NO	MUL	0	Код склада
id_kateg	int(10) unsigned	NO	MUL	0	Код категории прод.

Таблица 7 – Атрибуты сущности «Свойства товаров»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_svoystva	int(10) unsigned	NO	PRI	auto_increm.	Код свойства
id_post_prod	int(10) unsigned	NO	MUL	0	Код поступ. прод.
data_proizv	Date	NO		0000-00-00	Дата произв-ва прод.
srok_hran	Date	NO		0000-00-00	Срок хран. прод.
id_yach_hran	int(10) unsigned	NO	MUL	0	Код ячейки хран.
kolichество	Integer	NO		0	Кол-во продукции
id_sklada	int(10) unsigned	NO	MUL	0	Код склада
id_prod	int(10) unsigned	NO	MUL	0	Код продукции
ostatok	Integer	NO		0	Остаток продукции

Сущность «Склады» также служит справочником системы и хранит сведения о складах организации. Структура сущности приведена ниже.

Атрибуты сущности «Ячейки» приведены в таблице 9.

Сущность «Поставщики» хранит сведения о поставщиках организации (таблица 10).

Сущность «Клиенты» – соответственно, сведения клиентах (таблица 11).

Таблица 8 – Атрибуты сущности «Склады»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_sklada	int(10) unsigned	NO	PRI	auto_increm.	Код склада
nazvanie	varchar(45)	NO			Название склада
adres	varchar(45)	YES			Адрес склада
id_nach_sklada	int(10) unsigned	NO	MUL	0	Код начальн. склада

Таблица 9 – Атрибуты сущности «Ячейки»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_yach_hran	int(10) unsigned	NO	PRI	auto_increm.	Код ячейки хранения
nomer	varchar(5)	NO			Номер ячейки хранения
mest_vsego	integer	NO		0	Кол-во мест
id_sklada	int(10) unsigned	NO	MUL	0	Код склада

Таблица 10 – Атрибуты сущности «Поставщики»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id	int(10) unsigned	NO	PRI	auto_incremen	Код клиента

				t	
nazvanie	varchar(150)	NO			Название компании
adres	varchar(150)	NO			Адрес клиента
telefon	varchar(20)	NO			Телефон клиента
e-mail	varchar(20)	NO			Электронный адрес
director	varchar(45)	NO			ФИО директора
buhgalter	varchar(45)	NO			ФИО бухгалтера
INN/KPP	varchar(25)	YES			ИНН/КПП
raschet_schet	varchar(25)	YES			Расчетный счет
bank	varchar(45)	YES			Банк
kor-resp_schet	varchar(25)	YES			Корресп. счет
BIK	varchar(25)	YES			БИК

Таблица 11 – Атрибуты сущности «Клиенты»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_klienta	int(10) unsigned	NO	PRI	auto_increm.	Код клиента
FIO	varchar(45)	NO			ФИО конт. лица
adres	varchar(100)	YES			Адрес конт. лица
telefon	varchar(20)	NO			Телефон конт. лица
e-mail	varchar(20)	YES			Электронный адрес

Сущность «Сотрудники» содержит важные сведения о сотрудниках организации. Структура сущности представлена в таблице 12.

Таблица 12 – Атрибуты сущности «Сотрудники»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
-----	-----	------	------	--------------	----------

id_sotrud	int(10) unsigned	NO	PRI	auto_increment	Код сотрудника
FIO	varchar(45)	NO			ФИО
doljnost	varchar(100)	YES			Должность
DR	date	YES			Дата рождения
adres	varchar(100)	YES			Адрес
telefon	varchar(20)	YES			Телефон
e-mail	varchar(20)	YES			Электронный адрес
rol	varchar(20)	YES			Роль
login	varchar(15)	YES			Логин
parol	varchar(15)	YES			Пароль
komment	varchar(300)	YES			Комментарий

Сущность «Заказы» служит для регистрации заказов от клиентов. В ней хранятся общие сведения о заказе (его номер, дата, название клиента, сроки доставки и т.п.) Атрибуты сущности приведены ниже.

Таблица 13 – Атрибуты сущности «Заказы»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_zakaza	int(10) unsigned	NO	PRI	auto_increment	Код заказа
id_klienta	int(10) unsigned	NO	MUL	0	Код клиента

Продолжение таблицы 13

adres_dost	varchar(100)	YES			Адрес доставки
data_dost	datetime	YES			Дата доставки
stoim_dost	decimal(8,2)	YES			Стоимость доставки
stoim_prod	decimal(8,2)	YES			Стоимость прод.
stoim_zakaza	decimal(8,2)	YES			Стоимость заказа

id_menejera	int(10) unsigned	NO	MUL	0	Код менеджера
status	varchar(10)	NO		Новый	Статус заказа
data	datetime	NO			Дата заказа
data_zaversh	datetime	YES			Дата заверш. заказа
id_sklada	int(10) unsigned	NO	MUL	0	Код склада

Сущность «Заказанные товары» хранит перечень товаров, заказанных в рамках определенного заказа. В таблице 14 приведена структура данной сущности. Сущность «Комплектование товаров» хранит сведения о результатах упаковки товаров, представленных в заказе. В таблице 15 приведена структура данной сущности.

Таблица 14 – Атрибуты сущности «Заказанные товары»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_zakaz_prod	int(10) unsigned	NO	PRI	auto_increm.	Код заказан. прод.
id_zakaza	int(10) unsigned	NO	MUL	0	Код заказа
id_prod	int(10) unsigned	NO	MUL	0	Код продукции
kolichestvo	int(10) unsigned	NO		0	Количество
summa	decimal(8,2)	NO		0.00	Сумма
id_sklada	int(10) unsigned	NO	MUL	0	Код склада
id_kateg	int(10) unsigned	NO	MUL	0	Код категории

Таблица 15 – Атрибуты сущности «Комплектование товаров»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_complex	int(10) unsigned	NO		auto_increm.	Код комплектования
id_zakaz_prod	int(10) unsigned	NO		auto_increm.	Код заказан. продукции
id_zakaza	int(10) unsigned	NO		0	Код заказа
id_kateg	int(10) unsigned	NO		0	Код категории

id_prod	int(10) unsigned	NO		0	Код продукции
kolichество	int(10) unsigned	NO		0	Количество
summa	decimal(8,2)	NO		0.00	Сумма
id_sklada	int(10) unsigned	NO		0	Код склада
id_yach_hran	int(10) unsigned	NO		auto_increm.	Код ячейки хранения
data_proizv	date	NO		0000-00-00	Дата произв-ва прод.
id_svoystva	int(10) unsigned	NO		auto_increm.	Код свойства

Сущность «История перемещения» необходима для регистрации информации о перемещениях товаров внутри склада. Атрибуты сущности «История перемещения» представлены в таблице 16.

Таблица 16 – Атрибуты сущности «История перемещения»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_peremesh	int(10) unsigned	NO	PRI	auto_increm.	Код перемещения
yach	varchar(10)	NO		0	Ячейка хранения
data_peremesh	date	NO		0	Дата перемещения
id_svoystva	int(10) unsigned	NO	MUL	0	Код свойства

Сущность «Списания» необходима для списания товаров с истекшим сроком. Атрибуты сущности «Списания» представлены в таблице 17.

Описание обозначений базы данных представлено в приложении 1.

Таблица 17 – Атрибуты сущности «Списания»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id_spis	int(10) unsigned	NO	PRI	auto_increm.	Код списания

id_post_prod	int(10) unsigned	NO	MUL	0	Код постп. прод.
data_proizv	date	NO			Дата производства
srok_hran	date	NO			Срок хранения
id_yach_hran	int(10) unsigned	NO	MUL	0	Код ячейки
ostatok	Integer	NO			Остаток
id_prod	int(10) unsigned	NO	MUL	0	Код продукции
id_sklada	int(10) unsigned	NO	MUL	0	Код склада
data_spisaniya	Date	NO			Дата списания

Сущность «О компании» содержит важные сведения о самой организации «Современные смазочные материалы» и необходима для формирования исходящих документов. Структура сущности «О компании» представлена в таблице 18.

Таблица 18 – Атрибуты сущности «О компании»

Имя	Тип	Ноль	Ключ	Знач. по ум.	Описание
id	int(10) unsigned	NO	PRI	auto_increment	
nazvanie	varchar(150)	NO			Название компании
adres	varchar(150)	NO			Адрес
telefon	varchar(20)	NO			Телефон
e-mail	varchar(20)	NO			Электронный адрес

Продолжение таблицы 18

director	varchar(45)	NO			ФИО директора
buhgalter	varchar(45)	NO			ФИО бухгалтера
INN/KPP	varchar(25)	YES			ИНН/КПП
raschet_schet	varchar(25)	YES			Расчетный счет
bank	varchar(45)	YES			Банк
korresp_schet	varchar(25)	YES			Корресп. счет
BIK	varchar(25)	YES			БИК

2.4. Описание реализации клиентской части ИС

Для реализации проекта нами была выбрана среда объектно-ориентированного программирования «Delphi 7».

Delphi – это среда разработки программ, ориентированных на работу в операционных системах семейства Windows. Программы в Delphi создаются на основе современной технологии визуального проектирования [22].

Разработанная нами информационная система состоит из 41 формы, перечень которых представлен в приложении 2.

Основные функциональные возможности системы представлены на рисунке 16 в виде дерева функций.

При запуске программы «Складской учет» появляется заставка с названием программы (рисунок 17). Длительность отображения заставки 1 секунда, после чего открывается окно «Вход в систему» (рисунок 18).

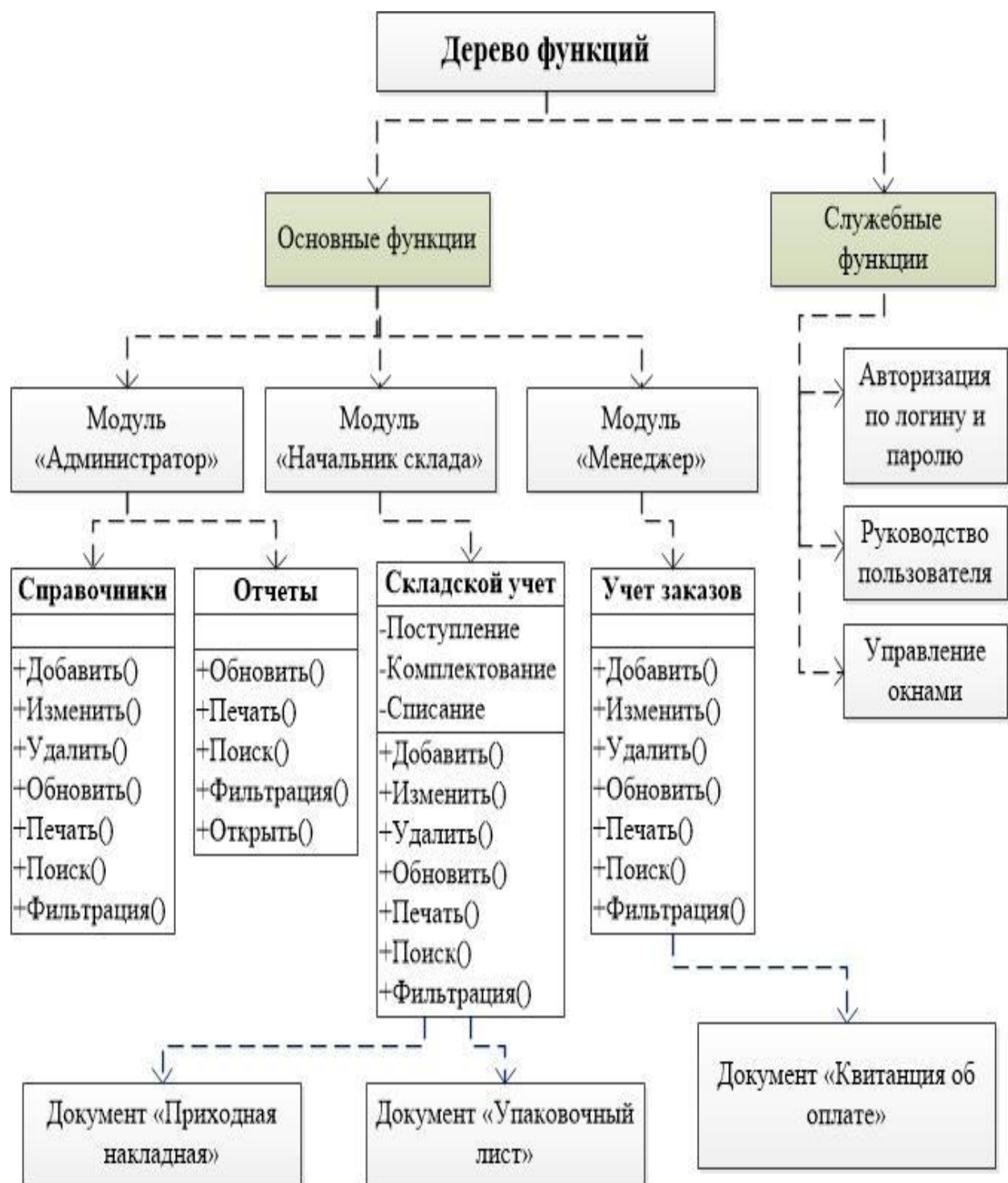


Рисунок 16 – Дерево функций

ИНФОРМАЦИОННАЯ
СИСТЕМА

Складской
учет



Рисунок 17 – Вид заставки

Рисунок 18 – Форма «Вход в систему»

Как показывает рисунок 18, вход в систему осуществляется путем ввода (заранее установленного администратором) уникального логина и пароля.

Если данные, введенные пользователем, не соответствуют установленным, то система выведет на экран сообщение: «Неверный логин или пароль. Попробуйте еще раз».

После успешной авторизации в системе, пользователь, в зависимости от своей должности, получит доступ к одному из трех модулей программы: «Администратор», «Начальник склада» или «Менеджер».

Рассмотрим особенности работы с каждым модулем.

Модуль «Администратор»

Главное меню модуля «Администратор» имеет следующую структуру (рисунок 19).

Раздел меню «Справочники» содержит перечень всех справочников системы. Доступ к добавлению, редактированию или удалению данных в справочниках имеют только пользователи с ролью администратора.

Все справочники имеют единый принцип работы:

- добавление, изменение и удаление записей происходит после нажатия на соответствующую кнопку, расположенную на панели инструментов;
- добавление и изменение записей производится в отдельном окне;
- двойной щелчок левой кнопкой мыши по какой-либо записи в таблице выводит пользователя в окно редактирования данных;
- содержимое справочников можно фильтровать и выводить на печать;
- для поиска записей необходимо вызвать окно поиска, нажав кнопку «Поиск» и ввести в строку поиска искомый текст.

Справочник «Категории» содержит перечень категорий, к которым относятся товары, реализуемые организацией (рисунок 20). Ввод данных о категории осуществляется на форме «Ввод названия категории» (рисунок 21).

Справочник «Товары» хранит сведения о перечне товаров, реализуемых организацией, их стоимости и сроках хранения. Для поиска необходимых товаров можно воспользоваться поиском и набрать в окне поиска часть имени искомого товара (рисунок 22).

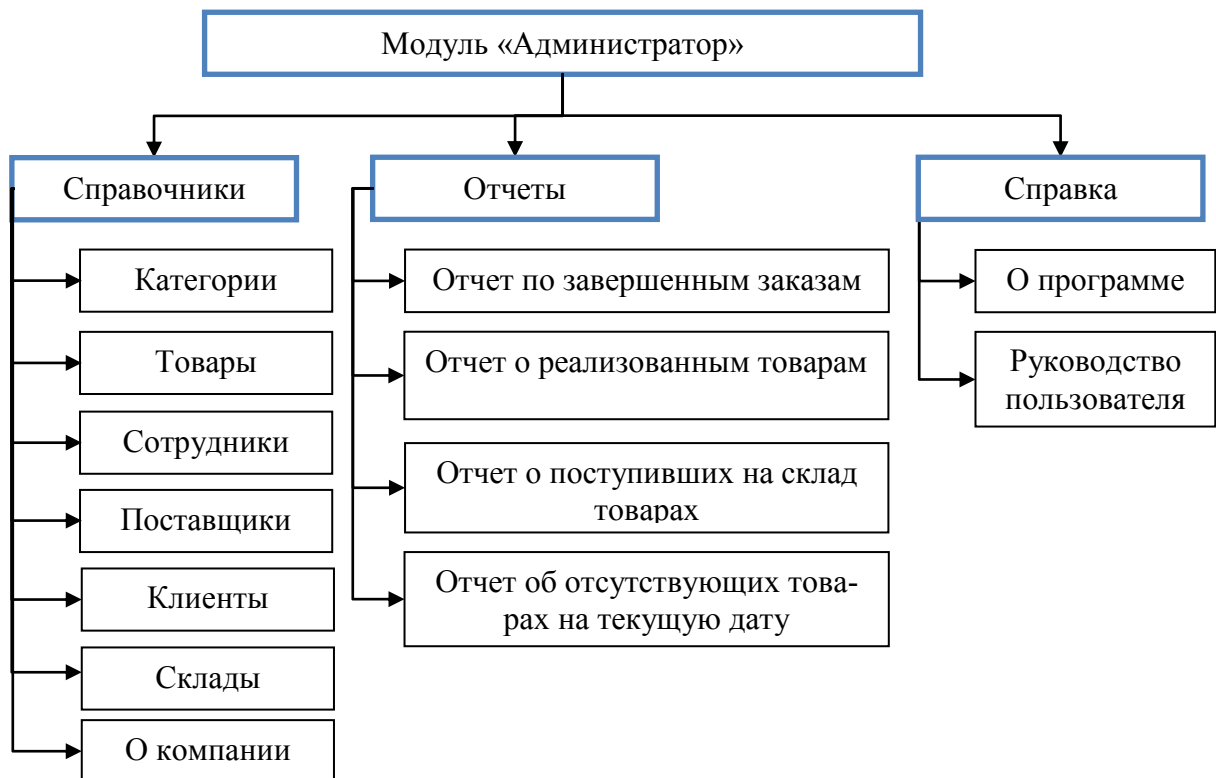


Рисунок 19 – Структура меню модуля «Администратор»

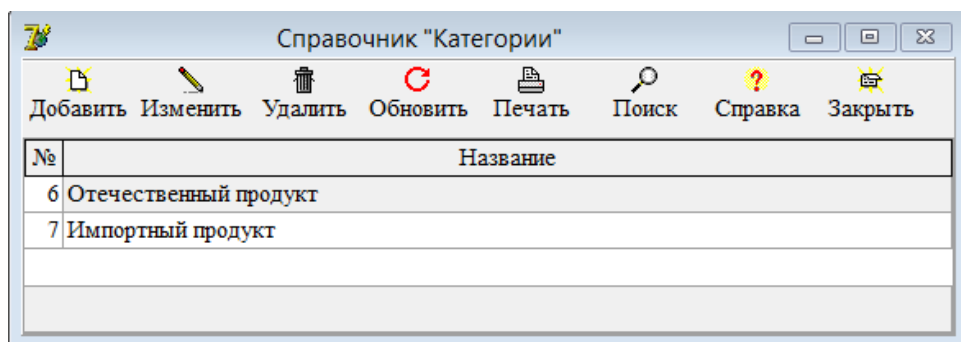


Рисунок 20 – Справочник «Категории»

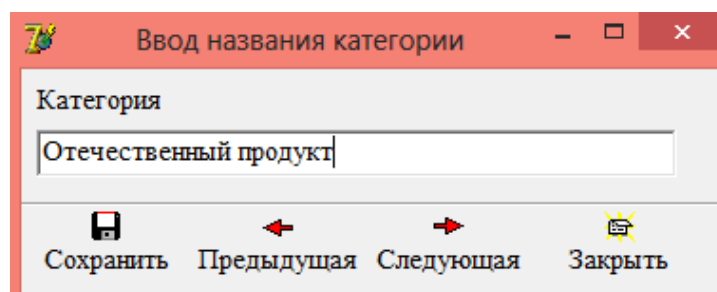


Рисунок 21 – Форма «Ввод названия категории»

Справочник "Товары"

Категория

Добавить Изменить Удалить Обновить Печать Поиск Справка Закрыть

№	Категория	Название	Ед. изм.	Цена за ед.	Срок хранения
1	Отечественный продукт	Gazpromneft ATF DX II (1л) трансм. для АКПП	шт.	172	365
2	Отечественный продукт	Gazpromneft Diesel Extra 15W40 CF-4/CF/SG (5	шт.	638	365
3	Отечественный продукт	Gazpromneft GL-4 75w90 (4л) п/с	шт.	938	365
4	Отечественный продукт	Gazpromneft GL-4 75w90 (20л) п/с	шт.	4452	365
5	Отечественный продукт	М/моторное M10Г2к (4,7л) г.Пермь	шт.	295	365
6	Импортный продукт	BP AUTRAN DX III 1л трансм. масло	шт.	366	365
7	Импортный продукт	BP Visco 2000 Diesel 15w40 1л	шт.	251	365
8	Импортный продукт	BP Visco 3000 Diesel 10w40 1л	шт.	270	365
9	Импортный продукт	Castrol EDGE 5w30 1л синтетика	шт.	2759	365
10	Импортный продукт	Castrol GTX 15w40 A3/B3 1л минеральное	шт.	296	365
11	Импортный продукт	Chevron Delo 400 MGX 15w40 (3,78л) (мин. диз)	шт.	754	365
12	Импортный продукт	ENEOS ATF Dexron III (20л)	шт.	6490	365
13	Импортный продукт	FANFARO OUTBOART 2Tact TC-W3 1л	шт.	309	365
14	Импортный продукт	G-Energy Многофункциональная смазка (210мл)	шт.	150	365
15	Импортный продукт	G-Energy S Synth 10W40 SL/CF (4л) п/с	шт.	850	365
16	Импортный продукт	Масло моторное Kixx D1 15w40 п/с 4л (дизель)	шт.	950	365
17	Отечественный продукт	Масло "Универсальное" (2,3л) г.Пермь	шт.	175	365

Рисунок 22 – Справочник «Товары»

Ввод данных о товаре осуществляется в окне, изображенном на рисунке 23.

Ввод данных о товаре

Категория: Отечественный продукт

Название: Gazpromneft ATF DX II (1л) трансм. для АКПП

Ед. изм.: шт.

Цена за ед.: 172 руб.

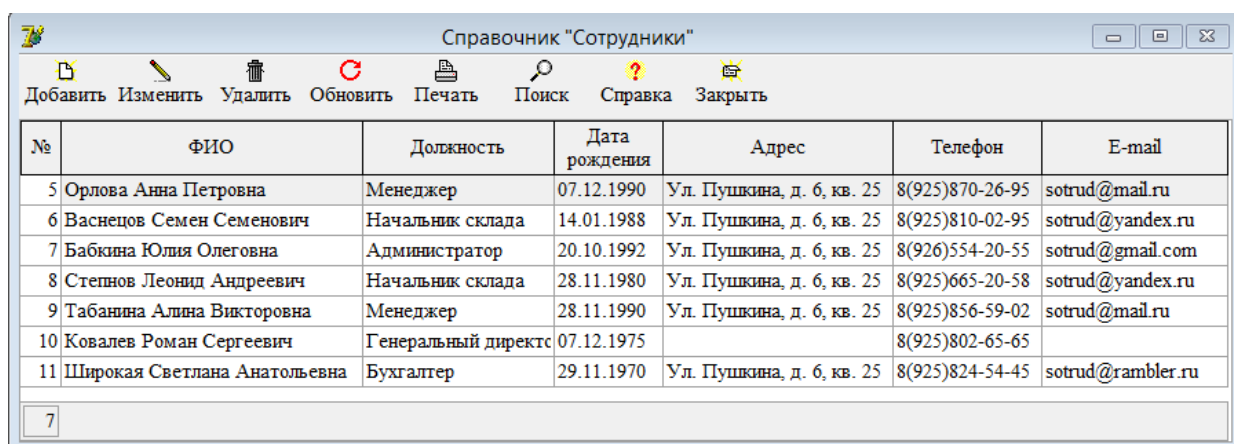
Срок хранения: 365 дней.

Сохранить ← → Закрыть

Рисунок 23 – Форма «Ввод данных о товаре»

Справочник «Сотрудники» позволяет вести учет сотрудников организации (рисунок 24). Для редактирования данных в справочнике достаточно про-

известить двойной щелчок по интересующей записи и внести изменения на форме «Ввод данных о сотруднике» (рисунок 25).



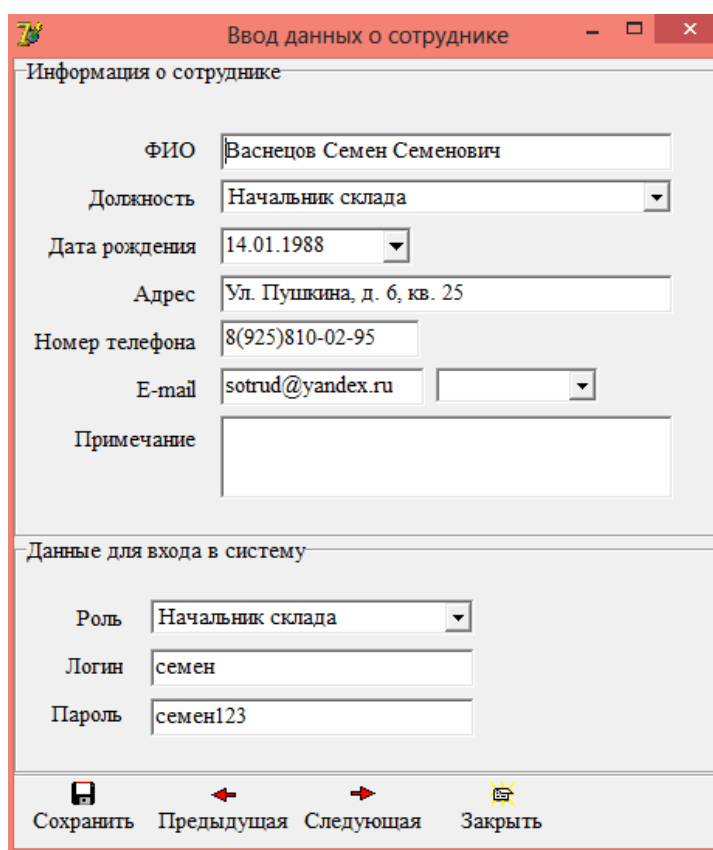
Справочник «Сотрудники»

Добавить Изменить Удалить Обновить Печать Поиск Справка Закрывать

№	ФИО	Должность	Дата рождения	Адрес	Телефон	E-mail
5	Орлова Анна Петровна	Менеджер	07.12.1990	Ул. Пушкина, д. 6, кв. 25	8(925)870-26-95	sotrud@mail.ru
6	Васнецов Семен Семенович	Начальник склада	14.01.1988	Ул. Пушкина, д. 6, кв. 25	8(925)810-02-95	sotrud@yandex.ru
7	Бабкина Юлия Олеговна	Администратор	20.10.1992	Ул. Пушкина, д. 6, кв. 25	8(926)554-20-55	sotrud@gmail.com
8	Степнов Леонид Андреевич	Начальник склада	28.11.1980	Ул. Пушкина, д. 6, кв. 25	8(925)665-20-58	sotrud@yandex.ru
9	Табанина Алина Викторовна	Менеджер	28.11.1990	Ул. Пушкина, д. 6, кв. 25	8(925)856-59-02	sotrud@mail.ru
10	Ковалев Роман Сергеевич	Генеральный директор	07.12.1975		8(925)802-65-65	
11	Широкая Светлана Анатольевна	Бухгалтер	29.11.1970	Ул. Пушкина, д. 6, кв. 25	8(925)824-54-45	sotrud@rambler.ru

7

Рисунок 24 – Справочник «Сотрудники»



Ввод данных о сотруднике

Информация о сотруднике

ФИО: Васнецов Семен Семенович

Должность: Начальник склада

Дата рождения: 14.01.1988

Адрес: Ул. Пушкина, д. 6, кв. 25

Номер телефона: 8(925)810-02-95

E-mail: sotrud@yandex.ru

Примечание:

Данные для входа в систему

Роль: Начальник склада

Логин: семен

Пароль: семен123

Сохранить Предыдущая Следующая Закрывать

Рисунок 25 – Форма «Ввод данных о сотруднике»

В данном окне требуется ввод:

- данных о сотруднике;
- данных для входа в систему.

Данные для входа в систему заполняются в том случае, если сотрудник занимает должность менеджера, начальника склада или администратора. Для остальных сотрудников эти поля заполнять не нужно.

Справочник «Поставщики» хранит информацию обо всех поставщиках организации. Вид данного справочника представлен на рисунке 26. Ввод данных о поставщике производится на одноименной форме (рисунок 27).

№	Название	Адрес	Телефон	E-mail	ИНН
6	Газпромнефть	Ул. Название 5	8(499)454-65-85	postavshik0@mail.ru	
7	ГОСТ Россия	Ул. Название 5	8(499)454-65-85	postavshik1@mail.ru	
8	Камский завод масел	Ул. Название 59	8(499)455-98-74	postavshik2@mail.ru	
9	Лукойл	Ул. Название 75	8(499)454-65-85	postavshik3@mail.ru	
10	Нефтеcинтез	Ул. Название 5	8(499)455-98-74	postavshik4@mail.ru	
11	BP	Ул. Название 35	8(499)455-98-74	postavshik5@mail.ru	
12	CASTROL	Ул. Название 92	8(499)455-98-74	postavshik6@mail.ru	
13	CHEVRON	Ул. Название 58	8(499)454-65-85	postavshik7@mail.ru	
14	ENEOS	Ул. Название 55	8(499)454-65-85	postavshik8@mail.ru	
15	ENI	Ул. Название 59	8(499)454-65-85	postavshik9@mail.ru	
16	FANFARO	Ул. Название 69	8(499)454-65-85	postavshik10@mail.ru	

Рисунок 26 – Справочник «Поставщики»

Справочник «Клиенты» хранит информацию обо всех клиентах организации. Вид данного справочника представлен на рисунке 28.

Форма ввода данных о клиенте имеет следующий вид (рисунок 29).

Справочник «Склады» хранит сведения о складах организации (рисунок 30). Для ввода информации о складе необходимо открыть форму «Ввод данных о складе», произведя двойной щелчок по интересующей записи (рисунок 31).

Как показывает рисунок 31, форма «Ввод данных о складе» тоже разбита на 2 части: склад и ячейки хранения. Ввод информации о ячейках производится на форме «Ввод данных о ячейке хранения» (рисунок 32).

Ввод данных о поставщике

Название: Газпромнефть

Адрес: Ул. Название 5

Телефон: 8(499)454-65-85

Е-mail: postavshik0@mail.ru

ИНН/КПП:

Расчетный счет:

Банк:

Корресп. счет:

БИК:

Сохранить Предыдущая Следующая Закрыть

Рисунок 27 – Форма «Ввод данных о поставщике»

Клиенты

Добавить Изменить Удалить Обновить Поиск Справка Закрыть

№	Название	Телефон	Е-mail	Адрес
1	ООО "Клиент 1"	8(925)654-54-51	klent1@gmail.com	Ул. Название, д. 1
2	ООО "Клиент 2"	8(925)684-23-36	klent2@gmail.com	Ул. Название, д. 2
3	ООО "Клиент 3"	8(925)874-12-35	klent3@gmail.com	Ул. Название, д. 3
4	ООО "Клиент 4"	8(925)445-46-54	klent4@gmail.com	Ул. Название, д. 4
5	ООО "Клиент 5"	8(926)710-99-20	klent5@gmail.com	Ул. Название, д. 5
6	ООО "Клиент 6"	8(925)698-12-16	klent6@gmail.com	Ул. Название, д. 6
7	ООО "Клиент 7"	8(925)698-12-16	klent7@gmail.com	Ул. Название, д. 7
8	ООО "Клиент 8"	8(925)698-12-16	klent8@gmail.com	Ул. Название, д. 8
9	ООО "Клиент 9"	8(925)698-12-16	klent9@gmail.com	Ул. Название, д. 9
10	ООО "Клиент 10"	8(925)698-12-16	klent10@gmail.com	Ул. Название, д. 10
11	ООО "Клиент 11"	8(925)698-12-16	klent11@gmail.com	Ул. Название, д. 11

Рисунок 28 – Справочник «Клиенты»

Ввод данных о клиенте

Название: ООО "Клиент 1"

Адрес: Ул. Название, д. 1

Номер телефона: 8(925)654-54-51

E-mail: klient1@gmail.com

Сохранить Предыдущая Следующая Закрыть

Рисунок 29 – Форма «Ввод данных о клиенте»

Справочник "Склады"

Добавить Изменить Удалить Обновить Справка Закрыть

№	Название	Адрес	Начальник склада
3	Основной	Ул. Ленина, д. 41	Васнецов Семен Семенович
4	Дополнительный	Ул. Строительная, д. 69	Степнов Леонид Андреевич

Рисунок 30 – Справочник «Склады»

Ввод данных о складе

Название: Основной

Начальник склада: Васнецов Семен Семенович

Адрес: Ул. Ленина, д. 41

Ячейки хранения

Добавить Изменить Удалить Обновить Справка

№	Номер	Мест всего
16	A1	100
17	A2	100
18	A3	100
19	A4	100
20	A5	100
36	B1	100
37	B2	100
38	B3	100
39	B4	100
40	B5	100

Сохранить Предыдущая Следующая Закрыть

Рисунок 31 – Форма «Ввод данных о складе»

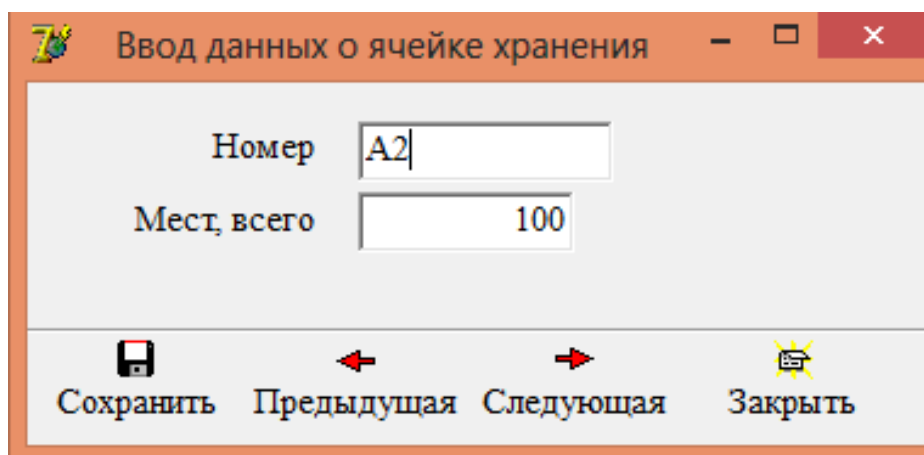


Рисунок 32 – Форма «Ввод данных о ячейке хранения»

Справочник «О компании» содержит общие сведения о самой организации. Информация из данного справочника используется при формировании выходных документов (рисунок 33).

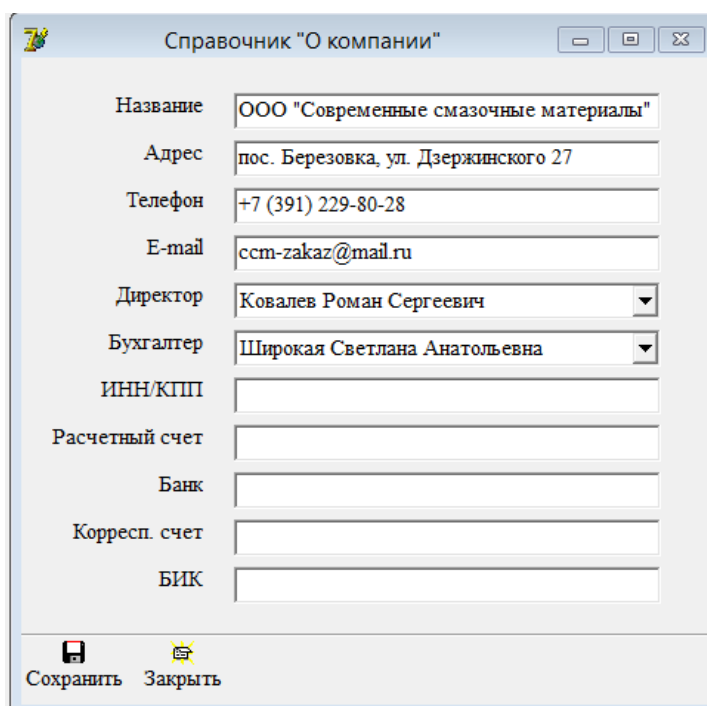


Рисунок 33 – Справочник «О компании»

Раздел меню «Отчеты» содержит 4 пункта, названия которых совпадают с именами формируемых в них отчетов. Работа с отчетами аналогична работе со

справочниками, но без доступа к операциям по добавлению, изменению или удалению данных.

Во всех отчетах системы можно проводить фильтрацию записей по интересующему пользователя временному интервалу. Для этого достаточно на верхней панели инструментов указать необходимый период и нажать кнопку «Отобразить».

Форма «Отчет по завершенным заказам» содержит информацию обо всех заказах, имеющих статус «Завершен» за определенный временной интервал (рисунок 34).

№	Дата	Клиент	Дата доставки	Стоимость доставки	Стоимость товаров	Стоимость заказа	Статус	Дата завершения	Склад	Менеджер
1	02.05.2018	ООО "Клиент 1"	07.05.2018	500	105719	106219	Завершен	05.05.2018	Основной	Орлова Анна Петровна
2	02.05.2018	ООО "Клиент 3"	07.05.2018	600	18064	18664	Завершен	05.05.2018	Дополнительный	Орлова Анна Петровна
3	03.05.2018	ООО "Клиент 6"	08.05.2018	500	15473	15973	Завершен	05.05.2018	Основной	Орлова Анна Петровна
4	03.05.2018	ООО "Клиент 7"	08.05.2018	500	58490	58990	Завершен	05.05.2018	Дополнительный	Орлова Анна Петровна
5	04.05.2018	ООО "Клиент 4"	09.05.2018	400	44290	44690	Завершен	05.05.2018	Основной	Орлова Анна Петровна
6	04.05.2018	ООО "Клиент 10"	08.05.2018	500	116033	116533	Завершен	05.05.2018	Основной	Орлова Анна Петровна

3000 358069 361069

Открыть Статистика Обновить Поиск Печать Справка Закрыть

Рисунок 34 – Форма «Отчет по завершенным заказам»

«Отчет по реализованным товарам» представлен перечнем товаров, реализованных за определенный период времени. Информация о реализованных товарах берется из заказов, имеющих статус «Завершен» (рисунок 35).

«Отчет о поступивших на склад товарах» отражает сведения о поступивших на склад товарах в рамках выбранного временного интервала (рисунок 36).

Раздел меню «Справка» состоит из пунктов «О программе» и «Руководство пользователя». Вид формы «О программе» представлен на рисунке 37.

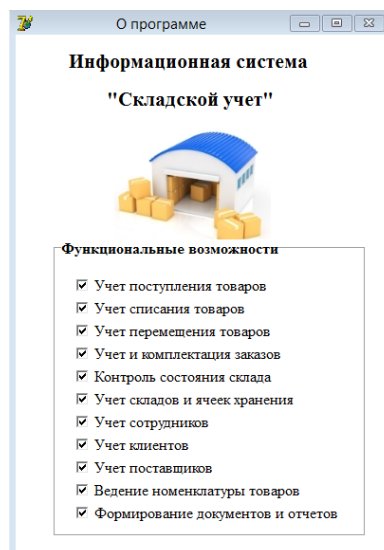


Рисунок 37 – Форма «О программе»

Модуль «Менеджер»

Структура меню модуля «Менеджер» имеет следующий вид (рисунок 38).

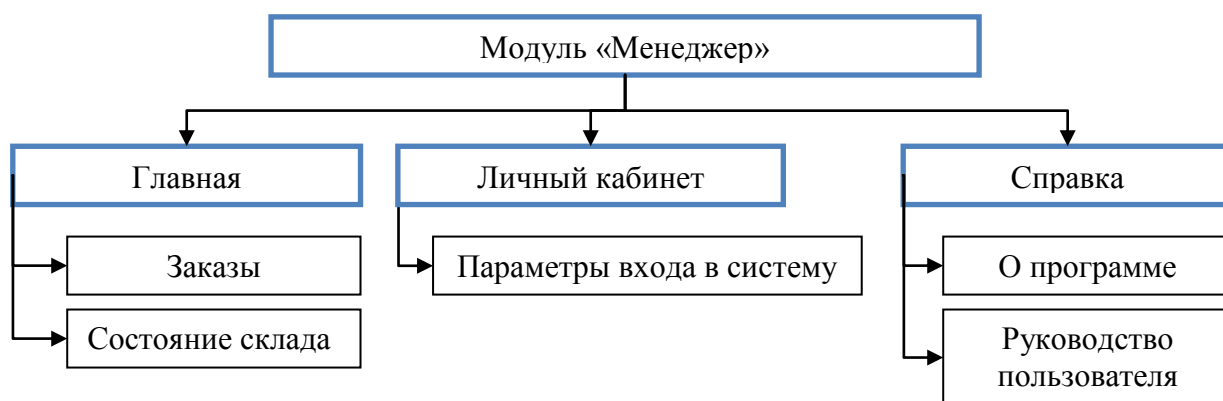


Рисунок 38 – Структура меню модуля «Менеджер»

Раздел меню «Главная» состоит из двух пунктов: «Заказы» и «Состояние склада».

Пункт меню «Заказы» позволяет вести регистрацию и учет всех заказов, поступающих от клиентов.

По умолчанию, при входе в систему, на экране отображаются заказы, имеющие статус «Новый». Для просмотра заказов с иным статусом, необходимо выбрать статус на верхней панели инструментов (рисунок 39).

Для облегчения поиска заказа, менеджер может воспользоваться поиском по названию клиента.

Чтобы изменить данные о заказе, можно нажать на кнопку «Изменить» на панели инструментов, либо произвести двойной щелчок левой кнопки мыши по интересующей записи.

№	Дата	Клиент	Дата доставки	Стоимость доставки	Стоимость товаров	Стоимость заказа	Статус	Склад	Дата завершения
1	02.05.2018	ООО "Клиент 1"	07.05.2018	500	105719	106219	Завершен	Основной	05.05.2018
2	02.05.2018	ООО "Клиент 3"	07.05.2018	600	18064	18664	Завершен	Дополнительный	05.05.2018
3	03.05.2018	ООО "Клиент 6"	08.05.2018	500	15473	15973	Завершен	Основной	05.05.2018
4	03.05.2018	ООО "Клиент 7"	08.05.2018	500	58490	58990	Завершен	Дополнительный	05.05.2018
5	04.05.2018	ООО "Клиент 4"	09.05.2018	400	44290	44690	Завершен	Основной	05.05.2018
6	04.05.2018	ООО "Клиент 10"	08.05.2018	500	116033	116533	Завершен	Основной	05.05.2018

3000 358069 361069

Рисунок 39 – Форма «Заказы»

Для удаления записи из таблицы, предусмотрена кнопка «Удалить», нажав на которую система потребует подтверждения совершаемого действия, выдав следующее окно (рисунок 40).

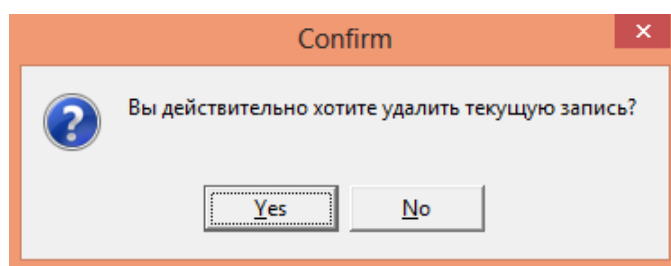


Рисунок 40 – Форма подтверждения удаления записи

Для регистрации нового заказа в системе, необходимо нажать кнопку «Добавить». Ввод информации о заказе осуществляется в окне «Ввод данных о заказе» (рисунок 41).

Ввод данных о заказе

Статус: Завершен

№: 1 от 02.05.2018

Клиент: ООО "Клиент 1"

Адрес доставки: Ул. Название, д. 1

Дата доставки: 07.05.2018

Склад: Основной

Стоимость доставки: 500 руб.

Стоимость продукции: 105719 руб.

Стоимость заказа: 106219 руб.

Дата завершения: 05.05.2018

Заказанные товары

№	Категория	Название	Кол-во	Сумма
1	Отечественный продукт	Gazpromneft ATF DX II (1л) транс. д	6	1032
2	Отечественный продукт	Gazpromneft GL-4 75w90 (20л) п/с	10	44520
3	Импортный продукт	BP Visco 2000 Diesel 15w40 1л	7	1757
4	Импортный продукт	ENEOS ATF Dexron III (20л)	9	58410
			32	105719

Сохранить Предыдущая Следующая Закрыть

Рисунок 41 – Форма «Ввод данных о заказе»

При создании нового заказа часть полей заполняется автоматически и не подлежит самостоятельному изменению:

- поле «Статус» принимает значение «Новый»;
- в поле «Дата» вписывается текущая дата;
- поле «№» заполняется автоматически в момент сохранения записи (при последующем просмотре записи, поле «№» уже будет иметь соответствующее значение).

Поля «Стоимость продукции», «Стоимость заказа» и «Дата завершения» тоже заполняются автоматически на основе дальнейших действий пользователя.

Все поля, значение в которых устанавливается системой, можно назвать неактивными, т.к. пользователь не имеет прямого доступа к ним. Значение таких полей обозначено серым цветом.

Регистрация сведений о заказе начинается с указания клиента, сделавшего заказ. Для этого необходимо выбрать название клиента из раскрывающегося

списка. Если клиент не был ранее зарегистрирован в системе, необходимо его добавить, выбрав кнопку «Добавить нового клиента» напротив поля «Клиент» (рисунок 42).

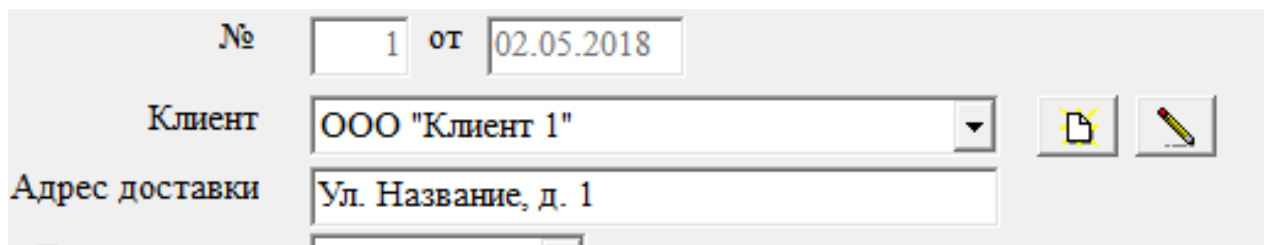


Рисунок 42 – Кнопка «Добавить нового клиента»

Поле «Адрес доставки» можно заполнить автоматически, щелкнув правой кнопкой мыши по данному полю и выбрав «Основной» (рисунок 43). Вслед за данным действием в поле «Адрес доставки» будет вписан адрес выбранного клиента, указанный в справочнике «Клиенты».

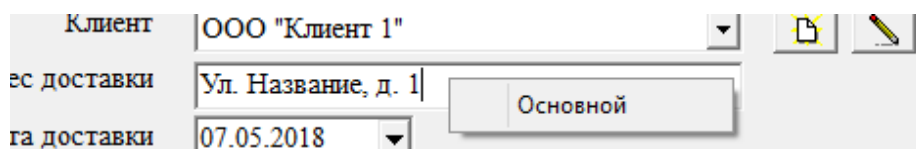


Рисунок 43 – Автозаполнение адреса доставки

Заполнив все необходимые поля, можно приступить к добавлению товаров в заказ. Для этого нажимаем кнопку «Добавить» и в окне «Ввод данных о заказанной продукции» указываем необходимые сведения (рисунок 44).

Рисунок 44 – Форма «Ввод данных о заказанных товарах»

На указанной форме от пользователя требуется указать категорию, к которой относится заказанный товар, а затем выбрать название заказанной продукции. В поле «Цена» автоматически пропишется цена на данный товар.

Как видно на рисунке 44, напротив поля «Количество» отобразилась подсказка, содержащая сведения об остатке выбранного товара на складе и его количестве, доступного для заказа. Данные значения могут отличаться друг от друга, т.к. некоторые товары могут быть заказаны и оплачены другими клиентами, но не отгружены со склада.

Мы видим, что выбранный товар доступен для заказа в количестве 20 штук, но это не значит, что клиент не сможет заказать больше 20 единиц товара данного вида. Можно указать любое количество товара, если клиент готов ждать его поступления на склад.

Указав требуемое клиенту количество товара, в поле «Сумма» пропишется стоимость, рассчитанная как произведение цены на количество выбранного товара.

Нажав кнопку сохранить в поле «Стоимость заказа» и «Стоимость продукции» на форме «Ввод данных о заказе» будет автоматически прописываться новое значение.

Для выдачи клиенту квитанции об оплате необходимо нажать кнопку печать. На экране появится вид данного документа (рисунок 45).

Квитанция об оплате

Номер заказа 1
 Дата заказа 02.05.2018
 Клиент ООО "Клиент 1"
 Адрес доставки Ул. Название, д. 1
 Дата доставки 07.05.2018
 Менеджер Орлова Анна Петровна

Категория	Продукция	Кол-во	Сумма
Отечественный продукт	Gazpromneft ATF DX II (1л) трансм. для А	6	1032
Отечественный продукт	Gazpromneft GL-4 75w90 (20л) п/с	10	44520
Импортный продукт	BP Visco 2000 Diesel 15w40 1л	7	1757
Импортный продукт	ENEOS ATF Dexron III (20л)	9	58410



Стоимость доставки 500 руб.
 Стоимость продукции 105719 руб.
 Стоимость заказа 106219 руб.

ООО "Современные смазочные материалы"
 пос. Березовка, ул. Дзержинского 27
 +7 (391) 229-80-28

Рисунок 45 – «Квитанция об оплате»

Для изменения статуса заказа, нужно щелкнуть правой кнопкой мыши по интересующему заказу и из появившегося списка выбрать нужный статус (рисунок 46).

.2018	500	105719	106219	Завершен	Основной	05.05.2018
.2018	600	18064	18664	Завершен	Дополнительный	05.05.2018
.2018	500	15473	15973	Завершен	Основной	05.05.2018
.2018	500	58490	58990			05.05.2018
.2018	400	44290	44690			05.05.2018
.2018	500	116033	116533	Завершен	Основной	05.05.2018

Рисунок 46 – Изменение статуса заказа

Перейдем в пункт меню «Состояние склада», где отражается информация о том, сколько продукции и какого вида находится на том или ином складе (рисунок 47). Выбрав определенный товар, в таблице справа отразятся сведения о данном товаре (дата производства, срок хранения, ячейка хранения и остаток).

Состояние склада

Склад: Основной

Обновить Печать Поиск Справка Состояние ячеек Закрыть

Остаток товаров на складе

№	Категория	Товар	Остаток
1	Отечественный продукт	Gazpromneft ATF DX II (1л) трансм. для АКПП	52
2	Отечественный продукт	Gazpromneft Diesel Extra 15W40 CF-4/CF/SG (5	50
3	Отечественный продукт	Gazpromneft GL-4 75w90 (4л) п/с	55
4	Отечественный продукт	Gazpromneft GL-4 75w90 (20л) п/с	42
5	Отечественный продукт	М/моторное М10Г2к (4,7л) г.Пермь	46
6	Импортный продукт	BP ATRAN DX III 1л трансм. масло	30
7	Импортный продукт	BP Visco 2000 Diesel 15w40 1л	13
8	Импортный продукт	BP Visco 3000 Diesel 10w40 1л	30
9	Импортный продукт	Castrol EDGE 5w30 1л синтетика	10
11	Импортный продукт	Chevron Delo 400 MGX 15w40 (3,78л) (мин.диз)	30
12	Импортный продукт	ENEOS ATF Dexron III (20л)	21
13	Импортный продукт	FANFARO OUTBOART 2Tact TC-W3 1л	48
17	Отечественный продукт	Масло "Универсальное" (2,3л) г.Пермь	11
18	Отечественный продукт	Масло индустриальное И-20 Россия (1л)	20
19	Отечественный продукт	Масло моторное М8В (4,7л) г.Пермь	4
			478

Свойства товаров

№	Дата производства	Срок хранения	Ячейка	Остаток
1	01.08.2017	01.08.2018	A1	2
32	01.05.2018	01.05.2019	A2	50

52

Рисунок 47 – Форма «Состояние склада»

Раздел меню «Личный кабинет» состоит из пункта «Параметры входа в систему» (рисунок 48). Здесь пользователь может изменить логин и пароль для входа в систему.

Параметры входа в си...

Логин: ваня

Пароль: ваня123

Сохранить Закрыть

Рисунок 48 – Форма «Параметры входа в систему»

Модуль «Начальник склада»

Модуль «Начальник склада» - основной модуль нашей программы. Структура меню данного модуля имеет следующий вид (рисунок 49).

Раздел меню «Главная» состоит из 4 пунктов:

– «Поступление товаров», где фиксируется информация обо всех поступлениях продукции от поставщиков на склад;

- «Заказы», где представлен перечень и состав заказов, сделанных клиентами;
- «Состояние склада», в котором отражено состояние складов организации;
- «Списания», где проводятся операции по списанию товаров с истекшим сроком годности.

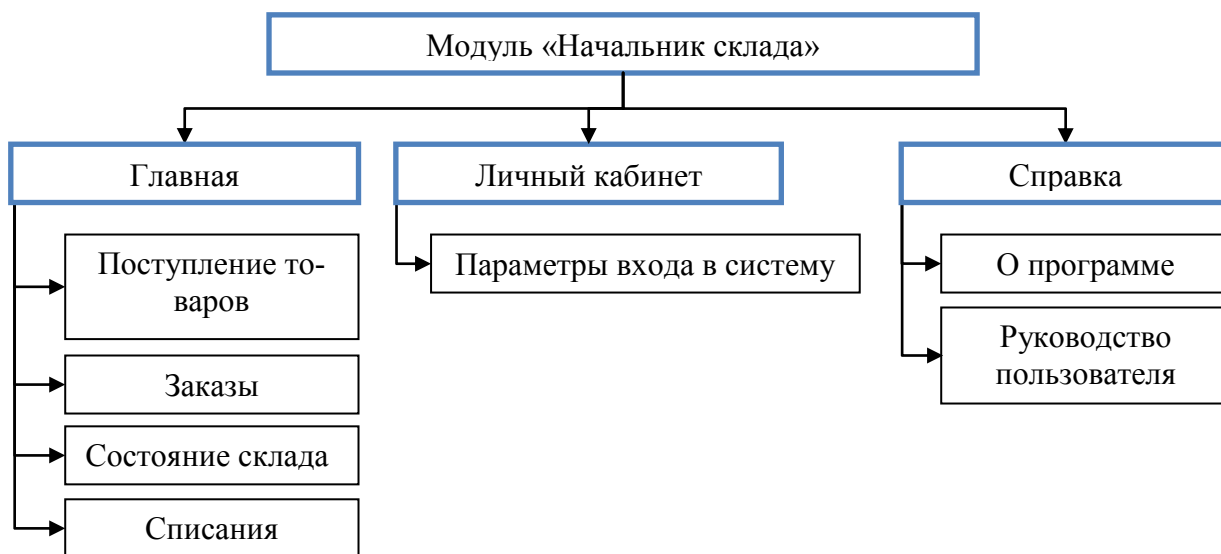


Рисунок 49 – Структура меню модуля «Менеджер»

На рисунке 50 представлен вид формы «Поступление товаров». Ввод данных о поступлении производится на форме «Ввод данных о поступлении товаров» (рисунок 51).

Поступление товаров				
№	Дата	Поставщик	Склад	Сумма
1	01.05.2018	Газпромнефть	Основной	79522
2	01.05.2018	ГОСТ Россия	Основной	14335
3	02.05.2018	Камский завод масел	Дополнительный	23215
4	02.05.2018	Лукойл	Дополнительный	329940
5	02.05.2018	BP	Основной	97240
6	03.05.2018	CASTROL	Дополнительный	61100
7	03.05.2018	CHEVRON	Основной	152420
8	03.05.2018	ENEOS	Основной	135980
9	04.05.2018	ENEOS	Дополнительный	266960
10	04.05.2018	Газпромнефть	Основной	324750
11	04.05.2018	BP	Дополнительный	35480
12	05.05.2018	Газпромнефть	Дополнительный	310000
				1830942

Рисунок 50 – Форма «Поступление товаров»

Как видно на рисунке 51, форма «Ввод данных о поступлении товаров» состоит из трех разделов:

- 1) – где требуется указать склад, на который поступила продукция и где отражена общая стоимость поступивших товаров;
- 2) – где представлен перечень поступивших товаров;
- 3) – где фиксируются свойства поступивших товаров.

Для вывода на печать «Приходной накладной» необходимо нажать кнопку «Печать». Вид документа «Приходная накладная» при предварительном просмотре выглядит следующим образом (рисунок 52).

Ввод данных о поступлении товаров

Склад: Основной

Поставщик: Газпромнефть

Сумма: 79522 руб.

Поступившие товары

Добавить Изменить Удалить Обновить Печать Справка

№	Категория	Товар	Кол-во	Сумма
1	Отечественный продукт	Gazpromneft ATF DX II (1л) транс. для АКПП	18	3096
2	Отечественный продукт	Gazpromneft Diesel Extra 15W40 CF-4/CF/SG (5	14	8932
3	Отечественный продукт	Gazpromneft GL-4 75w90 (4л) п/с	15	14070
4	Отечественный продукт	Gazpromneft GL-4 75w90 (20л) п/с	12	53424
				79522

Свойства поступивших товаров

Добавить Изменить Удалить Обновить Справка

№	Дата производства	Срок хранения	Ячейка	Кол-во поступило	Остаток
1	01.08.2017	01.08.2018	A1	18	2

18

Сохранить ← Следующая → Закрыть

Рисунок 51 – Форма «Ввод данных о поступлении товаров»

Для добавления данных о поступивших товарах, необходимо нажать кнопку добавить и в открывшемся окне «Ввод данных о поступивших товарах» (рисунок 53) указать необходимые сведения о товаре.

Добавив товар, нужно обязательно зафиксировать его свойства, нажав кнопку добавить в разделе «Свойства поступивших товаров». Ввод данных осуществляется в окне «Ввод данных о свойствах поступивших товаров» (рисунок 54).

При появлении данной формы, в поле «Количество» автоматически прописывается количество поступивших товаров. При указании «Даты производства», значение поля «Срок хранения» рассчитывается автоматически на основе срока хранения выбранного товара. При выборе ячейки хранения, в нижней части формы отображается информация о количестве свободных в данной ячейке мест.

Приходная накладная

Код 1
 Дата 01.05.2018
 Склад Основной
 Сумма 79522 руб.
 Поставщик Газпромнефть

Категория	Товар	Кол-во	Сумма
Отечественный продукт	Gazpromneft ATF DX II (1л) трансм. для АКПП	18	3096
Отечественный продукт	Gazpromneft Diesel Extra 15W40 CF-4/CF/SG (5	14	8932
Отечественный продукт	Gazpromneft GL-4 75w90 (4л) п/с	15	14070
Отечественный продукт	Gazpromneft GL-4 75w90 (20л) п/с	12	53424

ИТОГО, руб. 79522

Рисунок 52 – Вид документа «Приходная накладная»

Рисунок 53 – Форма «Ввод данных о поступивших товарах»

Рисунок 54 – «Ввод данных о свойствах поступивших товаров»

В правой части данной формы фиксируется история перемещения товара.

В случае если поступившие товары имеют разную дату производства, то необходимо для каждой партии товара создать отдельное свойство.

К тому же товары могут быть размещены в разных ячейках хранения. В таком случае тоже требуется создать отдельное свойство для товаров, размещенных в различных ячейках.

№	Дата производства	Срок хранения	Ячейка	Кол-во поступило	Остаток
4	01.12.2017	01.12.2018	A1	6	0
5	01.12.2017	01.12.2018	A2	6	0

Рисунок 55 – Свойства поступивших товаров

Раздел меню «Заказы» содержит перечень заказов, которые менеджер зарегистрировал в системе через свой интерфейс (рисунок 56).

№	Дата	Клиент	Дата доставки	Стоимость заказа	Статус	Склад
1	02.05.2018	ООО "Клиент 1"	07.05.2018	106219	Завершен	Основной
2	02.05.2018	ООО "Клиент 3"	07.05.2018	18664	Завершен	Дополнительный
3	03.05.2018	ООО "Клиент 6"	08.05.2018	15973	Завершен	Основной
4	03.05.2018	ООО "Клиент 7"	08.05.2018	58990	Завершен	Дополнительный
5	04.05.2018	ООО "Клиент 4"	09.05.2018	44690	Завершен	Основной
6	04.05.2018	ООО "Клиент 10"	08.05.2018	116533	Завершен	Основной

Рисунок 56 – Форма «Заказы»

В модуле «Начальник склада» над заказами проводится процедура комплектации заказа. Для этого необходимо открыть заказ для просмотра (рисунок 57).

Статус: В работе
 №: 7 от 05.05.2018
 Клиент: ООО "Клиент 4"
 Адрес доставки: Ул. Название, д. 4
 Дата доставки: 08.05.2018
 Склад: Основной
 Стоимость доставки: 600 руб.
 Стоимость продукции: 63525 руб.
 Стоимость заказа: 64125 руб.
 Дата завершения:

Заказанные товары

Укомплектовать Обновить Справка

№	Категория	Товар	Кол-во	Сумма
26	Импортный продукт	BP Visco 2000 Diesel 15w40 1л	10	2510
27	Импортный продукт	Castrol EDGE 5w30 1л синтетика	5	13795
28	Импортный продукт	BP Visco 3000 Diesel 10w40 1л	10	2700
29	Отечественный продукт	Gazpromneft GL-4 75w90 (20л) п/с	10	44520
			35	63525

Документ "Упаковочный лист"

№	Категория	Товар	Кол-во	Ячейка	Дата производства
<No data to display>					

Всего:

← Предыдущая → Следующая ✖ Закрыть

🖨 Печать 🗑 Удалить

Рисунок 57 – Форма «Просмотр и комплектование заказа» до проведения комплектования заказа

Чтобы начать комплектовать заказ, нужно выбрать заказанную продукцию и нажать кнопку «Укомплектовать». Откроется форма, где представлены сведения о свойствах выбранного товара, упорядоченные по столбцу «Срок хранения», чтобы начальник склада мог видеть товары с истекающим сроком годности (рисунок 58).

№	Название	Дата производства	Срок хранения	Ячейка	Остаток
34	Gazpromneft GL-4 75w90 (20л) п/с	01.05.2018	01.05.2019	A4	42
45	Gazpromneft GL-4 75w90 (20л) п/с	03.05.2018	03.05.2019	A3	30

Собрать заказ Отмена

Рисунок 58 – Форма «Комплектование заказанных товаров»

Выбрав необходимое свойство товара, нужно нажать кнопку «Собрать заказ» и в открывшемся окне указать количество требуемого товара (рисунок 59).

Когда все товары укомплектованы, на форме «Просмотр заказа» таблица справа, должна принять следующий вид (рисунок 60).

Количество

Укажите количество

ОК

Рисунок 59 – Форма «Количество»

Просмотр и комплектование заказа

Статус: В работе
 № 7 от 05.05.2018
 Клиент: ООО "Клиент 4"
 Адрес доставки: Ул. Название, д. 4
 Дата доставки: 08.05.2018
 Склад: Основной
 Стоимость доставки: 600 руб.
 Стоимость продукции: 63525 руб.
 Стоимость заказа: 64125 руб.
 Дата завершения:

Документ "Упаковочный лист"

№	Категория	Товар	Кол-во	Ячейка	Дата производства
Товар: Газпромнефт GL-4 75w90 (20л) п/с (, 10)					
41	Отечественный продукт	Газпромнефт GL-4 75w90 (20л)	10	A4	01.05.2018
			Количество	10	
Товар: BP Visco 2000 Diesel 15w40 1л(, 10)					
38	Импортный продукт	BP Visco 2000 Diesel 15w40 1л	10	B1	06.02.2018
			Количество	10	
Товар: BP Visco 3000 Diesel 10w40 1л(, 10)					
40	Импортный продукт	BP Visco 3000 Diesel 10w40 1л	10	B1	02.03.2018
			Количество	10	
Товар: Castrol EDGE 5w30 1л синтетика (, 5)					
39	Импортный продукт	Castrol EDGE 5w30 1л синтетика	5	A1	20.04.2018
			Количество	5	
			Всего	35	

Укомплектовать Обновить Справка

№	Категория	Товар	Кол-во	Сумма
26	Импортный продукт	BP Visco 2000 Diesel 15w40 1л	10	2510
27	Импортный продукт	Castrol EDGE 5w30 1л синтетика	5	13795
28	Импортный продукт	BP Visco 3000 Diesel 10w40 1л	10	2700
29	Отечественный продукт	Газпромнефт GL-4 75w90 (20л) п/с	10	44520
			35	63525

← Предыдущая → Следующая 🗑 Закрыть

🖨 Печать 🗑 Удалить

Рисунок 60 – Форма «Просмотр и комплектование заказа» после проведения комплектования заказа

На основе данной таблицы формируется документ «Упаковочный лист», который можно вывести на печать, нажав на кнопку «Печать». Вид отчета при предварительном просмотре будет выглядеть следующим образом (рисунок 61).

Упаковочный лист

№ заказа 7
 Дата заказа 05.05.2018
 Клиент ООО "Клиент 4"
 Адрес доставки Ул. Название, д. 4
 Дата доставки 08.05.2018

Категория	Товар	Кол-во	Ячейка	Дата произв.
Импортный продукт	BP Visco 2000 Diesel 15w40 1л	10	B1	06.02.2018
Импортный продукт	Castrol EDGE 5w30 1л синтетика	5	A1	20.04.2018
Импортный продукт	BP Visco 3000 Diesel 10w40 1л	10	B1	02.03.2018
Отечественный продукт	Газпромнефт GL-4 75w90 (20л) п/с	10	A4	01.05.2018
Итого		35		

Начальник склада
 Васнецов Семен Семенович

Подпись




Рисунок 61 – «Упаковочный лист»

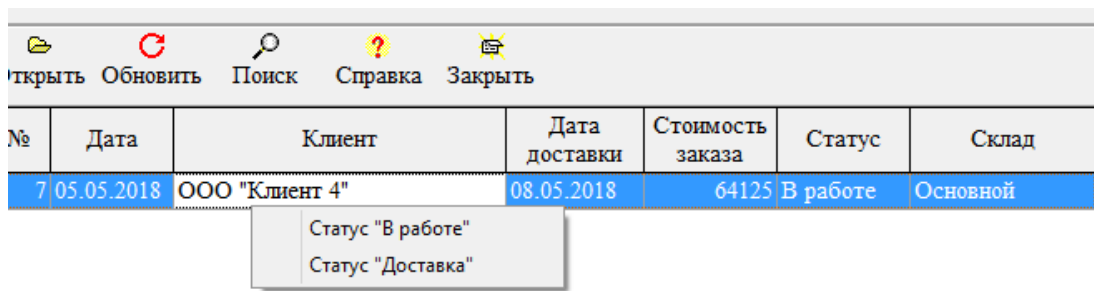


Рисунок 62 – Изменение статуса заказа

Пункт меню «Состояние склада» отражает текущее состояние складов организации (рисунок 63).

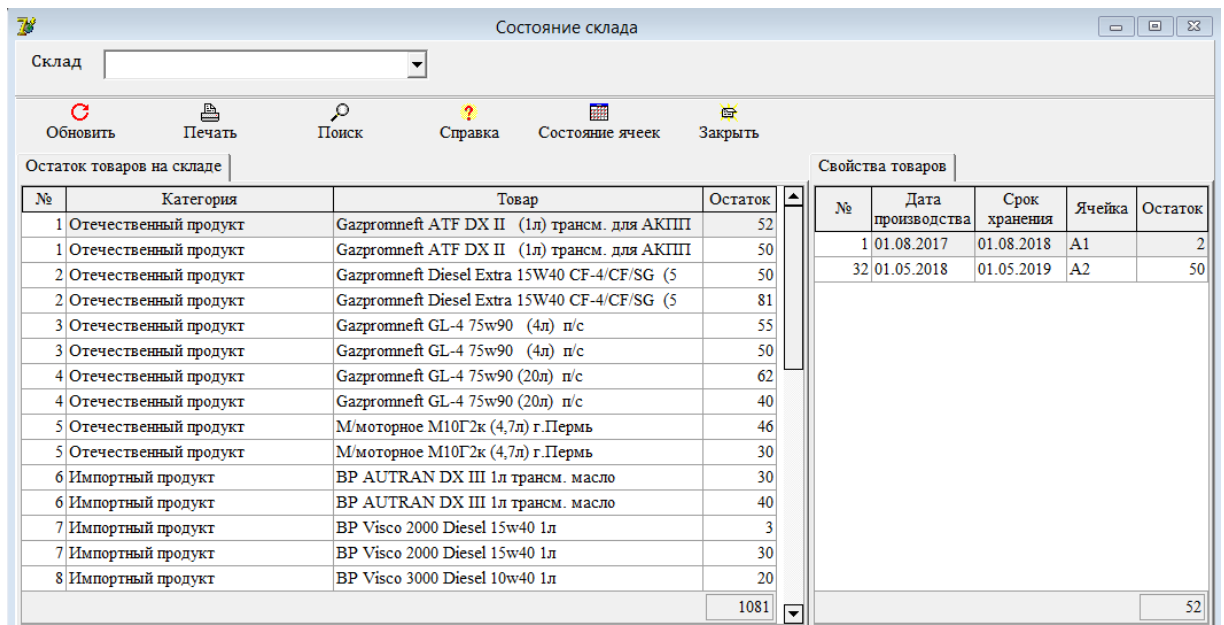


Рисунок 63 – Форма «Состояние склада»

Чтобы получить информацию о состоянии ячеек на складе, необходимо нажать на одноименную кнопку «Состояние ячеек», в след за чем на экран будет выведена форма «Состояние ячеек» (рисунок 64). Данная форма отражает, сколько мест в ячейке хранения занято или свободно в настоящий момент времени.

Ячейка	Мест всего	Занято	Свободно
A1	100	82	18
A2	100	80	20
A3	100	81	19
A4	100	43	57
A5	100	68	32
B1	100	27	73
B2	100	92	8

Рисунок 64 – Форма «Состояние ячеек»

Пункт меню «Списания» содержит информацию о товарах с истекшим сроком годности. Данные товары необходимо списать, нажав кнопку «Списать» (рисунок 65).

№	Название	Срок хранения	Ячейка	Склад	Остаток
1	Gazpromneft ATF DX II (1л) транс. для АКПП	01.05.2018	A1	Основной	2

Рисунок 65 – Форма «Отчет о товарах с истекшим сроком годности»

Для получения перечня списанных товаров, нужно нажать кнопку «Списания» и в открывшемся окне «Списанные товары» просмотреть сведения о списанной продукции (рисунок 66).

Списанные товары							
№	Дата списания	Название	Дата производства	Срок хранения	Склад	Ячейка	Списано
1	05.05.2018	Gazpromneft ATF DX II (1л) трансм. для АКПП	01.08.2017	01.05.2018	Основной	A1	2

Рисунок 66 – Форма «Списанные товары»

3 Обоснование экономической эффективности проекта

3.1 Определение экономической эффективности проекта

Внедрение любой информационной системы должно сопровождаться повышением экономической эффективности деятельности организации. Обеспечить это возможно благодаря созданию возможности решения текущих задач организации с использованием новых, современных способов получения, обработки, анализа и хранения информации.

Под экономической эффективностью проекта понимается оценка результатов функционирования системы, или, иначе говоря, происходит оценка результатов от внедрения системы с затратами, выделенными на ее разработку.

Положительного эффекта от внедрения информационной системы можно добиться только в случае, если будут достигнуты следующие цели, ради решения которых изначально проектировалась система. Главной целью любой автоматизированной информационной системы является оптимизация бизнес-процессов предприятия и улучшение показателей ее деятельности.

Достижение главной цели обеспечивается путем достижения подцелей, таких как: снижение затрат на обработку информации; повышение актуальности, достоверности и точности информации; сокращение времени получения результатной информации; создание механизмов для проведения анализа данных; получение новой информации, получить которую без применения вычислительных средств было невозможно, либо очень трудоемко.

Такой информацией может служить:

- информация о местоположении каждой единицы товара на складе;
- информация о свободных ячейках на складе;
- информация о просроченных товарах;
- информация о наличии товаров на складе и т. п.

Очевидно, что информация подобного характера способна оказать положительное влияние на результат функционирования организации и на рост экономического эффекта от ее деятельности.

Такой вид экономической эффективности принято иначе называть «косвенной эффективностью», характеризующей качественные изменения на объекте автоматизации, происходящие в результате внедрения информационной системы.

Таковыми качественными изменениями могут быть: увеличение продаж; привлечение новых клиентов; увеличение скорости создания и обработки заказов; оперативный обмен информацией между различными подразделениями организации; снижение расходов на оплату труда за счет сокращения числа сотрудников, занятых обработкой информации.

В любом случае, внедрение информационной системы и обработка информации на ЭВМ может стать источником повышения эффективности работы организации, чем и обосновывается целесообразность ее использования.

Показателями экономической эффективности проектируемой системы могут быть снижение трудовых и стоимостных затрат, высвобождение людей из сферы обработки информации и др. Значения показателей данного вида эффективности рассчитываются исходя из объемов перерабатываемой информации и производительности технических средств. Данный вид экономической эффективности называют прямой эффективностью.

3.2 Определение показателей прямой экономической эффективности

Показатели прямой экономической эффективности информационных систем определяются в результате сравнения затрат на обработку информации в изначальном (базовом, до внедрения ИС) периоде и предлагаемом (после внедрения ИС) периоде.

Примером базового варианта выступает период до внедрения разработанной системы, а предлагаемый вариант – это промежуток времени с момента ее

внедрения. При сравнении необходимо оценить какие плюсы или минусы получила организация от использования новой системы и стоит ли продолжать ее использовать, либо вернуться к привычному способу обработки информации (как в базовом периоде).

Экономическая эффективность от внедрения проекта (Э) складывается из двух составляющих:

– косвенного эффекта, который характеризуется увеличением прибыли, привлечением новых покупателей, уменьшение количества рекламаций, получаемых от покупателей и т. д.;

– прямого эффекта, который характеризуется снижением трудовых, стоимостных показателей и увеличением числа довольных покупателей и сотрудников, работа которых упростилась в разы.

К трудовым показателям эффективности относятся следующие:

1) абсолютное снижение трудовых затрат (ΔT) в часах за год (формула 7):

$$\Delta T = T_0 - T_1, \quad (7)$$

где

T_0 – трудовые затраты в часах за год на обработку информации по базовому варианту;

T_1 – трудовые затраты в часах за год на обработку информации по предлагаемому варианту.

До внедрения разработанной информационной системы, время, которое тратили сотрудники организации на обработку информации, составляло 6 часов в день. Т. е. это 144 часа в месяц (6 ч/день * 24 рабочих дня в месяце) и 1728 часов в год (144 ч/месяц * 12 месяцев в году).

Таким образом, значение показателя трудовых затрат в начальном (T_0) периоде составляло 1728 ч/год.

После внедрения информационной системы, количество часов, требуемых на обработку информации, снизилось с 6 до 2 часов в день. Т. е. это, опять же, 48 часов в месяц (2 ч/день * 24 рабочих дня в месяце) и 576 часов в год (48 ч/месяц * 12 месяцев в году).

Значит значение показателя трудовых затрат по предлагаемому варианту (T_1) составило 576 ч/год.

С учетом известных показателей T_1 и T_0 , можем перейти к расчету абсолютного снижения трудовых затрат (ΔT), вычисляемому по формуле 2:

$$\Delta T = 1728 - 576 = 1152 \text{ ч/год};$$

2) коэффициент относительного снижения трудовых затрат (K_T) (формула 8):

$$K_T = \Delta T / T_0 * 100\%. \quad (8)$$

$$K_T = 1152 / 1728 * 100\% = 66,6\%;$$

3) индекс снижения трудовых затрат или повышение производительности труда (Y_T) (формула 9):

$$Y_T = T_0 / T_1. \quad (9)$$

$$Y_T = 1728 / 576 = 3 \text{ ч/год}.$$

Результат расчета трудовых показателей эффективности приведен в таблице 19.

Таблица 19 – Расчет трудовых показателей эффективности, ч/год

Показатель	T_0	T_1	ΔT	$K_T, \%$	Y_T
Трудовые затраты	1728	576	1152	66,6	3

Результаты расчетов подтверждают снижение трудовых затрат на обработку информации по регистрации и исполнению заявок.

К стоимостным показателям эффективности относятся такие показатели как:

1) абсолютное снижение стоимостных затрат (ΔC) в рублях за год, вычисляемое по формуле 10:

$$\Delta C = C_0 - C_1. \quad (10)$$

Стоимостные затраты на обработку информации складываются из заработной платы сотрудников организации, занимающихся обработкой этой ин-

формации. Если раньше на выполнение операций по обработке и комплектации заказов на складе требовалось 6 человек, то с внедрением информационной системы, количество сотрудников, занятых в этой сфере, снизилось до 4 человек, за счет того, что программа позволяет ускорить выполнение многих операций по регистрации, обработке, передаче информации и формированию исходящих документов.

С учетом того, что средняя заработная плата работников этой должности составляет 28 тыс./руб., а количество работников равно 6, рассчитаем уровень стоимостных затрат в базовом периоде. Расчет производится по следующей формуле (формула 11):

$$C_0 = C * P * M, \quad (11)$$

где

C – размер ежемесячной заработной платы одного сотрудника;

P – количество сотрудников, занятых обработкой информации;

M – количество месяцев в году, равное 12.

$$C_0 = 28 * 6 * 12 = 2 \text{ млн. руб./год.}$$

Для расчета показателя стоимостных затрат по предлагаемому периоду, будем использовать формулу 12, но с учетом того, что штат сотрудников сократился до 4 человек.

$$C_1 = C * P * M, \quad (12)$$

$$C_1 = 28 * 4 * 12 = 1,34 \text{ млн. руб./год.}$$

Зная значение показателей C_0 и C_1 , можно приступить к вычислению значения абсолютного снижения стоимостных затрат (ΔC), вычисляемого по формуле 4:

$$\Delta C = 2 - 1,34 = 0,66 \text{ млн. руб./год.}$$

2) коэффициент относительного снижения стоимостных затрат (K_C) (формула 13)

$$K_C = \Delta C / C_0 * 100\% , \quad (13)$$

$$K_C = 0,66 / 2 * 100\% = 33 \%$$

3) индекс снижения стоимостных затрат (Y_C), рассчитываемый по форму-

ле 14.

$$Y_C = C_0 / C_1. \quad (14)$$

$$Y_C = 2 / 1,34 = 1,5.$$

Результат расчета стоимостных показателей эффективности представлен в таблице 20.

Таблица 20 – Расчет стоимостных показателей эффективности, млн.руб./год

Показатель	C_0	C_1	ΔC	$K_C, \%$	Y_C
Стоимостные затраты	2	1,34	0,66	33	1,5

Расчет стоимостных показателей эффективности также показывает положительный эффект от внедрения ИС.

Помимо рассмотренных показателей целесообразно также рассчитать срок окупаемости затрат на внедрение проекта машинной обработки информации ($T_{ок}$) в годах по формуле 15:

$$T_{ок} = K_{\Pi} / \Delta C, \quad (15)$$

где

K_{Π} – затраты на создание проекта (проектирование и внедрение), вычисляемые по следующей формуле:

$$K_{\Pi} = K_{пр} + K_{тс} + K_{лс} + K_{по} + K_{ио} + K_{уч} + K_{оэ}, \quad (16)$$

где $K_{пр}$ – затраты на проектирование ИС, равные 0,06 млн. руб.;

$K_{тс}$ – затраты вычислительную технику, равные 0,1 млн. руб.;

$K_{лс}$ – затраты на создание линий связи локальных сетей, 0,05 млн. руб.;

$K_{по}$ – затраты на программное обеспечение, 0,1 млн. руб.;

$K_{ио}$ – затраты на формирование информационной базы, 0,04 млн. руб.;

$K_{уч}$ – затраты на обучение персонала, равные 0 руб.;

$K_{оэ}$ – затраты на опытную эксплуатацию, 0,03 млн. руб.

Затраты на формирование информационной базы ($K_{ио}$) относятся к условно-постоянной информации, но при разработке системы встает необходимость заполнения справочников системы.

Затраты на опытную эксплуатацию (K_{03}) считаются единовременными, т.к. они происходят только на начальных этапах внедрения системы, когда одновременно работают обе системы – базовая и разработанная система.

Структура единовременных затрат – это не только их составляющие, но и удельный вес отдельных статей затрат.

Итак, $K_{\Pi} = 0,06 + 0,1 + 0,05 + 0,1 + 0,04 + 0 + 0,03 = 0,38$ млн. руб.

Таким образом, затраты на создание проекта в сумме составят 380 тыс. руб. или 0,38 млн. руб., тогда

$$T_{ок} = 0,38 / 0,66 = 0,58.$$

Таким образом, срок окупаемости проекта составит около полугода – это время, за которое полностью окупится стоимость программы и затраты на ее внедрение в компанию, в результате чего снизится объем ручного ввода информации и повысится качество и оперативность обработки информации.

На основе выполненных расчетов, составим таблицу, наглядно отражающую экономическую эффективность внедрения проекта (таблица 21).

Таблица 21 – Показатели экономической эффективности внедрения проекта

Показатели	Базовый вариант	Проектный вариант
1. Трудовые затраты	1728 ч/год	576 ч/год
2. Стоимостные затраты	2 млн. руб./год	1,34 млн. руб./год
3. Абсолютное снижение – трудовых затрат – стоимостных затрат	1152 ч/год 0,66 млн. руб./год	
4. Коэфф. относительного снижения: – трудовых затрат – стоимостных затрат	66,6 % 33 %	

Продолжение таблицы 21

5. Индекс снижения – трудовых затрат – стоимостных затрат	3 ч/год 1,5	
6. Срок окупаемости	6 месяцев	

Планируемый уровень рентабельности проекта (P) равен:

$$P = \Delta C / K_{\Pi} * 100\% = 0,66 / 0,38 * 100\% = 73,6\% \quad (16)$$

Анализ проведенных расчетов показывает, что внедрение разработанного программного продукта в ООО «Современные смазочные материалы» позволит не только снизить трудовые затраты на обработку информации на 1152 ч/год, но также снизит расходы организации на содержание персонала на 0,66 млн. руб./год. Согласно этим данным, можно смело утверждать, что внедрение разработанного программного продукта станет экономически-выгодным решением для исследуемой организации.

ЗАКЛЮЧЕНИЕ

Для успешного функционирования ООО «Современные смазочные материалы» в условиях растущей конкуренции и требований со стороны покупателей к уровню оказываемых услуги качеству приобретаемых товаров, необходимо внедрение информационной системы автоматизированного учета складских операций.

Существующие на отечественном рынке решения для автоматизации учета на складе не соответствуют требованиям, предъявленным исследуемой компанией, в связи с чем, было принято решение о разработке новой и качественной системы, ориентированной конкретно под процессы ООО «Современные смазочные материалы».

В ходе разработки информационной системы для автоматизации учета на складе ООО «Современные смазочные материалы» были получены следующие результаты.

Проведенный анализ предметной области позволил выявить процессы учета, требующие автоматизации: поступление товаров, прием заказов, комплектование и реализация товаров. Перечисленные операции лежат в основе деятельности всей организации, поэтому к качеству проводимого в них учета предъявляются особые требования, не допускающие наличие ошибок и упущений, что не всегда достигается из-за большого объема обрабатываемой информации.

Анализ организационной структуры организации позволил выявить роли пользователей в разрабатываемой системе (администратор, менеджер, начальник склада) и обозначить задачи, которые выбранные группы пользователей смогут решать в рамках ИС.

Моделирование существующей технологии учета складских операций позволило выявить проблемы, с которыми сталкиваются менеджеры и работники склада в процессе выполнения своих обязанностей.

В целях поиска решения, способного устранить выявленные проблемы учета, был проведен анализ ИС от известных разработчиков, который показал, что наиболее рациональным решением для исследуемой организации является разработка собственной системы.

Модель предлагаемой технологии учета складских операций (т.е. после внедрения собственной ИС) должна устранить все существующие недостатки складского учета.

Для разработки системы была выбрана среда программирования «Delphi 7», так как данный язык является основным в организации, и СУБД «MySQL», так как организация уже много лет использует его.

Разработанная система состоит из 41 формы. База данных включает 16 таблиц. Программа делится на три модуля: «Администратор», «Начальник склада» и «Менеджер». Доступ к каждому модулю осуществляется посредством ввода логина и пароля.

ИС обеспечивает решение следующих задач: учет поступлений товаров от поставщиков; учет перемещения и списания товаров; комплектование товаров; учет заказов и заказанных товаров; учет работы сотрудников; учет поставщиков; учет клиентов; учет сроков исполнения заказов; контроль состояния склада; формирование исходящих документов и отчетов и т.п.

Разработанная информационная система хранит и обрабатывает информацию обо всех процедурах учета, чтобы на основе этих данных строить аналитические отчеты, отражающие слабые и сильные стороны организации.

Для облегчения работы пользователя с программой, предусмотрено «Руководство пользователя», поэтапно описывающее и наглядно демонстрирующее ход выполнения каждой операции в программе. Простой и удобный интерфейс программы позволяет разобраться в ней любому пользователю, не требуя специальной компьютерной подготовки.

Все кнопки программы сопровождаются всплывающими подсказками, что придает пользователю большую уверенность при работе с системой на начальных этапах ее внедрения.

Расчет показателей экономической эффективности проекта показал, что внедрение разработанного программного продукта в ООО «Современные смазочные материалы» позволит не только снизить трудовые затраты на обработку информации на 1152 ч/год, но также снизит расходы организации на содержание персонала на 0,66 млн. руб./год. Согласно этим данным, можно смело утверждать, что внедрение разработанного программного продукта станет экономически-выгодным решением для ООО «Современные смазочные материалы».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. «Автоматизация склада» от компании «Abmcloud» [Электронный ресурс]. – Режим доступа: http://abmcloud.com/solution/avtomatizatsiya-sklada/?utm_source=Yandex&utm_medium=cpc&utm_campaign=avtomatizaciya-sklada&yclid=5853935203144960945
2. «Мой склад» от компании «Логнекс» [Электронный ресурс]. – Режим доступа: <https://www.moysklad.ru/register/>
3. 1С: Предприятие 8. 1С-Логистика: Управление складом 3.0. [Электронный ресурс]. – Режим доступа: <https://rarus.ru/1c8/1c8-logistika-sklad-3/>
4. Агальцов, В. П. Базы данных. В 2-х т. Т. 2. Распределенные и удаленные базы данных: Учебник / В. П. Агальцов. – в 2-х т. Т. 2. – Москва : ИД ФОРУМ, НИЦ ИНФРА-М, 2013. – 272 с.
5. Введение в среду визуального программирования Delphi: Методические указания - Ч.2 / С. В. Борисов, С. С. Комалов, И. Л. Серебрякова и др.; Под ред. Б. Г. Трусова. – Москва : Изд-во МГТУ им. Н. Э. Баумана, 2011. – 96 с.
6. Вдовин, В. М. Предметно-ориентированные экономические информационные системы: учебное пособие / В. М. Вдовин, Л. Е. Суркова, А. А. Шурупов. – 3-е изд. – Москва : Дашков и К°, 2013. – 388 с.
7. Вдовин, В. М. Предметно-ориентированные экономические информационные системы: учебное пособие / В. М. Вдовин, Л. Е. Суркова, А. А. Шурупов. – 3-е изд. – Москва : Дашков и К°, 2013. – 388 с.
8. Глушаков, С. В. «Delphi 2007». Самоучитель» / С. В. Глушаков, А. Л. Клевцов. – Москва : АСТ, 2008. – 448 с.
9. Голицына, О. Л. Базы данных: Учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. – Москва : Форум, 2012. – 400 с.
10. Гольцман, В. MySQL 5.0. Библиотека программиста / В. Гольцман. – Санкт-Петербург : Питер, 2015. – 370 с.

11. Грекул, В. И. Проектирование информационных систем: курс лекций, учебное пособие / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 304 с.
12. Громов, Ю. Ю. Надежность информационных систем: учебное пособие / Ю. Ю. Громов, О. Г. Иванова, Н. Г. Мосягина, К. А. Набатов. – Тамбов : Изд-во ГОУ ВПО ТГТУ, 2003. – 160 с.
13. Диго, С. М. Базы данных: проектирование и использование: учебник / С. М. Диго. – Москва : Финансы и статистика, 2015. – 595 с.
14. Емельянов, В. И. Основы программирования на Delphi / В. И. Емельянов, В. И. Воробьев, Т. П. Тюрина; Под ред. В. М. Черненко. – Москва : Высшая школа, 2007. – 232 с.
15. Желонкин, А. Основы программирования в интегрированной среде Delphi / А. Желонкин. – Москва : Бинوم. Лаборатория знаний, 2004. – 240 с.
16. Избачков, Ю. С. Информационные системы: учебник для вузов / Ю. С. Избачков, В. Н. Петров. – 2-е изд. – Санкт-Петербург : Питер, 2015. – 656 с.
17. Информационная модель [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Информационная_модель
18. Карпова, И. П. Базы данных: Учебное пособие / И. П. Карпова. – Санкт-Петербург : Питер, 2013. – 240 с.
19. Кириллов, В. В. Введение в реляционные базы данных. Введение в реляционные базы данных / В. В. Кириллов, Г. Ю. Громов. – Санкт-Петербург : БХВ-Петербург, 2012. – 464 с.
20. Концептуальное логическое и физическое моделирование данных [Электронный ресурс]. – Режим доступа: <http://frameworx.ru/SID/datamodelling.html>
21. Коцюба, И. Ю. Основы проектирования информационных систем: курс лекций, учебное пособие / И. Ю. Коцюба, А. В. Чунаев, А. Н. Шиков. – Санкт-Петербург : Университет ИТМО, 2015. – 206 с.

22. Кэнту, М. Delphi 7: Для профессионалов / М. Кэнту. – Санкт-Петербург : Питер, 2004. – 1101 с.
23. Майерс, Г. Надежность программного обеспечения / Г. Майерс; пер. с англ. Ю. Ю. Галимова ; под ред. В. Ш. Кауфмана. – Москва : Мир, 1980. – 356 с.
24. Монахов, Ю. М. Функциональная устойчивость информационных систем. В 3 ч. Ч. 1. Надежность программного обеспечения : учеб. пособие / Ю. М. Монахов; Владим. гос. ун-т. – Владимир : Изд-во Владим. гос. ун-та, 2011. – 60 с.
25. ООО «Современные смазочные материалы» [Электронный ресурс]. – Режим доступа: <http://ccm-zakaz.ru/gpndst/dist17.nsf/all/m01>.
26. Описание функциональной модели [Электронный ресурс]. – Режим доступа: <http://resheniya.info/index.php?id=80&print=1>.
27. Пирогов, В. Ю. Информационные системы и базы данных: организация и проектирование : учеб. пособие / В. Ю. Пирогов. – Санкт-Петербург : БХВ-Петербург, 2009. – 528 с.
28. Плещев, В. В. Базы данных. Visual FoxPro, Access, SQL Server, Oracle с примерами и упражнениями : учеб. пособие / В. В. Плещев. – 4-е изд., испр. и доп. – Екатеринбург : Изд-во Урал. гос. экон. ун-та, 2014. – 323 с.
29. Плещев, В. В. Высокоуровневые методы информатики и программирования. Delphi 5, С++ Builder 5, Visual С++ с примерами и упражнениями : учебник / В. В. Плещев. – 2-е изд., испр. и доп. – Екатеринбург: Изд-во Урал. гос. экон. ун-та, 2014. – 255 с.
30. Плещев, В. В. Разработка и стандартизация программных средств и информационных технологий : учеб. пособие / В. В. Плещев. – Тюмень : Изд-во Тюменского гос. ун-та, 2011. – 310 с.
31. Плещев, В. В. Сравнительный анализ программных средств разработки приложений и баз данных и индивидуализация учебного процесса их изучения / В. В. Плещев // Известия УрГЭУ. – 2013. – №7. – С. 63–69.

32. Покупатель [Электронный ресурс]. – Режим доступа:
<https://ru.wikipedia.org/wiki/Покупатель>.
33. Продавец [Электронный ресурс]. – Режим доступа:
<https://ru.wikipedia.org/wiki/Продавец>.
34. Склад [Электронный ресурс]. – Режим доступа:
<https://ru.wikipedia.org/wiki/Склад>.
35. Товар [Электронный ресурс]. – Режим доступа:
<https://ru.wikipedia.org/wiki/Товар>.
36. Торговля [Электронный ресурс] . – Режим доступа:
<https://ru.wikipedia.org/wiki/Торговля>.
37. Тоу, Д. Настройка SQL: для профессионалов / Д. Тоу. – Санкт-Петербург : Питер, 2014. – 333 с.
38. Тюкачев, Н. А. Программирование в Delphi для начинающих / Н. А. Тюкачев, К. С. Рыбак, Е. Е. Михайлова. – Санкт-петербург : БХВ-Петербург, 2007. – 659 с.
39. Шеер, А. В. Моделирование бизнес-процессов / А. В. Шеер. – Москва : Весть-МетаТехнология, 2000. – 175 с.
40. Шполянская, И. Ю. Проектирование информационных систем с использованием UML и Rational Rose : методические указания по изучению дисциплины для студентов заочной формы обучения / И. Ю. Шполянская. – Ростов-на Дону : РГЭУ «РИНХ», 2003. – 45 с.

ПРИЛОЖЕНИЕ А

Описание обозначений базы данных

Primary key (PRI) – первичный ключ.

Foreign key (MUL) – внешний ключ.

Unsigned – данное свойство означает, что в столбце запрещены отрицательные значения.

Auto_increment – данное свойство обеспечивает автоматическую нумерацию строк таблицы.

Описание типов данных:

Integer или **Int** – целое число в диапазоне либо от -2 147 483 648 до 2 147 483 647, либо (если указано свойство **Unsigned**) от 0 до 4 294 967 295.

Date - дата в формате «уууу-мм-дд», в диапазоне от «0000-01-01» до «9999-12-31».

Varchar – символьная строка переменной длины, содержащая не более указанного количества символов (от 0 до 65 535).

Decimal или **Dec** - точное (неокругляемое) число с фиксированной точкой. Может содержать до 65 значащих цифр и до 30 цифр после десятичного разделителя (в нашем случае – 8 значащих цифр и 2 после десятичного разделителя).

ПРИЛОЖЕНИЕ Б

Перечень форм разработанной программы

Таблица Б.1 – Перечень форм разработанной программы

Форма	Название
Unit 1	Вход в систему
Unit 2	Заставка
Unit 3	Модуль «Администратор»
Unit 4	Справочник «Категории»
Unit 5	Модуль БД
Unit 6	Ввод названия категории
Unit 7	Справочник «Товары»
Unit 8	Ввод данных о товаре
Unit 9	Справочник «Сотрудники»
Unit 10	Ввод данных о сотруднике
Unit 11	Справочник «Поставщики»
Unit 12	Ввод данных о поставщике
Unit 13	Ввод данных о клиенте
Unit 14	Справочник «Склады»
Unit 15	Ввод данных о складе
Unit 16	Справочник «О компании»
Unit 17	Модуль «Начальник склада»
Unit 18	Поступление товаров
Unit 19	Ввод данных о поступлении товаров
Unit 20	Ввод данных о поступивших товарах
Unit 21	Состояние склада
Unit 22	
Unit 23	Ввод данных о ячейке хранения

Продолжение таблицы Б.1

Unit 24	Количество
---------	------------

Unit 25	Заказы
Unit 26	Просмотр и комплектование заказа
Unit 27	Модуль «Менеджер»
Unit 28	Заказы
Unit 29	Ввод данных о заказе
Unit 30	Ввод данных о заказанных товарах
Unit 31	Комплектование заказанных товаров
Unit 32	Параметры входа в систему
Unit 33	О программе
Unit 34	Руководство пользователя
Unit 35	Ввод количества
Unit 36	Списанные товары
Unit 37	Отчет по завершенным заказам
Unit 38	Отчет по реализованным товарам
Unit 39	Отчет о поступивших на склад товарах
Unit 40	Состояние ячеек склада
Unit 41	Справочник «Клиенты»

ПРИЛОЖЕНИЕ В

Листинг программы

```
unit Unit1;                                     end;

interface                                       var
                                                Form1: TForm1;

uses
    Windows, Messages, SysUtils,             implementation
    Variants, Classes, Graphics, Controls,
    Forms,                                     uses Unit5, Unit4, Unit3, Unit17,
    Dialogs, StdCtrls, Buttons;              Unit27;

type
    TForm1 = class(TForm)                      {$R *.dfm}
    Label1: TLabel;                            procedure
    Label2: TLabel;                            TForm1.BitBtn2Click(Sender:
    Edit1: TEdit;                               TObject);
    Edit2: TEdit;                               begin
    BitBtn2: TBitBtn;                           if (Edit1.Text=EmptyStr) or (Ed-
    Edit5: TEdit;                               it2.Text=EmptyStr) then begin
    Edit6: TEdit;                               ShowMes-
    Button1: TButton;                           sage('Введите логин и пароль');
    Button2: TButton;                           end
    Button3: TButton;                           else begin
    Button4: TButton;                           dm.t44.Close;
    Edit3: TEdit;                               dm.t44.SQL.Clear;
    procedure Bit-                             dm.t44.SQL.Add('Select * from
    Btn2Click(Sender: TObject);                sotrudniki where login LIKE
    procedure But-                             '"+Edit1.Text+"' and parol LIKE
    ton1Click(Sender: TObject);                '"+Edit2.Text+"'");
    procedure But-                             dm.t44.Open;
    ton2Click(Sender: TObject);               if dm.t44.RecordCount=0 then
    procedure But-                             begin
    ton3Click(Sender: TObject);               ShowMessage('Неверный логин или па-
    procedure But-                             роль. Попробуйте еще раз. ');
    ton4Click(Sender: TObject);               end
    private                                     else begin
    { Private declarations }                   Ed-
    public                                     it3.Text:=dm.t44.FieldByName('id_sotr-
    { Public declarations }                    ud').Value;
```

```

        Ed-
it5.Text:=dm.t44.FieldByName('fio').V
alue;
        Ed-
it6.Text:=dm.t44.FieldByName('rol').V
alue;
        Button1.Click;
        end
        end

        end;

        procedure
TForm1.Button1Click(Sender:
TObject);
        begin
            if Edit6.Text='Администратор'
then begin Button2.Click;
                end
            else if Edit6.Text='Менеджер'
then begin Button3.Click;
                end
            else if
Edit6.Text='Начальниксклада' then
begin Button4.Click;
                end
            end;

        procedure
TForm1.Button2Click(Sender:
TObject);
        begin
            Form3.WindowState:=wsMaxim
ized;
            Form3.show;
            Form1.Hide;
            end;

        procedure
TForm1.Button3Click(Sender:
TObject);
        begin

```

```

Form27.WindowState:=wsMaxi
mized;
        Form27.Label1.Caption:=Edit5.
Text;
        Form27.Label2.Caption:=Edit3.
Text;
        Form27.show;
        Form1.Hide;
        end;

        procedure
TForm1.Button4Click(Sender:
TObject);
        begin
            Form17.WindowState:=wsMaxi
mized;
            Form17.Label1.Caption:=Edit5.
Text;
            Form17.Label2.Caption:=Edit3.
Text;
            Form17.show;
            Form1.Hide;
            Form17.show;
            end;

        end.

        unit Unit2;

        interface

        uses
            Windows, Messages, SysUtils,
Variants, Classes, Graphics, Controls,
Forms,
            Dialogs, ExtCtrls, jpeg;

        type
            TForm2 = class(TForm)
                Image1: TImage;

```

```

    Timer1: TTimer;
    procedure Timer1Timer(Sender: TObject);
    procedure Image1Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form2: TForm2;

implementation

    {$R *.dfm}

    procedure
TForm2.Timer1Timer(Sender:
TObject);
    begin
        Timer1.Enabled := false;
    end;

unit Unit3;

interface

uses
    Windows, Messages, SysUtils,
Variants, Classes, Graphics, Controls,
Forms,
    Dialogs, Menus;

type
    TForm3 = class(TForm)
        ScrollBox1: TScrollBar;
        MainMenu1: TMainMenu;
        N2: TMenuItem;
        N7: TMenuItem;
        N15: TMenuItem;
        N16: TMenuItem;
        N8: TMenuItem;
        N17: TMenuItem;
        N12: TMenuItem;
        N19: TMenuItem;
        N20: TMenuItem;
        N5: TMenuItem;
        N6: TMenuItem;
        N23: TMenuItem;
        N24: TMenuItem;
        N3: TMenuItem;
        N9: TMenuItem;
        N10: TMenuItem;
        N1: TMenuItem;
        procedure N1Click(Sender:
TObject);
        procedure N8Click(Sender:
TObject);
        procedure N7Click(Sender:
TObject);
        procedure N15Click(Sender:
TObject);
        procedure N12Click(Sender:
TObject);
        procedure N20Click(Sender:
TObject);
        procedure FormClose(Sender:
TObject; var Action: TCloseAction);
        procedure N9Click(Sender:
TObject);
        procedure N6Click(Sender:
TObject);
        procedure N23Click(Sender:
TObject);
        procedure N24Click(Sender:
TObject);
        procedure N10Click(Sender:
TObject);
    private
        { Private declarations }
    public

```



```

    { Public declarations }
end;

var
  Form3: TForm3;

implementation

uses Unit4, Unit7, Unit9, Unit11,
Unit14, Unit16, Unit1, Unit33, Unit5,
  Unit37, Unit38, Unit39, Unit34;

{$R *.dfm}

procedure
TForm3.N1Click(Sender: TObject);
begin
  Form4.Parent:=Form3.ScrollBox
1;
  Form4.show;
end;

procedure
TForm3.N8Click(Sender: TObject);
begin
  Form7.Parent:=Form3.ScrollBox
1;
  Form7.show;
end;

procedure
TForm3.N7Click(Sender: TObject);
begin
  Form9.Parent:=Form3.ScrollBox
1;
  Form9.show;
end;

procedure
TForm3.N15Click(Sender: TObject);
begin
  Form11.Parent:=Form3.ScrollBo
x1;
  Form11.show;
end;

procedure
TForm3.N12Click(Sender: TObject);
begin
  Form14.Parent:=Form3.ScrollBo
x1;
  Form14.show;
end;

procedure
TForm3.N20Click(Sender: TObject);
begin
  Form16.Parent:=Form3.ScrollBo
x1;
  Form16.show;
end;

procedure
TForm3.FormClose(Sender: TObject;
var Action: TCloseAction);
  Var X: word;
begin
  X :=
MessageDlg('Вы действительно хотит
епокинуть программу?', MTConfirma-
tion, [MBYes, MBNo], 0);
  If X = mrYes then Form1.close
else Form1.Show;
end;

procedure
TForm3.N9Click(Sender: TObject);
begin
  Form33.Parent:=Form3.ScrollBo
x1;
  Form33.show;
end;

procedure
TForm3.N6Click(Sender: TObject);
begin

```

```

dm.t18.Close;
dm.t18.SQL.Clear;
dm.t18.SQL.Add('SELECT *
FROM zaprosi where status like "3a-
вершен");
dm.t18.Open;
Form37.Parent:=Form3.ScrollBo
x1;
Form37.show;
end;

procedure
TForm3.N23Click(Sender: TObject);
begin
dm.t32.Close;
dm.t32.Open;
Form38.Parent:=Form3.ScrollBo
x1;
Form38.show;
end;

procedure
TForm3.N24Click(Sender: TObject);
begin
Form39.Parent:=Form3.ScrollBo
x1;
Form39.show;
end;

procedure
TForm3.N10Click(Sender: TObject);
begin
Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.htm');
Form34.show;
end;

end.

unit Unit4;

```

```

interface

uses
  Windows, Messages, SysUtils,
  Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, cxGraphics, cxCon-
  trols, cxLookAndFeelFeels,
  cxLookAndFeelPainters,
  cxStyles, dxSkinsCore,
  dxSkinscxPCPainter, cxCustomData,
  cxFilter,
  cxData, cxDataStorage, cxEdit,
  DB, cxDBData, cxDBLookupCom-
  boBox,
  StdCtrls, Buttons, cxGridLevel,
  cxGridCustomTableView,
  cxGridTableView,
  cxGridDBTableView,
  cxClasses, cxGridCustomView,
  cxGrid, ExtCtrls,
  ComCtrls, ToolWin, ImgList,
  dxDockControl, dxDockPanel;

type
  TForm4 = class(TForm)
    Panel2: TPanel;
    cxGrid2: TcxGrid;
    cxGrid2DBTableView1:
  TcxGridDBTableView;
    cxGrid2Level1:
  TcxGridLevel;
    cxGrid2DBTableView1id_kateg:
  TcxGridDBColumn;
    cxGrid2DBTableView1nazvanie:
  TcxGridDBColumn;
    ImageList1: TImageList;
    ToolBar1: TToolBar;
    ToolButton1: TToolButton;
    ToolButton2: TToolButton;
    ToolButton3: TToolButton;

```

```

    ToolButton4: TToolButton;
    ToolButton5: TToolButton;
    ToolButton6: TToolButton;
    ToolButton7: TToolButton;
    ToolButton8: TToolButton;
    dxDockPanel1: TdxDockPan-
el;
    dxFloatDockSite1:
TdxFloatDockSite;
    Edit1: TEdit;
    procedure ToolBut-
ton1Click(Sender: TObject);
    procedure ToolBut-
ton2Click(Sender: TObject);
    procedure ToolBut-
ton3Click(Sender: TObject);
    procedure
cxGrid2DBTableView1CellDbClick(
    Sender: TcxCus-
tomGridView;
    ACellViewInfo:
TcxGridViewDataCellViewInfo;
AButton: TMouseButton;
    AShift: TShiftState; var
AHandled: Boolean);
    procedure ToolBut-
ton8Click(Sender: TObject);
    procedure ToolBut-
ton6Click(Sender: TObject);
    procedure Ed-
it1Change(Sender: TObject);
    procedure FormShow(Sender:
TObject);
    procedure ToolBut-
ton7Click(Sender: TObject);
    private
    { Private declarations }
    public
    { Public declarations }
    end;
var
    Form4: TForm4;

```

implementation

```

uses Unit5, Unit6, Unit34;

{$R *.dfm}

    procedure
TForm4.ToolButton1Click(Sender:
TObject);
    begin
    dm.t1.Insert;
    Form6.showModal;
    end;

    procedure
TForm4.ToolButton2Click(Sender:
TObject);
    begin
    Form6.showModal;
    end;

    procedure
TForm4.ToolButton3Click(Sender:
TObject);
    begin
    dm.t1.Delete;
    end;

    procedure
TForm4.cxGrid2DBTableView1CellD
bClick(
    Sender: TcxCus-
tomGridView;
    ACellViewInfo:
TcxGridViewDataCellViewInfo;
AButton: TMouseButton;
    AShift: TShiftState; var
AHandled: Boolean);
    begin
    ToolButton2.Click;
    end;

```

```

    procedure
TForm4.ToolButton8Click(Sender:
TObject);
    begin
    Form4.Close;
    end;

```

```

    procedure
TForm4.ToolButton6Click(Sender:
TObject);
    begin
    dxDockPanel1.Show;
    end;

```

```

    procedure
TForm4.Edit1Change(Sender:
TObject);
    begin
    if Edit1.Text=Emptystr then
begin
    dm.t1.Filtered:=false;
    end
    else begin
    dm.t1.Close;
    dm.t1.Filtered:=true;
    dm.t1.Filter:='nazvanie like
"%'+Edit1.Text+'%";
    dm.t1.Open;
    end
    end;

```

```

    procedure
TForm4.FormShow(Sender: TObject);
    begin
    dxDockPanel1.Hide;
    end;

```

```

    procedure
TForm4.ToolButton7Click(Sender:
TObject);
    begin

```

```

    Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.htm');
    Form34.show;
    end;

    end.

```

```

unit Unit6;

```

```

interface

```

```

uses

```

```

    Windows, Messages, SysUtils,
Variants, Classes, Graphics, Controls,
Forms,
    Dialogs, StdCtrls, Mask,
DBCtrls, ImgList, ComCtrls, ToolWin;

```

```

type

```

```

TForm6 = class(TForm)

```

```

    ToolBar1: TToolBar;

```

```

    ToolButton1: TToolButton;

```

```

    ToolButton3: TToolButton;

```

```

    ToolButton4: TToolButton;

```

```

    ToolButton6: TToolButton;

```

```

    ImageList1: TImageList;

```

```

    DBEdit1: TDBEdit;

```

```

    Label1: TLabel;

```

```

    procedure ToolBut-

```

```

ton1Click(Sender: TObject);

```

```

    procedure ToolBut-

```

```

ton6Click(Sender: TObject);

```

```

    procedure FormClose(Sender:

```

```

TObject; var Action: TCloseAction);

```

```

    procedure ToolBut-

```

```

ton3Click(Sender: TObject);

```

```

    procedure ToolBut-

```

```

ton4Click(Sender: TObject);

```

```

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form6: TForm6;

implementation

uses Unit5;

{$R *.dfm}

procedure
TForm6.ToolButton1Click(Sender:
TObject);
begin
  dm.t1.Edit;
  dm.t1.Post;
  Form6.Close;
end;

procedure
TForm6.ToolButton6Click(Sender:
TObject);
  Var X: word;
begin
  X :=
MessageDlg('Сохранить изменения?',
MTConfirmation, [MBYes, MBNo],
0);
  If X = mrYes then ToolBut-
ton1.Click
  else dm.t1.Cancel;
  Form6.Close;
end;

procedure
TForm6.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
dm.t1.Cancel;
end;

procedure
TForm6.ToolButton3Click(Sender:
TObject);
begin
  dm.t1.Prior;
end;

procedure
TForm6.ToolButton4Click(Sender:
TObject);
begin
  dm.t1.Next;
end;

end.

unit Unit7;

interface

uses
  Windows, Messages, SysUtils,
  Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, cxGraphics, cxCon-
  trols, cxLookAndFeels,
  cxLookAndFeelPainters,
  cxStyles, dxSkinsCore,
  dxSkinscxPCPainter, cxCustomData,
  cxFilter,
  cxData, cxDataStorage, cxEdit,
  DB, cxDBData, cxContainer,
  cxGridCustomTableView,
  cxGridTableView, cxGridDBTa-
  bleView, Buttons,
  StdCtrls, cxTextEdit,
  cxMaskEdit, cxDropDownEdit,
  cxLookupEdit,

```

```

    cxDBLookupEdit, cxD-
    BLookupComboBox, cxGridLevel,
    cxClasses,
    cxGridCustomView, cxGrid,
    ExtCtrls, ComCtrls, ToolWin, ImgList,
    DBCtrls,
    Mask, dxDockControl,
    dxDockPanel;

```

```

type
    TForm7 = class(TForm)
        Panel2: TPanel;
        cxGrid2: TcxGrid;
        cxGrid2DBTableView1:
        TcxGridDBTableView;
        cxGrid2Level1:
        TcxGridLevel;
        cxGrid2DBTableView1id:
        TcxGridDBCOLUMN;

```

```

        cxGrid2DBTableView1id_kateg:
        TcxGridDBCOLUMN;

```

```

        cxGrid2DBTableView1nazvanie:
        TcxGridDBCOLUMN;

```

```

        cxGrid2DBTableView1cena_ed:
        TcxGridDBCOLUMN;

```

```

        ImageList1: TImageList;
        ToolBar1: TToolBar;
        ToolButton1: TToolButton;
        ToolButton2: TToolButton;
        ToolButton3: TToolButton;
        ToolButton4: TToolButton;
        ToolButton5: TToolButton;
        ToolButton6: TToolButton;
        ToolButton7: TToolButton;
        ToolButton8: TToolButton;
        Panel5: TPanel;
        Label1: TLabel;
        Button1: TButton;
        DBEdit1: TDBEdit;

```

```

        DBLookupComboBox1:
        TDBLookupComboBox;
        dxDockPanel1: TdxDockPan-
        el;
        dxFloatDockSite1:
        TdxFloatDockSite;
        Edit1: TEdit;
        procedure ToolBut-
        ton1Click(Sender: TObject);
        procedure ToolBut-
        ton2Click(Sender: TObject);
        procedure ToolBut-
        ton3Click(Sender: TObject);
        procedure
        cxGrid2DBTableView1CellDbClick(
        Sender: TcxCus-
        tomGridTableView;
        ACellViewInfo:
        TcxGridTableDataCellViewInfo;
        AButton: TMouseButton;
        AShift: TShiftState; var
        AHandled: Boolean);
        procedure But-
        ton1Click(Sender: TObject);
        procedure cxDBLookupCom-
        boBox1PropertiesChange(Sender:
        TObject);
        procedure DBLookupCom-
        boBox1CloseUp(Sender: TObject);
        procedure ToolBut-
        ton8Click(Sender: TObject);
        procedure Ed-
        it1Change(Sender: TObject);
        procedure ToolBut-
        ton6Click(Sender: TObject);
        procedure FormShow(Sender:
        TObject);
        procedure ToolBut-
        ton7Click(Sender: TObject);
        private
        { Private declarations }
        public
        { Public declarations }

```

```

end;

var
  Form7: TForm7;

implementation

uses Unit5, Unit8, Unit34;

{$R *.dfm}

procedure
TForm7.ToolButton1Click(Sender:
TObject);
begin
  dm.t2.Insert;
  Form8.showModal;
end;

procedure
TForm7.ToolButton2Click(Sender:
TObject);
begin
  Form8.showModal;
end;

procedure
TForm7.ToolButton3Click(Sender:
TObject);
begin
  dm.t2.Delete;
end;

procedure
TForm7.cxGrid2DBTableView1CellD
blClick(
  Sender: TcxCus-
tomGridTableView;
  ACellViewInfo:
TcxGridTableDataCellViewInfo;
  AButton: TMouseButton;
  AShift: TShiftState; var
  AHandled: Boolean);
begin
  ToolButton2.Click;
end;

procedure
TForm7.Button1Click(Sender:
TObject);
begin
  dm.t2.Close;
  dm.t2.Filtered:=true;
  dm.t2.Filter:='[id_kateg] =
'+DBEdit1.Text+'";
  dm.t2.Open;
end;

procedure
TForm7.cxDBLookupComboBox1Pro
pertiesChange(Sender: TObject);
begin
  Button1.Click;
end;

procedure
TForm7.DBLookupComboBox1Close
Up(Sender: TObject);
begin
  Button1.Click;
end;

procedure
TForm7.ToolButton8Click(Sender:
TObject);
begin
  Form7.Close;
end;

procedure
TForm7.Edit1Change(Sender:
TObject);
begin
  X :=
MessageDlg('Сохранить изменения?',

```

```

MTConfirmation, [MBYes, MBNo],
0);
    If X = mrYes then ToolBut-
ton1.Click
    else dm.t6.Cancel;
    Form15.Close;
    end;

    procedure
TForm15.FormClose(Sender: TObject;
var Action: TCloseAction);
    begin
    dm.t6.Cancel;
    end;

    procedure
TForm15.ToolButton8Click(Sender:
TObject);
    begin
    dm.t6.Close;
    dm.t6.Open;
    dm.t7.Close;
    dm.t7.Open;
    end;

    procedure
TForm15.ToolButton11Click(Sender:
TObject);
    begin
    Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.htm');
    Form34.show;
    end;

    procedure
TForm15.ToolButton3Click(Sender:
TObject);
    begin
    dm.t6.Prior;
    end;

```

```

    procedure
TForm15.ToolButton4Click(Sender:
TObject);
    begin
    dm.t6.Next;
    end;

    end.

unit Unit16;

interface

uses
    Windows, Messages, SysUtils,
Variants, Classes, Graphics, Controls,
Forms,
    Dialogs, cxGraphics, cxCon-
trols, cxLookAndFeels,
cxLookAndFeelPainters,
    cxContainer, cxEdit,
dxSkinsCore, ComCtrls, ToolWin,
ImgList, cxTextEdit,
    cxMaskEdit, cxDropDownEdit,
cxLookupEdit, cxDBLookupEdit,
    cxDBLookupComboBox,
StdCtrls, Mask, DBCtrls;

type
TForm16 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;

```



```

DBEdit3: TDBEdit;
DBEdit4: TDBEdit;
DBEdit5: TDBEdit;
DBEdit6: TDBEdit;
DBEdit7: TDBEdit;
DBEdit8: TDBEdit;
DBEdit9: TDBEdit;
cxDBLookupComboBox1:
TcxDBLookupComboBox;
cxDBLookupComboBox2:
TcxDBLookupComboBox;
ImageList1: TImageList;
ToolBar1: TToolBar;
ToolButton1: TToolButton;
ToolButton6: TToolButton;
procedure ToolBut-
ton6Click(Sender: TObject);
procedure FormClose(Sender:
TObject; var Action: TCloseAction);
procedure ToolBut-
ton1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

var
Form16: TForm16;

implementation

uses Unit5;

{$R *.dfm}

procedure
TForm16.ToolButton6Click(Sender:
TObject);
Var X: word;
begin
X :=
MessageDlg('Сохранить изменения?',

```

```

MTConfirmation, [MBYes, MBNo],
0);
If X = mrYes then ToolBut-
ton1.Click
else dm.t15.Cancel;
Form16.Close;
end;

procedure
TForm16.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
dm.t15.Cancel;
end;

procedure
TForm16.ToolButton1Click(Sender:
TObject);
begin
dm.t15.Edit;
dm.t15.Post;
Form16.Close;
end;

end.

unit Unit17;

interface

uses
Windows, Messages, SysUtils,
Variants, Classes, Graphics, Controls,
Forms,
Dialogs, Menus, StdCtrls, Ex-
tCtrls;

type
TForm17 = class(TForm)
ScrollBox1: TScrollBox;
MainMenu1: TMainMenu;
N2: TMenuItem;

```

```

N1: TMenuItem;
N8: TMenuItem;
N7: TMenuItem;
N3: TMenuItem;
N9: TMenuItem;
N10: TMenuItem;
ScrollBox2: TScrollBox;
Panel1: TPanel;
Label1: TLabel;
Label2: TLabel;
N4: TMenuItem;
N5: TMenuItem;
procedure N8Click(Sender:
TObject);
    procedure N7Click(Sender:
TObject);
        procedure N1Click(Sender:
TObject);
            procedure N5Click(Sender:
TObject);
                procedure FormClose(Sender:
TObject; var Action: TCloseAction);
                    procedure N9Click(Sender:
TObject);
                        procedure N10Click(Sender:
TObject);
                            private
                                { Private declarations }
                            public
                                { Public declarations }
                            end;

var
    Form17: TForm17;

implementation

    uses Unit18, Unit22, Unit25,
        Unit5, Unit32, Unit1, Unit33, Unit35,
        Unit36,
            Unit34;

    {$R *.dfm}

    procedure
TForm17.N8Click(Sender: TObject);
    begin
        Form18.Parent:=Form17.ScrollB
ox1;
        Form18.show;
    end;

    procedure
TForm17.N7Click(Sender: TObject);
    begin
        dm.t16.Close;
        dm.t16.SQL.Clear;
        dm.t16.SQL.Add('SELECT *
from zapros1 where id_sklada like
'+Form22.DBEdit1.text+'');
        dm.t16.Open;
        dm.t17.Close;
        dm.t17.Open;
        Form22.Parent:=Form17.ScrollB
ox1;
        Form22.show;
    end;

    procedure
TForm17.N1Click(Sender: TObject);
    begin
        Form25.Parent:=Form17.ScrollB
ox1;
        dm.t18.Close;
        dm.t18.Open;
        dm.t12.Close;
        dm.t12.Open;
        Form25.show;
    end;

    procedure
TForm17.N5Click(Sender: TObject);
    begin
        Form32.Parent:=Form17.ScrollB
ox1;
        dm.t44.Close;

```

```

dm.t44.SQL.Clear;
dm.t44.SQL.Add('Select * from
sotrudniki where id_sotrud LIKE
"%'+Label2.Caption+'%"');
dm.t44.Open;
Form32.show;
end;

procedure
TForm17.FormClose(Sender: TObject;
var Action: TCloseAction);
  Var X: word;
  begin
  X :=
MessageDlg('Выдействительнохотит
епокинутьпрограмму?', MTConfirmation,
[MBYes, MBNo], 0);
  If X = mrYes then Form1.close
  else Form1.Show;
  end;

procedure
TForm17.N9Click(Sender: TObject);
  begin
  Form33.Parent:=Form17.ScrollB
ox1;
  Form33.show;
  end;

procedure
TForm17.N10Click(Sender: TObject);
  begin
  Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.htm');
  Form34.show;
  end;

end.

unit Unit18;

```

```

interface

uses
  Windows, Messages, SysUtils,
  Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, cxGraphics, cxCon-
  trols, cxLookAndFeelFeels,
  cxLookAndFeelPainters,
  cxStyles, dxSkinsCore,
  dxSkinscxPCPainter, cxCustomData,
  cxFilter,
  cxData, cxDataStorage, cxEdit,
  DB, cxDBData, ImgList, ComCtrls,
  ToolWin,
  cxGridLevel, cxGridCustom-
  TableView, cxGridTableView,
  cxGridDBTableView,
  cxClasses, cxGridCustomView,
  cxGrid, ExtCtrls, cxDBLookupCom-
  boBox;

type
  TForm18 = class(TForm)
    Panel2: TPanel;
    cxGrid2: TcxGrid;
    cxGrid2DBTableView1:
    TcxGridDBTableView;
    cxGrid2Level1:
    TcxGridLevel;
    ToolBar1: TToolBar;
    ToolButton1: TToolButton;
    ToolButton2: TToolButton;
    ToolButton3: TToolButton;
    ToolButton4: TToolButton;
    ToolButton7: TToolButton;
    ToolButton8: TToolButton;
    ImageList1: TImageList;

    cxGrid2DBTableView1id_postupl:
    TcxGridDBColumn;
    cxGrid2DBTableView1data:
    TcxGridDBColumn;

```

```

cxGrid2DBTableView1id_sklada:
TcxGridDBCColumn;

cxGrid2DBTableView1summa:
TcxGridDBCColumn;

cxGrid2DBTableView1Column1:
TcxGridDBCColumn;
    procedure ToolBut-
ton1Click(Sender: TObject);
    procedure ToolBut-
ton2Click(Sender: TObject);
    procedure ToolBut-
ton3Click(Sender: TObject);
    procedure
cxGrid2DBTableView1CellDbClick(
    Sender: TcxCus-
tomGridTableView;
    ACellViewInfo:
TcxGridTableDataCellViewInfo;
AButton: TMouseButton;
    AShift: TShiftState; var
AHandled: Boolean);
    procedure ToolBut-
ton8Click(Sender: TObject);
    procedure ToolBut-
ton4Click(Sender: TObject);
    procedure ToolBut-
ton7Click(Sender: TObject);
    private
    { Private declarations }
    public
    { Public declarations }
    end;

var
    Form18: TForm18;

implementation

uses Unit5, Unit19, Unit34;

```

```
{ $R *.dfm }
```

```

    procedure
TForm18.ToolButton1Click(Sender:
TObject);
    begin
        dm.t8.Insert;
        dm.t8.FieldName('data').Valu
e:=DateToStr(Date);
        Form19.showModal;
    end;

    procedure
TForm18.ToolButton2Click(Sender:
TObject);
    begin
        Form19.showModal;
    end;

    procedure
TForm18.ToolButton3Click(Sender:
TObject);
    Var X: word;
    begin
        if dm.t9.RecordCount=0 then
        begin
            X := MessageDlg('Вы действи-
тельно хотите удалить текущую за-
пись?', MTConfirmation, [MBYes,
MBNo], 0);
            If X = mrYes then
                dm.t8.Delete;
            end
            else begin
                ShowMessage('Сначала удалите све-
дения о поступивших товарах!');
            end
        end;
    end;

    procedure
TForm18.cxGrid2DBTableView1Cell
DbClick(

```

```

        Sender: TcxCustomGridTableView;
        ACellViewInfo:
TcxGridTableDataCellViewInfo;
        AButton: TMouseButton;
        AShift: TShiftState; var
        AHandled: Boolean);
    begin
        ToolButton2.Click;
    end;

    procedure
TForm18.ToolButton8Click(Sender:
TObject);
    begin
        Form18.Close;
    end;

    procedure
TForm18.ToolButton4Click(Sender:
TObject);
    begin
        dm.t8.Close;
        dm.t8.Open;
        dm.t9.Close;
        dm.t9.Open;
    end;

    procedure
TForm18.ToolButton7Click(Sender:
TObject);
    begin
        Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.htm');
        Form34.show;
    end;

end.

unit Unit19;

interface

```

```

uses
    Windows, Messages, SysUtils,
Variants, Classes, Graphics, Controls,
Forms,
    Dialogs, cxGraphics, cxCon-
trols, cxLookAndFeels,
cxLookAndFeelPainters,
    cxStyles, dxSkinsCore,
dxSkinscxPCPainter, cxCustomData,
cxFilter,
    cxData, cxDataStorage, cxEdit,
DB, cxDBData, cxContainer, StdCtrls,
Mask,
    DBCtrls, cxTextEdit,
cxMaskEdit, cxDropDownEdit,
cxLookupEdit,
    cxDBLookupEdit, cxD-
BLookupComboBox, ExtCtrls,
ImgList, cxGridLevel,
    cxGridCustomTableView,
cxGridTableView, cxGridDBTa-
bleView, cxClasses,
    cxGridCustomView, cxGrid,
ComCtrls, ToolWin, RpCon,
RpConDS, RpDefine,
    RpRave;

type
TForm19 = class(TForm)
    ToolBar1: TToolBar;
    ToolButton1: TToolButton;
    ToolButton3: TToolButton;
    ToolButton4: TToolButton;
    ToolButton6: TToolButton;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    cxGrid1: TcxGrid;
    cxGrid1DBTableView1:
TcxGridDBTableView;
    cxGrid1Level1:
TcxGridLevel;
    ToolBar2: TToolBar;

```

```

        ToolButton2: TToolButton;
        ToolButton5: TToolButton;
        ToolButton7: TToolButton;
        ToolButton8: TToolButton;
        ToolButton11: TToolButton;
        ImageList1: TImageList;
        ImageList2: TImageList;
        Panel1: TPanel;
        cxDBLookupComboBox1:
TcxDBLookupComboBox;
        Label1: TLabel;
        Label2: TLabel;
        DBEdit2: TDBEdit;
        Label3: TLabel;
        Button1: TButton;
        DBLookupComboBox1:
TDBLookupComboBox;
        Label4: TLabel;

cxGrid1DBTableView1id_post_prod:
TcxGridDBCColumn;

cxGrid1DBTableView1id_prod:
TcxGridDBCColumn;

cxGrid1DBTableView1kolichestvo:
TcxGridDBCColumn;

cxGrid1DBTableView1summa:
TcxGridDBCColumn;

cxGrid1DBTableView1id_kateg:
TcxGridDBCColumn;

cxGrid1DBTableView1id_yach_hran:
TcxGridDBCColumn;

cxGrid1DBTableView1ostatok:
TcxGridDBCColumn;
        Label5: TLabel;
        DBEdit1: TDBEdit;
        ToolButton9: TToolButton;
        RvProject19: TRvProject;
        RvDataSetConnection191:
TRvDataSetConnection;
        RvDataSetConnection192:
TRvDataSetConnection;
        RvDataSetConnection193:
TRvDataSetConnection;
        RvDataSetConnection194:
TRvDataSetConnection;
        RvDataSetConnection195:
TRvDataSetConnection;
        RvDataSetConnection196:
TRvDataSetConnection;
        RvDataSetConnection197:
TRvDataSetConnection;
        procedure ToolBut-
ton2Click(Sender: TObject);
        procedure ToolBut-
ton5Click(Sender: TObject);
        procedure ToolBut-
ton7Click(Sender: TObject);
        procedure ToolBut-
ton1Click(Sender: TObject);
        procedure
cxGrid1DBTableView1CellDblClick(
        Sender: TcxCus-
tomGridView;
        ACellViewInfo:
TcxGridViewTableDataCellViewInfo;
        AButton: TMouseButton;
        AShift: TShiftState; var
        AHandled: Boolean);
        procedure But-
ton1Click(Sender: TObject);
        procedure ToolBut-
ton9Click(Sender: TObject);
        procedure ToolBut-
ton6Click(Sender: TObject);
        procedure FormClose(Sender:
TObject; var Action: TCloseAction);
        procedure ToolBut-
ton8Click(Sender: TObject);
        procedure ToolBut-
ton11Click(Sender: TObject);

```

```

        procedure ToolBut-
ton3Click(Sender: TObject);
        procedure ToolBut-
ton4Click(Sender: TObject);
        private
        { Private declarations }
        public
        { Public declarations }
        end;

var
    Form19: TForm19;

implementation

uses Unit20, Unit21, Unit5,
Unit34;

{$R *.dfm}

procedure
TForm19.ToolButton2Click(Sender:
TObject);
begin
    dm.t8.Edit;
    dm.t8.Post;
    dm.t9.Insert;
    dm.t9.FieldName('kolichestvo
').Value:='1';
    Form20.showModal;
end;

procedure
TForm19.ToolButton5Click(Sender:
TObject);
begin
    Form20.showModal;
end;

procedure
TForm19.ToolButton7Click(Sender:
TObject);
    Var X: word;

```

```

begin
    if dm.t10.RecordCount=0 then
begin
        X := MessageDlg('Вы действи-
тельно хотите удалить текущую за-
пись?', MTConfirmation, [MBYes,
MNo], 0);
        If X = mrYes then
            dm.t9.Delete;
            Button1.Click;
        end
        else begin
            ShowMessage('Сначала удалите све-
дения о свойствах поступивших то-
варов из таблицы ниже');
        end
    end;

    procedure
TForm19.ToolButton1Click(Sender:
TObject);
begin
    dm.t8.Edit;
    dm.t8.Post;
    Form19.Close;
end;

    procedure
TForm19.cxGrid1DBTableView1Cell
DbClick(
        Sender: TcxCus-
tomGridView;
        ACellViewInfo:
TcxGridViewDataCellViewInfo;
        AButton: TMouseButton;
        AShift: TShiftState; var
AHandled: Boolean);
begin
    ToolButton5.Click;
end;

```

```

    procedure
TForm29.ToolButton5Click(Sender:
TObject);
    begin
    Form30.Label8.Visible:=false;
    Form30.Edit2.Visible:=false;
    Form30.Label9.Visible:=false;
    Form30.Edit1.Visible:=false;
    dm.t2.Close;
    dm.t2.Filtered:=true;
    dm.t2.Filter:='[id] =
'+Form30.DBEdit7.Text+'";
    dm.t2.Open;
    Form30.showModal;
    end;

```

```

    procedure
TForm29.ToolButton7Click(Sender:
TObject);
    Var X: word;
    begin
    X :=
MessageDlg('Выдействительнохотит
еудалитьтекущуюзапись?', MTCon-
firmation, [MBYes, MBNo], 0);
    If X = mrYes then
    dm.t19.Delete;
    end;

```

```

    procedure
TForm29.ToolButton1Click(Sender:
TObject);
    begin
    dm.t18.Edit;
    dm.t18.Post;
    Form29.Close;
    end;

```

```

    procedure
TForm29.cxGrid1DBTableView1Cell
DblClick(
    Sender: TcxCus-
tomGridView;

```

```

    ACellViewInfo:
TcxGridTableDataCellViewInfo;
    AButton: TMouseButton;
    AShift: TShiftState; var
    AHandled: Boolean);
    begin
    ToolButton5.Click;
    end;

    procedure
TForm29.N1Click(Sender: TObject);
    begin
    dm.t18.Edit;
    dm.t18.Post;
    end;

```

```

    procedure
TForm29.ToolButton6Click(Sender:
TObject);
    Var X: word;
    begin
    X :=
MessageDlg('Сохранитьизменения?',
MTConfirmation, [MBYes, MBNo],
0);
    If X = mrYes then ToolBut-
ton1.Click
    else dm.t18.Cancel;
    Form29.Close;
    end;

```

```

    procedure
TForm29.FormClose(Sender: TObject;
var Action: TCloseAction);
    begin
    dm.t18.Cancel;
    end;

```

```

    procedure
TForm29.ToolButton8Click(Sender:
TObject);
    begin
    dm.t19.Refresh;

```



```

end;

procedure
TForm29.BitBtn1Click(Sender:
TObject);
begin
dm.t4.Insert;
Form12.showModal;
end;

procedure
TForm29.BitBtn2Click(Sender:
TObject);
begin
Form11.showModal;
end;

procedure
TForm29.ToolButton9Click(Sender:
TObject);
begin
RvProject29.Execute;
end;

procedure
TForm29.ToolButton11Click(Sender:
TObject);
begin
Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.htm');
Form34.show;
end;

procedure
TForm29.ToolButton3Click(Sender:
TObject);
begin
dm.t18.Prior;
end;

```

```

procedure
TForm29.ToolButton4Click(Sender:
TObject);
begin
dm.t18.Next;
end;

end.

unit Unit30;

interface

uses
Windows, Messages, SysUtils,
Variants, Classes, Graphics, Controls,
Forms,
Dialogs, cxGraphics, cxCon-
trols, cxLookAndFeels,
cxLookAndFeelPainters,
cxContainer, cxEdit,
dxSkinsCore, ImgList, ComCtrls,
ToolWin, DBCtrls,
cxTextEdit, cxMaskEdit, cxD-
BEdit, StdCtrls, Mask,
cxDropDownEdit,
cxLookupEdit, cxD-
BLookupEdit, cxDBLookupCom-
boBox, cxSpinEdit;

type
TForm30 = class(TForm)
ToolBar1: TToolBar;
ToolButton1: TToolButton;
ToolButton6: TToolButton;
ImageList1: TImageList;
Label1: TLabel;
Label2: TLabel;
Label4: TLabel;
DBLookupComboBox1:
TDBLookupComboBox;
DBLookupComboBox2:
TDBLookupComboBox;

```

```

        cxDBSpinEdit1: TcxDB-
SpinEdit;
        Label8: TLabel;
        Button3: TButton;
        Label9: TLabel;
        DBEdit6: TDBEdit;
        Edit1: TEdit;
        DBEdit7: TDBEdit;
        DBEdit3: TDBEdit;
        Edit2: TEdit;
        procedure ToolBut-
ton1Click(Sender: TObject);
        procedure But-
ton3Click(Sender: TObject);
        procedure DBLookupCom-
boBox2CloseUp(Sender: TObject);
        procedure ToolBut-
ton6Click(Sender: TObject);
        procedure FormClose(Sender:
TObject; var Action: TCloseAction);
        private
        { Private declarations }
        public
        { Public declarations }
        end;

var
    Form30: TForm30;

implementation

uses Unit5, Unit29;

{$R *.dfm}

        procedure
TForm30.ToolButton1Click(Sender:
TObject);
        begin
        dm.t19.Edit;
        dm.t19.Post;
        Button3.Click;
        Form30.Close;
        end;

        procedure
TForm30.Button3Click(Sender:
TObject);
        var i,j,h:real;
        begin
        dm.t16.Close;
        dm.t16.SQL.Clear;
        dm.t16.SQL.Add('SELECT *
from zapros1 where id_prod like
"%'+DBEdit3.Text+'%" and id_sklada
like "%'+Form29.DBEdit6.Text+'%");
        dm.t16.Open;
        dm.t22.Close;
        dm.t22.Open;
        if dm.t16.RecordCount=0 then
        begin
        Edit1.Text:='0';
        Edit2.Text:='Отсутствует';
        Label8.Visible:=true;
        Edit2.Visible:=true;
        Label9.Visible:=true;
        Edit1.Visible:=true;
        end
        else if dm.t22.RecordCount=0
then begin
        i:=strtofloat(dm.t16.FieldByNam
e('sum(pp.ostatok)').Value);
        j:=0;
        h:=i-j;
        Edit1.Text:=floattostr(h);
        Edit2.Text:=floattostr(i);
        Label8.Visible:=true;
        Edit2.Visible:=true;
        Label9.Visible:=true;
        Edit1.Visible:=true;
        end
        else begin
        i:=strtofloat(dm.t16.FieldByNam
e('sum(pp.ostatok)').Value);
        j:=strtofloat(dm.t22.FieldByNam
e('sum(z.kolichestvo)').Value);

```

```

h:=i-j;
Edit1.Text:=floattostr(h);
Edit2.Text:=floattostr(i);
Label8.Visible:=true;
Label9.Visible:=true;
Label8.Visible:=true;
Edit2.Visible:=true;
Label9.Visible:=true;
Edit1.Visible:=true;
end
end;

procedure
TForm30.DBLookupComboBox2Close
Up(Sender: TObject);
begin
Button3.Click;
end;

procedure
TForm30.ToolButton6Click(Sender:
TObject);
Var X: word;
begin
X :=
MessageDlg('Сохранить изменения?',
MTConfirmation, [MBYes, MBNo],
0);
If X = mrYes then ToolBut-
ton1.Click
else dm.t19.Cancel;
Form30.Close;
end;

procedure
TForm30.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
dm.t19.Cancel;
end;

end.

```

```

unit Unit31;

interface

uses
Windows, Messages, SysUtils,
Variants, Classes, Graphics, Controls,
Forms,
Dialogs, cxGraphics, cxCon-
trols, cxLookAndFeels,
cxLookAndFeelPainters,
cxStyles, dxSkinsCore,
dxSkinscxPCPainter, cxCustomData,
cxFilter,
cxData, cxDataStorage, cxEdit,
DB, cxDBData, cxDBLookupCom-
boBox,
cxGridLevel, cxGridCustom-
TableView, cxGridTableView,
cxGridDBTableView,
cxClasses, cxGridCustomView,
cxGrid, cxSpinEdit, cxButtonEdit,
StdCtrls,
ExtCtrls, Menus, ComCtrls,
ToolWin, ImgList;

type
TForm31 = class(TForm)
cxGrid2: TcxGrid;
cxGridDBTableView1:
TcxGridDBTableView;
cxGridLevel1: TcxGridLevel;
PopupMenu1: TPopupMenu;
N1: TMenuItem;
cxGridDBTa-
bleView1id_svojtva: TcxGridDBColum-
n;
cxGridDBTa-
bleView1id_yach_hran: TcxGridDB-
Column;
cxGridDBTa-
bleView1id_prod: TcxGridDBColumn;

```



```

        procedure
TForm31.cxGridDBTableView1CellD
blClick(
    Sender: TcxCustomGridTableView;
    ACellViewInfo:
TcxGridTableDataCellViewInfo;
AButton: TMouseButton;
    AShift: TShiftState; var
AHandled: Boolean);
    begin
        ToolButton1.Click;
    end;

end.

unit Unit32;

interface

uses
    Windows, Messages, SysUtils,
    Variants, Classes, Graphics, Controls,
    Forms,
    Dialogs, StdCtrls, Mask,
    DBCtrls, ImgList, ComCtrls, ToolWin;

type
    TForm32 = class(TForm)
        Label1: TLabel;
        Label2: TLabel;
        DBEdit1: TDBEdit;
        DBEdit2: TDBEdit;
        ToolBar1: TToolBar;
        ToolButton1: TToolButton;
        ToolButton3: TToolButton;
        ImageList1: TImageList;
        procedure ToolButton1Click(Sender: TObject);
        procedure ToolButton3Click(Sender: TObject);
        procedure FormClose(Sender:
TObject; var Action: TCloseAction);

```

```

        private
        { Private declarations }
        public
        { Public declarations }
    end;

var
    Form32: TForm32;

implementation

uses Unit5;

{$R *.dfm}

    procedure
TForm32.ToolButton1Click(Sender:
TObject);
    begin
        dm.t44.Edit;
        dm.t44.Post;
        Form32.Close;
    end;

    procedure
TForm32.ToolButton3Click(Sender:
TObject);
        Var X: word;
    begin
        X := MessageDlg('Сохранить
изменения?', MTConfirmation,
[MBYes, MBNo], 0);
        If X = mrYes then ToolBut-
ton1.Click
        else dm.t44.Cancel;
        Form32.Close;
    end;

    procedure
TForm32.FormClose(Sender: TObject;
var Action: TCloseAction);
    begin
        dm.t44.Cancel;

```

```

end;                                { Public declarations }
                                     end;

end.

unit Unit35;                          var
                                     Form35: TForm35;

interface                             implementation

uses                                  uses Unit5, Unit36, Unit17,
    Windows, Messages, SysUtils,    Unit34;
    Variants, Classes, Graphics, Controls,
    Forms,
    Dialogs, cxGraphics, cxCon-
    trols, cxLookAndFeels,
    cxLookAndFeelPainters,
    cxStyles, dxSkinsCore,
    dxSkinscxPCPainter, cxCustomData,
    cxFilter,
    cxData, cxDataStorage, cxEdit,
    DB, cxDBData, cxGridCustomTa-
    bleView,
    cxGridTableView,
    cxGridDBTableView, cxGridLevel,
    cxClasses,
    cxGridCustomView, cxGrid,
    ComCtrls, ToolWin, ImgList, cxD-
    BLookupComboBox,
    StdCtrls, ExtCtrls, Buttons,
    cxContainer, cxTextEdit, cxMaskEdit,
    cxSpinEdit, cxDBEdit;

type
    TForm35 = class(TForm)
        Label1: TLabel;
        Panel1: TPanel;
        BitBtn1: TBitBtn;
        cxDBSpinEdit1: TcxDB-
        SpinEdit;
        procedure Bit-
        Btn1Click(Sender: TObject);
    private
        { Private declarations }
    public
                                     end.

                                     unit Unit36;

                                     interface

```

```

    uses
        Windows, Messages, SysUtils,
        Variants, Classes, Graphics, Controls,
        Forms,
        Dialogs, cxGraphics, cxCon-
        trols, cxLookAndFeels,
        cxLookAndFeelPainters,
        cxStyles, dxSkinsCore,
        dxSkinscxPCPainter, cxCustomData,
        cxFilter,
        cxData, cxDataStorage, cxEdit,
        DB, cxDBData, cxGridCustomTa-
        bleView,
        cxGridTableView,
        cxGridDBTableView, ImgList,
        cxGridLevel, cxClasses,
        cxGridCustomView, cxGrid,
        ComCtrls, ToolWin, cxD-
        BLookupComboBox;

```

```

type
    TForm36 = class(TForm)
        ToolBar1: TToolBar;
        ToolButton4: TToolButton;
        ToolButton5: TToolButton;
        ToolButton7: TToolButton;
        ToolButton8: TToolButton;
        cxGrid2: TcxGrid;
        cxGrid2DBTableView1:
        TcxGridDBTableView;
        cxGrid2Level1:
        TcxGridLevel;
        ImageList1: TImageList;

        cxGrid2DBTableView1id_spis:
        TcxGridDBColumn;

        cxGrid2DBTableView1id_yach_hran:
        TcxGridDBColumn;

        cxGrid2DBTableView1ostatok:
        TcxGridDBColumn;

```

```

        cxGrid2DBTableView1id_prod:
        TcxGridDBColumn;

        cxGrid2DBTableView1id_sklada:
        TcxGridDBColumn;

        cxGrid2DBTableView1Column1:
        TcxGridDBColumn;
        procedure ToolBut-
        ton4Click(Sender: TObject);
        procedure ToolBut-
        ton8Click(Sender: TObject);
        procedure ToolBut-
        ton7Click(Sender: TObject);
        private
            { Private declarations }
        public
            { Public declarations }
        end;

var
    Form36: TForm36;

implementation

uses Unit5, Unit34;

{$R *.dfm}

procedure
TForm36.ToolButton4Click(Sender:
TObject);
begin
    dm.t31.Close;
    dm.t31.Open;
end;

procedure
TForm36.ToolButton8Click(Sender:
TObject);
begin
    Form36.Close;

```

```

end;

procedure
TForm36.ToolButton7Click(Sender:
TObject);
begin
Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.htm');
Form34.show;
end;

end.

unit Unit37;

interface

uses
Windows, Messages, SysUtils,
Variants, Classes, Graphics, Controls,
Forms,
Dialogs, cxGraphics, cxCon-
trols, cxLookAndFeels,
cxLookAndFeelPainters,
cxStyles, dxSkinsCore,
dxSkinscxPCPainter, cxCustomData,
cxFilter,
cxData, cxDataStorage, cxEdit,
DB, cxDBData, cxDBLookupCom-
boBox, RpCon,
RpConDS, RpDefine, RpRave,
Buttons, StdCtrls, Mask, DBCtrls,
ComCtrls,
ExtCtrls, cxGridLevel,
cxGridCustomTableView,
cxGridTableView,
cxGridDBTableView,
cxClasses, cxGridCustomView,
cxGrid, ImgList, ToolWin;

type

```

```

TForm37 = class(TForm)
cxGrid2: TcxGrid;
cxGridDBTableView1:
TcxGridDBTableView;
cxGridLevel1: TcxGridLevel;
Panel2: TPanel;
Label2: TLabel;
Label3: TLabel;
DateTimePicker1:
TDateTimePicker;
DateTimePicker2:
TDateTimePicker;
Button2: TButton;
RvProject30: TRvProject;
RvDataSetConnection301:
TRvDataSetConnection;
RvDataSetConnection302:
TRvDataSetConnection;
RvDataSetConnection303:
TRvDataSetConnection;
RvDataSetConnection304:
TRvDataSetConnection;
cxGridDBTa-
bleView1id_zakaza: TcxGridDBCColumn;
cxGridDBTa-
bleView1data_dost: TcxGridDBCColumn;
cxGridDBTa-
bleView1id_menejera: TcxGridDB-
Column;
cxGridDBTableView1status:
TcxGridDBCColumn;
cxGridDBTa-
bleView1id_sklada: TcxGridDBCColumn;
cxGridDBTableView1data:
TcxGridDBCColumn;
cxGridDBTa-
bleView1data_zaversh: TcxGridDB-
Column;
ToolBar1: TToolBar;
ToolButton4: TToolButton;

```



```

        ToolButton5: TToolButton;
        ToolButton7: TToolButton;
        ToolButton8: TToolButton;
        ImageList1: TImageList;
        ToolButton1: TToolButton;
        ToolButton2: TToolButton;
        ToolButton3: TToolButton;
        procedure Button2Click(Sender: TObject);
        procedure ToolButton1Click(Sender: TObject);
        procedure ToolButton4Click(Sender: TObject);
        procedure ToolButton8Click(Sender: TObject);
        procedure ToolButton7Click(Sender: TObject);
        private
            { Private declarations }
        public
            { Public declarations }
        end;

    var
        Form37: TForm37;

    implementation

        uses Unit5, Unit29, Unit26,
            Unit34;

        {$R *.dfm}

        procedure TForm37.Button2Click(Sender:
            TObject);
            var StartDate, EndDate: string;
            begin
                StartDate:=formatDateTime('yyy
                y-mm-dd',DateTimePicker1.Date);
                EndDate:=formatDateTime('yyy
                y-mm-dd',DateTimePicker2.Date);
                dm.t18.Close;

                dm.t18.SQL.Clear;
                dm.t18.SQL.Text:='SELECT *
                FROM zaprosi z where (z.data between
                :StartDate AND :EndDate) and (sta-
                tus="Завершен")';
                dm.t18.Parameters.ParamByName
                ('StartDate').DataType:= ftDate;
                dm.t18.Parameters.ParamByName
                ('EndDate').DataType:= ftDate;
                dm.t18.Parameters.ParamValues[
                'StartDate']:=DateToStr(DateTimePick
                er1.Date);
                dm.t18.Parameters.ParamValues[
                'EndDate']:=DateToStr(DateTimePicke
                r2.Date);
                dm.t18.Open;
            end;

        procedure
            TForm37.ToolButton1Click(Sender:
            TObject);
            begin
                Form26.ShowModal;
            end;

        procedure
            TForm37.ToolButton4Click(Sender:
            TObject);
            begin
                dm.t18.Close;
                dm.t18.Open;
            end;

        procedure
            TForm37.ToolButton8Click(Sender:
            TObject);
            begin
                Form37.Close;
            end;

        procedure
            TForm37.ToolButton7Click(Sender:
            TObject);

```

```

begin
    Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.htm');
    Form34.show;
end;

end.

unit Unit38;

interface

uses
    Windows, Messages, SysUtils,
Variants, Classes, Graphics, Controls,
Forms,
    Dialogs, cxGraphics, cxCon-
trols, cxLookAndFeels,
cxLookAndFeelPainters,
    cxStyles, dxSkinsCore,
dxSkinscxPCPainter, cxCustomData,
cxFilter,
    cxData, cxDataStorage, cxEdit,
DB, cxDBData, cxGridCustomTa-
bleView,
    cxGridTableView,
cxGridDBTableView, ImgList,
ComCtrls, ToolWin, RpCon,
    RpConDS, RpDefine, RpRave,
StdCtrls, ExtCtrls, cxGridLevel,
cxClasses,
    cxGridCustomView, cxGrid,
cxDBLookupComboBox;

type
    TForm38 = class(TForm)
        cxGrid2: TcxGrid;
        cxGridDBTableView1:
TcxGridDBTableView;
        cxGridLevel1: TcxGridLevel;
        Panel2: TPanel;
        Label2: TLabel;
        Label3: TLabel;
        DateTimePicker1:
TDateTimePicker;
        DateTimePicker2:
TDateTimePicker;
        Button2: TButton;
        RvProject30: TRvProject;
        RvDataSetConnection301:
TRvDataSetConnection;
        RvDataSetConnection302:
TRvDataSetConnection;
        RvDataSetConnection303:
TRvDataSetConnection;
        RvDataSetConnection304:
TRvDataSetConnection;
        ToolBar1: TToolBar;
        ToolButton4: TToolButton;
        ToolButton5: TToolButton;
        ToolButton7: TToolButton;
        ToolButton8: TToolButton;
        ImageList1: TImageList;
        cxGridDBTa-
bleView1id_prod: TcxGridDBColumn;
        cxGridDBTa-
bleView1sumzpkolichestvo:
TcxGridDBColumn;
        cxGridDBTa-
bleView1id_kateg: TcxGridDBColum-
n;
        cxGridDBTa-
bleView1id_sklada: TcxGridDBColum-
n;
        cxGridDBTableView1data:
TcxGridDBColumn;
        ToolButton1: TToolButton;
        procedure But-
ton2Click(Sender: TObject);
        procedure ToolBut-
ton4Click(Sender: TObject);
        procedure ToolBut-
ton8Click(Sender: TObject);

```

```

        procedure ToolButton7Click(Sender: TObject);
        private
            { Private declarations }
        public
            { Public declarations }
        end;

var
    Form38: TForm38;

implementation

uses Unit5, Unit34;

{$R *.dfm}

procedure
TForm38.Button2Click(Sender:
TObject);
    var StartDate, EndDate: string;
    begin
        StartDate:=formatDateTime('yyy
y-mm-dd',DateTimePicker1.Date);
        EndDate:=formatDateTime('yyy
y-mm-dd',DateTimePicker2.Date);
        dm.t32.Close;
        dm.t32.SQL.Clear;
        dm.t32.SQL.Text:='SELECT *
FROM zpros5 z where (z.data be-
tween :StartDate AND :EndDate)';
        dm.t32.Parameters.ParamByNam
e('StartDate').DataType:= ftDate;
        dm.t32.Parameters.ParamByNam
e('EndDate').DataType:= ftDate;
        dm.t32.Parameters.ParamValues[
'StartDate']:=DateToStr(DateTimePick
er1.Date);
        dm.t32.Parameters.ParamValues[
'EndDate']:=DateToStr(DateTimePicke
r2.Date);
        dm.t32.Open;
    end;

```

```

        procedure
TForm38.ToolButton4Click(Sender:
TObject);
    begin
        dm.t32.Close;
        dm.t32.Open;
    end;

        procedure
TForm38.ToolButton8Click(Sender:
TObject);
    begin
        Form38.Close;
    end;

        procedure
TForm38.ToolButton7Click(Sender:
TObject);
    begin
        Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.htm');
        Form34.show;
    end;

end.

unit Unit39;

interface

uses
    Windows, Messages, SysUtils,
    Variants, Classes, Graphics, Controls,
    Forms,
    Dialogs, cxGraphics, cxCon-
    trols, cxLookAndFeels,
    cxLookAndFeelPainters,
    cxStyles, dxSkinsCore,
    dxSkinscxPCPainter, cxCustomData,
    cxFilter,

```

```

    cxData, cxDataStorage, cxEdit,
    DB, cxDBData, cxGridCustomTable
    View,
    cxGridTableView,
    cxGridDBTableView, ImgList,
    ComCtrls, ToolWin, RpCon,
    RpConDS, RpDefine, RpRave,
    StdCtrls, ExtCtrls, cxGridLevel,
    cxClasses,
    cxGridCustomView, cxGrid,
    cxDBLookupComboBox;

type
    TForm39 = class(TForm)
        cxGrid2: TcxGrid;
        cxGridDBTableView1:
        TcxGridDBTableView;
        cxGridLevel1: TcxGridLevel;
        Panel2: TPanel;
        Label2: TLabel;
        Label3: TLabel;
        DateTimePicker1:
        TDateTimePicker;
        DateTimePicker2:
        TDateTimePicker;
        Button2: TButton;
        RvProject30: TRvProject;
        RvDataSetConnection301:
        TRvDataSetConnection;
        RvDataSetConnection302:
        TRvDataSetConnection;
        RvDataSetConnection303:
        TRvDataSetConnection;
        RvDataSetConnection304:
        TRvDataSetConnection;
        ToolBar1: TToolBar;
        ToolButton4: TToolButton;
        ToolButton5: TToolButton;
        ToolButton7: TToolButton;
        ToolButton8: TToolButton;
        ImageList1: TImageList;
        cxGridDBTable
        View1id_prod: TcxGridDBColumn;

```

```

        cxGridDBTable
        View1sumppkolichestvo:
        TcxGridDBColumn;
        cxGridDBTable
        View1sumppsumma: TcxGridDB-
        Column;
        cxGridDBTable
        View1id_kateg: TcxGridDBCol-
        umn;
        cxGridDBTable
        View1id_sklada: TcxGridDBCol-
        umn;
        cxGridDBTableView1data:
        TcxGridDBColumn;
        ToolButton1: TToolButton;
        cxGridDBTable
        View1Column1: TcxGridDBCol-
        umn;
        procedure Button
        2Click(Sender: TObject);
        procedure ToolBut-
        ton4Click(Sender: TObject);
        procedure ToolBut-
        ton7Click(Sender: TObject);
        private
            { Private declarations }
        public
            { Public declarations }
        end;

var
    Form39: TForm39;

implementation

uses Unit5, Unit34;

{$R *.dfm}

procedure
TForm39.Button2Click(Sender:
TObject);
    var StartDate, EndDate: string;

```

```

begin
  StartDate:=formatDateTime('yyy
y-mm-dd',DateTimePicker1.Date);
  EndDate:=formatDateTime('yyy
y-mm-dd',DateTimePicker2.Date);
  dm.t33.Close;
  dm.t33.SQL.Clear;
  dm.t33.SQL.Text:='SELECT *
FROM zapros6 z where (z.data be-
tween :StartDate AND :EndDate)';
  dm.t33.Parameters.ParamByNam
e('StartDate').DataType:= ftDate;
  dm.t33.Parameters.ParamByNam
e('EndDate').DataType:= ftDate;
  dm.t33.Parameters.ParamValues[
'StartDate']:=DateToStr(DateTimePick
er1.Date);
  dm.t33.Parameters.ParamValues[
'EndDate']:=DateToStr(DateTimePicke
r2.Date);
  dm.t33.Open;

end;

```

```

procedure
TForm39.ToolButton4Click(Sender:
TObject);
begin
  dm.t33.Close;
  dm.t33.Open;
end;

```

```

procedure
TForm39.ToolButton7Click(Sender:
TObject);
begin
  Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.htm');
  Form34.show;
end;

```

```
end.
```

```

unit Unit40;

interface

uses
  Windows, Messages, SysUtils,
  Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, cxGraphics, cxCon-
  trols, cxLookAndFeels,
  cxLookAndFeelPainters,
  cxStyles, dxSkinsCore,
  dxSkinscxPCPainter, cxCustomData,
  cxFilter,
  cxData, cxDataStorage, cxEdit,
  DB, cxDBData, cxDBLookupCom-
  boBox,
  cxGridCustomTableView,
  cxGridTableView, cxGridDBTa-
  bleView, cxGridLevel,
  cxClasses, cxGridCustomView,
  cxGrid, ImgList, ComCtrls, ToolWin;

```

```

type
  TForm40 = class(TForm)
    ToolBar1: TToolBar;
    ToolButton4: TToolButton;
    ToolButton5: TToolButton;
    ToolButton6: TToolButton;
    ToolButton7: TToolButton;
    ToolButton8: TToolButton;
    ImageList1: TImageList;
    cxGrid2: TcxGrid;
    cxGrid2DBTableView1:
  TcxGridDBTableView;
    cxGrid2Level1:
  TcxGridLevel;
    cxGrid2DBTableView1id_yach_hran:
  TcxGridDBColumn;

```

```

cxGrid2DBTableView1mest_vsego:
TcxGridDBColumn;

cxGrid2DBTableView1zanyato:
TcxGridDBColumn;

cxGrid2DBTableView1svobodno:
TcxGridDBColumn;
    procedure ToolBut-
ton4Click(Sender: TObject);
    procedure ToolBut-
ton8Click(Sender: TObject);
    procedure ToolBut-
ton7Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form40: TForm40;

implementation

uses Unit5, Unit34;

{$R *.dfm}

    procedure
TForm40.ToolButton4Click(Sender:
TObject);
    begin
        dm.t35.Close;
        dm.t35.Open;
    end;

    procedure
TForm40.ToolButton8Click(Sender:
TObject);
    begin
        Form40.Close;
    end;

    procedure
TForm40.ToolButton7Click(Sender:
TObject);
    begin
        Form34.WebBrowser1.Navigate(
ExtractFilePath(Application.ExeName)
+ 'Помощь.htm');
        Form34.show;
    end;

end.

```