

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование кафедры)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Прикладная информатика в социальной сфере

(направленность (профиль)/специализация)

БАКАЛАВРСКАЯ РАБОТА

на тему «Автоматизация системы учета работы с документами для
ОАО «ВольтМастер»

Студент

Е.А. Мурзаков

(И.О. Фамилия)

(личная подпись)

Руководитель

Н.Н. Рогова

(И.О. Фамилия)

(личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский

(ученая степень, звание, И.О. Фамилия)

(личная подпись)

« _____ » _____ 20__ г.

Тольятти 2019

Аннотация

Целью выпускной квалификационной работы является разработка автоматизированной системы учёта работы с документами для ОАО «ВольтМастер».

Объект исследования – процессы хранения, обработки и ограничения доступа к электронным версиям документов, а также создание сервисов по резервированию, восстановлению и учету доступа пользователей к документам.

Предмет исследования - автоматизация учета работы с электронной документацией для сотрудников ОАО «ВольтМастер».

Пояснительная записка к выпускной квалификационной работе состоит из введения, трех глав, заключения, списка литературы и приложения.

Введение к выпускной квалификационной работе включает в себя описание цели и актуальности разработки системы, а также краткую структуру выпускной квалификационной работы.

В пояснительной записке к выпускной квалификационной работе проанализирована структура ОАО «ВольтМастер», изучены принципы существующей системы учета, выделены задачи требующие автоматизации, рассмотрены варианты решения данных задач существующими программными средствами, выполнена постановка задания на разработку собственной системы электронного учета для ОАО «ВольтМастер».

В 3 главе рассмотрены вопросы организационно-экономического обоснования процесса.

Приложение включают в себя листинг исходного кода модулей клиентской программы системы учета работы с документами.

Пояснительная записка выпускной квалификационной работы включает: 54 стр., 22 рисунков, 7 таблиц, 11 формул.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
Глава 1 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ ДЕЯТЕЛЬНОСТИ ОАО «ВОЛЬТМАСТЕР»	8
1.1 Техничко-экономическая характеристика ОАО «ВольтМастер»	8
1.1.1 Характеристика предприятия ОАО «ВольтМастер».....	8
1.1.2 Организация учета товаров, их классификация и себестоимость	9
1.1.3 Документация по учету движения комплектующих.....	11
1.2 Обзор и анализ аналогов системы учета работы с документами.....	12
1.3 Постановка задачи на разработку проекта создания автоматизированной системы учёта работы с документами	14
1.4 Разработка модели бизнес-процесса.....	15
Выводы по главе 1.....	19
Глава 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА РАБОТЫ С ДОКУМЕНТАМИ	20
2.1 Информационная модель системы и ее описание	20
2.1.1 Используемые идентификаторы и классификаторы	21
2.2 Логическая модель автоматизированной системы учета документов ...	22
2.2.1 Разработка диаграммы вариантов использования.....	22
2.2.2 Логическая модель базы данных.....	24
2.3 Описание требования к аппаратно-программному обеспечению.....	25
2.3.1 Обоснование выбора среды разработки	26
2.3.2 Архитектура сети ОАО «ВольтМастер»	28
2.4 Технологические процессы на предприятии.....	28
2.5 Пример реализации проекта и его описание.....	32
2.5.1 Работа со списками	35
2.5.2 Работа с товаром	35
Выводы по главе 2.....	39

Глава 3 ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА	41
3.1 Выбор и обоснование методики расчёта экономической эффективности	41
3.2 Расчёт показателей экономической эффективности проекта.....	43
Выводы по главе 3.....	49
ЗАКЛЮЧЕНИЕ	50
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	51
ПРИЛОЖЕНИЕ А.....	55

ВВЕДЕНИЕ

Укрупнение и консолидация торговых сетей, рост конкуренции и снижение прибыльности розничного бизнеса вызывает потребность в более качественных системах управления. Главная проблема руководства компаний в данном случае - это не только получения актуальной и полной информации о положении дел на предприятии, но и снижение затрат на эти системы. И поскольку, без стандартизации и автоматизации бизнес-процессов уже не обойтись, то для сохранения конкурентоспособности, предприятия малого и среднего бизнеса вынуждены снижать свои затраты на обслуживание. В связи с этим в ОАО «ВольтМастер» возникла необходимость создания автоматизированной системы учета документов, а не покупка уже существующих дорогостоящих аналогов.

Очень часто складывается ситуация, когда предприятие на определенном этапе своего развития сталкивается с необходимостью организации своей работы с использованием автоматизированной информационной системы на основе СУБД, но в силу их большой стоимости часто оттягивает момент внедрения или перехода на новую систему работы. Такое затягивание ведет часто к длительной регрессии в бизнесе, так как эти вложения не сразу и не прямо отражаются на ведении бизнеса. И только через некоторое время становится очевидным недальновидное поведение руководства.

Автоматизированная система, призвана решать целый ряд проблем, среди которых: учет и анализ товарооборота предприятия, должна содержать в себе комплекс решений для всех отделов и служб организации, таких как склад, отдел маркетинга, администрации, отдела сбыта и закупок. Функциональными особенностями проектируемой системы являются средства для обмена внутренней документацией, информация о транспортных поставках товаров, контроль и управление за складскими запасами, обработкой грузопотоков и упаковкой товаров.

Документооборот и учет документов предприятия – это процесс, направленный на снижение издержек, и в условиях роста объемов продаж,

автоматизированная информационная система такого типа позволяет существенно снизить временные затраты и человеческие ресурсы.

«В основе успешного бизнеса лежат использование новых информационных технологий и новых подходов к транспортировке и складской деятельности. К сожалению, в России такие технологии активно внедряют в основном только зарубежные фирмы, а отечественные компании с опаской смотрят на них. Но все же многие предприниматели, видя конкурентные преимущества новых систем, стремятся их внедрять» [2].

Применение подобных систем, как свидетельствует мировой и отечественный опыт, в торговле, позволяет существенно повысить ее эффективность. Это достигается за счет:

- значительного сокращения запасов товаров;
- максимального использования полезных площадей и объемов предприятий розничной и оптовой торговли;
- ускорение денежного и товарооборота предприятия;
- значительное сокращение транспортных расходов;
- сокращение затрат на переработки и ручного труда.

Актуальность данной разработки заключается в том, что для повышения эффективности работы сотрудников на предприятии ОАО «ВольтМастер», необходима разработка и внедрение качественной, отвечающей всем требованиям заказчика автоматизированной системы учета работы с документами.

Целью данной работы является разработка автоматизированной системы учета работы с документами для ОАО «ВольтМастер», позволяющей удовлетворить потребности работников предприятия.

В ходе выполнения данной работы необходимо выполнить следующие задачи:

1. Дать технико-экономическую характеристику деятельности ОАО «ВольтМастер»;

2. Провести анализ существующих подобных разработок, выявить их «слабые» стороны;

3. Разработать концептуальную и логическую модели системы и реализовать их;

4. Провести расчет экономической эффективности проекта.

Объектом исследования является создание автоматизированной системы учета работы с документами для ОАО «ВольтМастер».

Предметом работы является разработка автоматизированной системы учета.

Задача работы — это управление процессами, как на микро-, так и на макроуровне. Все они отличаются, как по функциональности, так и по составу решаемых задач. К основным задачам системы учета, можно отнести следующие:

- создание и оптимизация звеньев логистической цепи;
- управление редко изменяющимися данными;
- планирование продаж;
- общее управление документами;
- управление резервами и другие задачи.

Применение информационных систем учета предоставляет ряд преимуществ: увеличение скорости обмена информацией, уменьшения количества ошибок в учете, уменьшение объема бумажных отчетов, сведение всех процессов воедино.

Таким образом, можно подвести итог, что информационная оптимизация торговли – неизбежный и естественный путь развития предприятия, и залог его успешной конкурентоспособности

Глава 1 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ ДЕЯТЕЛЬНОСТИ ОАО «ВОЛЬТМАСТЕР»

1.1 Технико-экономическая характеристика ОАО «ВольтМастер»

1.1.1 Характеристика предприятия ОАО «ВольтМастер»

Открытое акционерное общество «ВольтМастер» (ОАО «ВольтМастер») создано в 2009 году.

ОАО «ВольтМастер» хранит комплектующие для производства и готовую продукцию на территории, занимающей общую площадь более 150м², на территории оборудованы складские помещения, подъездные пути, погрузочно-разгрузочные устройства.

Долгая и успешная работа позволила компании стать профессионалом в своём деле. В ОАО «ВольтМастер» создана необходимая организационная структура, имеются значительные материально-технические ресурсы (собственное офисное помещение, компьютерное и техническое оснащение, склад и т.п.), работают квалифицированные кадры, применяются современные технологии.

Коллектив компании работает очень слаженно, эффективно и быстро решает поставленные перед ними задачи.

Задачами, которые решает ОАО «ВольтМастер», являются:

- прием и оформление заказов;
- сборка из комплектующих ПК;
- подготовка отчётной документации;
- заключение договоров и иных сделок.

На рисунке 1.1 представлена организационная структура предприятия.

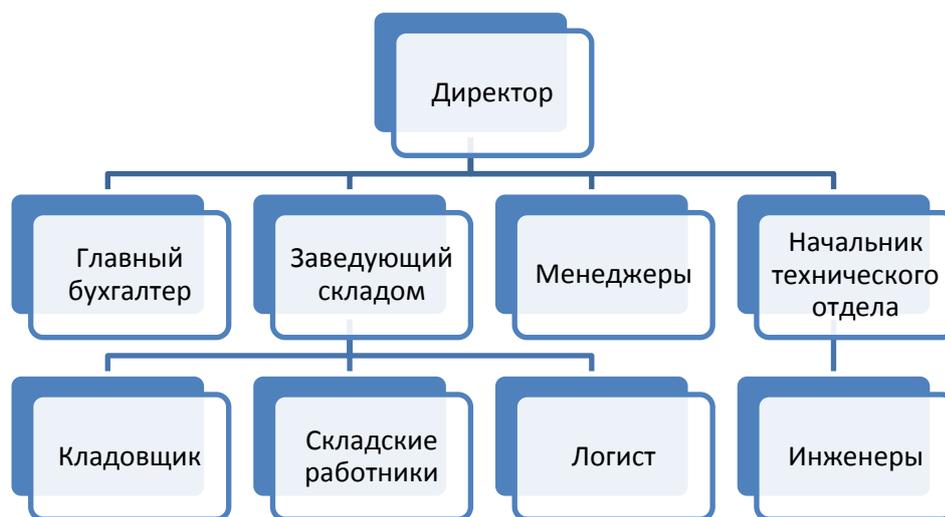


Рисунок 1.1 - Организационная структура предприятия ОАО «ВольтМастер»

Персонал компании состоит из 11 сотрудников. Каждый сотрудник организации оснащён компьютеризированным рабочим местом и имеет доступ к серверу документации.

1.1.2 Организация учета товаров, их классификация и себестоимость

Организация и учет реализуется посредством решения следующих прикладных задач:

- организация контроля за имеющимися материальными ресурсами в момент их хранения и на стадии обработки;

- документный учет производимых операций, отслеживание сопутствующих затрат, учет фактической себестоимости использованных материалов и их реализованной части;

- проведение систематического планового контроля над количеством запасов, мониторинг излишков и недостач;

- мониторинг работы с поставщиками товаров.

Для обеспечения действенного контроля над сохранностью материальных ценностей немаловажное значение имеют организация снабжения на предприятиях, а также состояние складского хозяйства.

Все материалы по способу их использования и назначения в

производственном процессе можно разделить таким образом:

- комплектующие;
- основные материалы;
- покупные материалы;
- вспомогательные материалы;
- топливо;
- запасные части;
- малоценные и быстроизнашивающиеся предметы.

Представленная выше классификация позволяет предприятию контролировать весь ход технологического процесса и вести имущественный учет. Решение задач по организации учета материалов требует ряда действий:

- наличие номенклатуры и ценников;
- отлаженная система документации и учета;
- регулярная инвентаризация остатков и избытков товаров с занесением полученных результатов в документы;
- внедрение автоматизированных систем учета.

Для правильной организации учета материалов на предприятии разрабатывается номенклатура и номенклатура-ценник.

В данном случае под термином номенклатура понимается систематизированный перечень наименований товаров, материалов и любых материальных ценностей, используемых в процессе купли-продажи на предприятии.

Номенклатура материальных ценностей содержит ряд сведений каждом материале или товаре:

- технически правильное наименование;
- характеристика основных сведений о товаре, таких как марка и сорт, единица измерения и размер, другое;
- номенклатурный номер, по существу являющийся условным обозначением подробных характеристик товара.

Среди наиболее распространенных – цифровая номенклатура материалов.

Наличие указанной в номенклатуре учетной цены каждого наименования товара или материала уже позволяет говорить о такой системе учета как ценник.

В процессе движения материалов и товаров каждый документ в обязательном порядке должен содержать не только наименование продукции, но также ее номенклатурный номер. Наличие правильно оформленной номенклатуры позволяет в дальнейшем избежать ошибок при заполнении приходных и расходных документов, а также при записях в складском и бухгалтерском учете материалов.

Расходы на содержание отдела снабжения, а также заводских складов в себестоимость материалов не включаются. Их относят на общезаводские (общехозяйственные) расходы предприятия.

1.1.3 Документация по учету движения комплектующих

«Для учета движения комплектующих применяется типовая учетная документация, отвечающая требованиям основных положений по учету материалов и приспособленная для автоматизированной обработки информации. Количество экземпляров выписываемых документов и их учет устанавливаются в зависимости от типа предприятий, особенностей организации их производства, материально-технического снабжения и от системы организации учета. Количество экземпляров должно быть минимальным: при использовании ЭВМ – один экземпляр, а при ручной обработке – не более двух» [6].

За поступлением и движением товаров на предприятии осуществляется поступенчатый контроль. В первую очередь контроль осуществляет отдел снабжения, в обязанностях которого организация работы с поставщиками. Кроме этого осуществляется работа за выполнением договорных обязательств, ведение споров с выставлением претензий по качеству и недоимкам материалов и товаров.

В случае осуществления такого вида доставки, как централизованная, со

склада поставщик выписывает следующие товарно-транспортные накладные:

- покупателю – для получения и приема материалов (может быть также выписан приходный ордер);
- поставщику – для учета за переданными материальными ценностями;
- водителям для начисления заработной платы за осуществление доставки грузов;
- для банка.

Как правило все приходные документы должны быть составлены в день поступления материалов. Основные реквизиты, отражаемые в приходных ордерах, накладных и актах – это наименование поставщика товара, пронумерованный список материалов, их сорт, размер (если требуется), количество единиц товара, цену за единицу товара, сумму, на которую получены материалы в целом, подписи лиц, принявших и сдавших материалы.

При внутреннем перемещении товаров на складе или между производственными помещениями используется внутренняя документация в виде накладных. В ней, как правило, отображается учет неиспользованных и сэкономленных товаров, брак производства или поставщика, естественная убыль посредством устаревания или обесценивания товаров. При таком виде учета от материально ответственного лица требуется документация в двух экземплярах: цеху или производству - для списания материалов, другой экземпляр - складу.

1.2 Обзор и анализ аналогов системы учета работы с документами

Заказчик рассматривал внедрение в работу предприятия три системы электронного учета «Дело» (платная), «BB workspace» и «Alfresco ECM». Для выявления достоинств и недостатков систем, а также принятия решения о внедрении был проведён их сравнительный анализ:

Система электронного учета «Дело» разработанная компанией «Электронные Офисные Системы» (ЭОС) – комплексное решение,

обеспечивающее автоматизацию процессов делопроизводства компаний различных масштабов и сфер деятельности, а также ведение полностью электронного учета в организации.

Система электронного учета «ВВ workspace» это комплексная система электронного учета и бюджетирования с человеко-ориентированной архитектурой НОА (Human Oriented Architecture).

Система «ВВ workspace» относится к классу ECM-систем (Enterprise Content Management) и поддерживает полный жизненный цикл управления документами от создания и регистрации, до архивного хранения в отдельных базах данных за каждый календарный год.

Для проведения сравнительного анализа аналогов все важные характеристики рассмотрены в таблице 1.1.

Таблица 1.1 - Сравнительный анализ выбранных заказчиком систем

№ п/п	Краткое описание требования	Дело	ВВ workspace	Alfresco ECM
1	Разграничение доступа к документам между пользователями и группами пользователей	+	+	+
2	Предоставление доступа к документу авторизованному пользователю с достаточным уровнем прав	+	+	+
3	Автозапуск документа в подходящей программе	-	-	-
4	Блокировка замены документа при работе с ним	+	+	-
5	Подробный лог работы пользователей с документами	-	-	-
6	Автосохранение старых версий документа	+	+	-
7	Доступ к системе из любого места, где доступны услуги сети Интернет	+	-	-

8	Простота использования и обучения персонала	-	-	+
9	Лёгкая установка и настройка	+	-	-

Продолжение таблицы 1.1

10	Простота переноса на другой сервер	+	+	-
11	Минимальная стоимость внедрения и эксплуатации	-	+	-

На основе данных, приведённых в таблице 1.1, был сделан вывод, что ни одно из рассмотренных программных средств полностью не удовлетворяет требованиям заказчика по функционалу. Следовательно, было принято решение разработать собственную автоматизированную систему учета документов.

1.3 Постановка задачи на разработку проекта создания автоматизированной системы учёта работы с документами

Для создания системы учета работы с документами для ОАО «ВольтМастер» необходимо разработать автоматизированную систему, удовлетворяющую следующим требованиям:

- все документы, необходимые для работы ОАО «ВольтМастер» будут храниться в электронной базе данных;
- каждый документ будет иметь категорию, название, описание, комментарий, время создания и последнего редактирования;
- в системе будет создан список пользователей. Каждый пользователь будет наделён соответствующим набором прав;
- каждый из документов будет соотноситься к определённой категории;
- автозапуск документа в соответствующей ему программе будет осуществляться с помощью специальной функции, которая автоматически выбирает нужную программу, установленную для данного вида документов в реестре операционной системы;

- редактируемый пользователем документ будет помечен в базе данных специальным статусом блокировки. Таким образом, другой пользователь в этот момент не сможет его заменить;
- в системе будет реализован поиск документа, что значительно упростит работу;
- сохранение старых копий документа при редактировании будет выполняться автоматически;
- при необходимости можно будет восстановить любую из сохранённых копий документа;
- по каждой операции при работе с документом будет вестись запись в специальном журнале;
- установка и развёртывание системы будет осуществляться автоматически с помощью специальной установочной программы;
- перенос системы на другой сервер будет осуществляться простым копированием файла базы данных;

Реализация перечисленных требований позволит выполнить все требования к разрабатываемой системе учета работы с документами для ОАО «ВольтМастер».

1.4 Разработка модели бизнес-процесса

Характеристика основного бизнес-процесса компании изображена на рисунке 1.2.

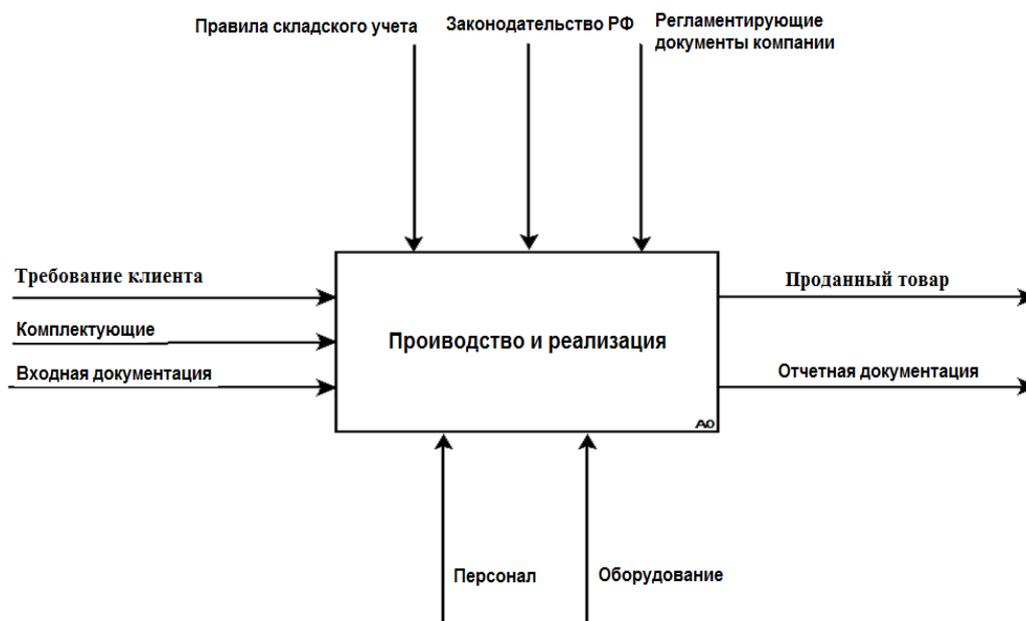


Рисунок 1.2 - Бизнес-процесс компании в нотации IDEF0

Основным видом деятельности компании ОАО «ВольтМастер», является оптовая и розничная реализация товаров. Этот процесс реализуется следующим образом: производится закупка комплектующих, складировается, согласно правил хранения, поставляется на производство в нужном объеме. Далее готовая продукция поступает на склад, отгружается и доставляется в точки реализации.

Декомпозиция данного процесса изображена на рисунке 1.3.

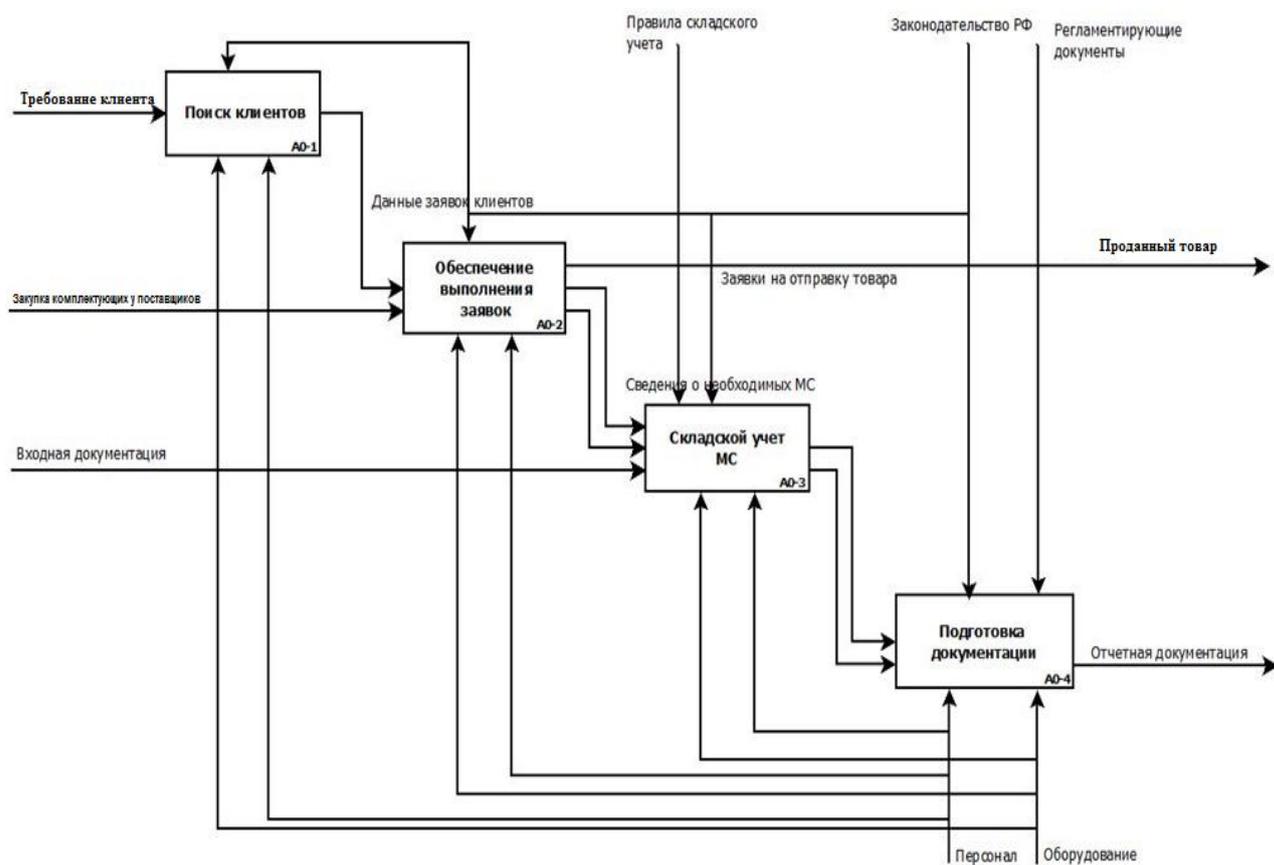


Рисунок 1.3 - Декомпозиция процесса в нотации IDEF0

В этих условиях существенно завышались временные затраты на работу с документами, происходили потеря и путаница с поступающими товарами, что в целом отрицательно сказывалось на получаемой предприятием прибыли.

В настоящее время система учета товаров на складе ведется с использованием стандартных таблиц Microsoft Excel. Этот способ требует достаточно много временных затрат и может приводить к частым и значительным ошибкам в вычислениях и получении информации. Кроме того, отсутствует возможность получения актуальной информации о наличии или отсутствии товара, так как не предусматривает синхронизацию данных в таблицах. Наблюдение и практика показали, что данный способ работы не позволяет с необходимой легкостью и актуальностью обрабатывать информацию и вести учет за товарами и местами их хранения.

Кроме того, для существующего учета также характерны следующие проблемы:

- невысокая скорость и точность выполнения заказов.
- неэффективное использование пространства склада.
- слабый контроль работы персонала.

«В связи с этим, руководством предприятия было принято решение о более полной автоматизации учета, то есть создании информационной системы для повышения качества учета и уменьшения временных затрат на обработку необходимых документов» [5].

Основной задачей проекта является автоматизация процесса учета работы с документами. Его основой являются следующие факторы:

- входная информация – заявка на поставку комплектующих от поставщика, заявка на отгрузку товара, сопроводительная документация, правила складского учета, законодательство РФ, регламентирующие документы компании;
- выходная информация – готовый товар, сопроводительная документация.

Исполнителями являются заведующий складом и складские работники.

Прием комплектующих на склад состоит из последовательности следующих действий: проверка правильности оформления документов, поиск места для расположения комплектующих, проверка правильности упаковки, прием комплектующих, оформление необходимой документации.

Выдача товара со склада состоит из последовательности следующих действий: проверка правильности заполнения документов, поиска места расположения товара, выдачи товара, внесения изменения в учетные документы.

Для улучшения качества деятельности работников и уменьшения допускаемых в их работе ошибок, необходимо введение системы автоматизации учета. В процессе работы используются определенные схемы учета и следующие основные документы:

- счет-фактура;
- заявка на отгрузку товара;

– товарная накладная.

Но существующая система учета очень сложна и громоздка, в следствие большого числа различных форм документов, многоэтапности их формирования, большого часть дублирования различных показателей.

В практическом выражении суть процесса выглядит таким образом, что менеджер, для удовлетворения потребностей покупателей вынужден использовать не самые быстрые способы работы. Так как наличие заказа уточняется по телефону, далее следует заполнение документации для получения товара со склада, а затем реализации товара конечному потребителю.

Такой же порядок получения товара организован на складе. Получение груза, документирование и размещение материалов требует значительных временных затрат от начальника склада.

«Такая постановка вопроса не позволяет вести своевременный и правильный учет, ведет к многочисленным расхождениям учетных данных, имеющегося на складе товара и результатов проводимых ревизий, что создает путаницу в складском хозяйстве и предпосылки для воровства» [4].

Итогами проводимой автоматизации учета должно стать: наличие точных сведений о товарах, имеющихся на складе в данный момент; получение актуальных и максимально полных сведений и характеристик товара или материала; ведение документооборота с учетом движения товара; учет товарных поступлений и сопутствующих расходов.

Выводы по главе 1

При анализе предметной области выявлена необходимость разработки собственной системы учета работы с документами для ОАО «ВольтМастер» в рамках разработки собственного комплекса программных средств. Определены основные функции, которые необходимо реализовать, чтобы проект удовлетворял требованиям к разрабатываемой системе.

Глава 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА РАБОТЫ С ДОКУМЕНТАМИ

2.1 Информационная модель системы и ее описание

«Информационная модель представляет собой схему движения входных, промежуточных и результативных потоков и функций предметной области. Кроме того, она объясняет, на основе каких входных документов и какой нормативно-справочной информации происходит выполнение функций по обработке данных и формирование конкретных выходных документов» [20].

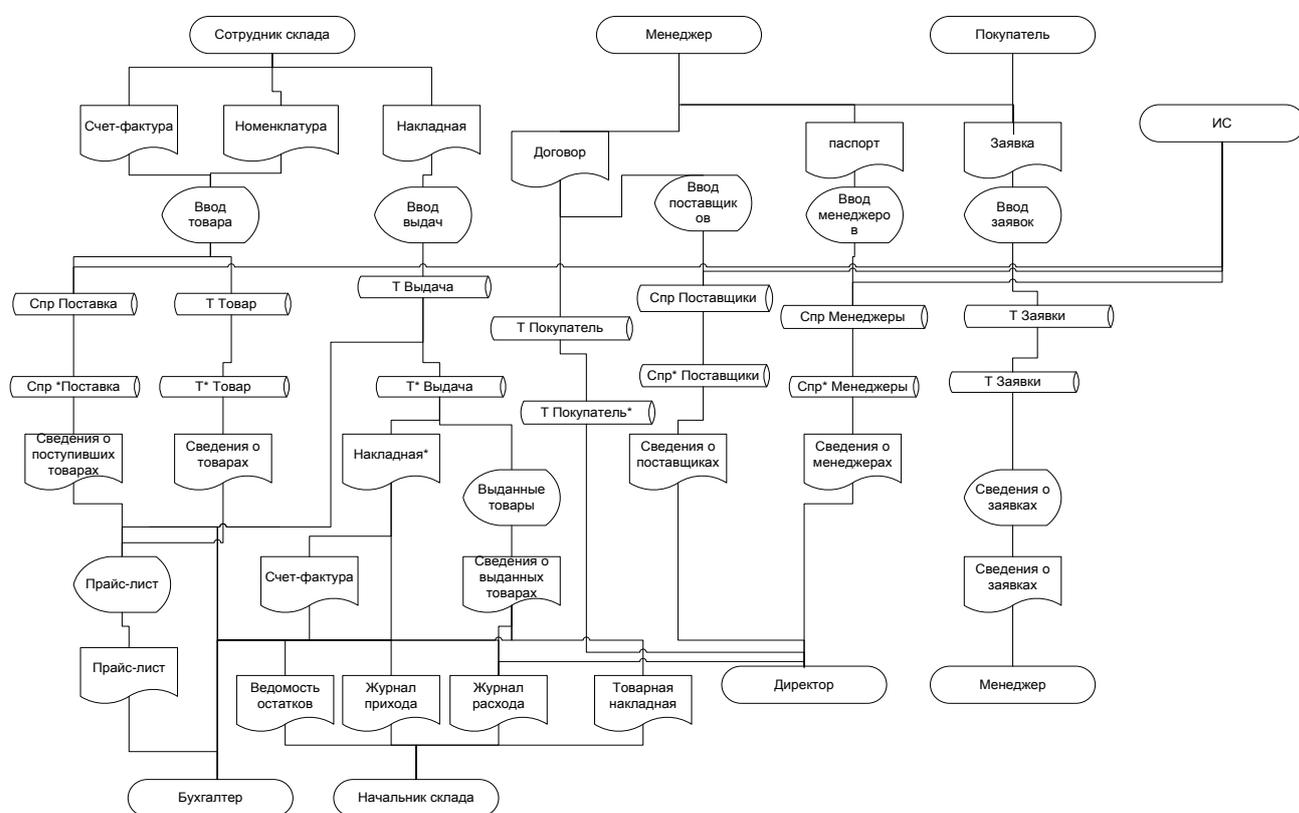


Рисунок 2.1 – Информационная модель системы

В соответствии с представленной информационной моделью при получении накладной на получение товара сотрудник склада вносит ее в базу данных, где указывает наименование товара, цену, количество и артикул. При этом вносятся изменения в два справочника – «Товар» и «Поставка».

Посредством запросов сотрудник имеет возможность получить актуальную информацию о поступивших товарах и имеющихся на складе товарах по отдельным реквизитам. А также сформировать прайс-лист отдельного или целого спектра товаров, где будет указаны только имеющиеся в наличии, в частности, на складе материалы.

Подобным образом происходит работа и при выдаче товара со склада. Сотруднику необходимо внести основные сведения о товаре, среди которых основные наименование товара или материала, количество единиц товара, стоимость товара, кому выдается (ФИО и должность). Дата и время занесения данных в систему формируется автоматически. Так как информация обновляется в режиме реального времени, то это дает возможность управляющему компании в любой момент запросить обновленные сведения, используя для доступа и формирования отчета один из реквизитов товара, наименование, дата выдачи, получатель и другое.

При получении заявки на поставку товара или материала менеджер фирмы также вносит информацию в систему, при этом в отчете автоматически проставляется дата и фамилия менеджера.

2.1.1 Используемые идентификаторы и классификаторы

В системе используется 4 вида кодирования, предназначенные для однозначной идентификации товара, поставки, выдачи, менеджера. Виды системы кодирования, указанные ниже, в таблице 2.1.

Таблица 2.1 - Используемые идентификаторы

№	Наименование кодируемого множества объектов	Значность кода	Система кодирования
1	Номер товара	4	Порядковая
2	Номер поставки	4	Порядковая
3	Номер выдачи	4	Порядковая
4	Номер менеджера	4	Порядковая

Идентификационный номер указывается для обеспечения однозначности учета и для избегания путаницы.

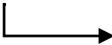
Классификатор товара.

Структурная формула классификатора:

F = [Наименование]: [XXXX]  Код

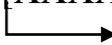
Классификатор **поставки**.

Структурная формула классификатора:

F = [Наименование]: [XXXX]  Код поставки

Классификатор **выдачи**.

Структурная формула классификатора:

F = [Наименование]: [XXXX]  Номер

Классификатор **менеджеров**.

Структурная формула классификатора:

F = [ФИО]: [XXXX]  Номер

2.2 Логическая модель автоматизированной системы учета документов

2.2.1 Разработка диаграммы вариантов использования

Дерево функций разрабатываемой системы представлено в виде модели вариантов использования в нотации UML представленная на рисунке 2.2.

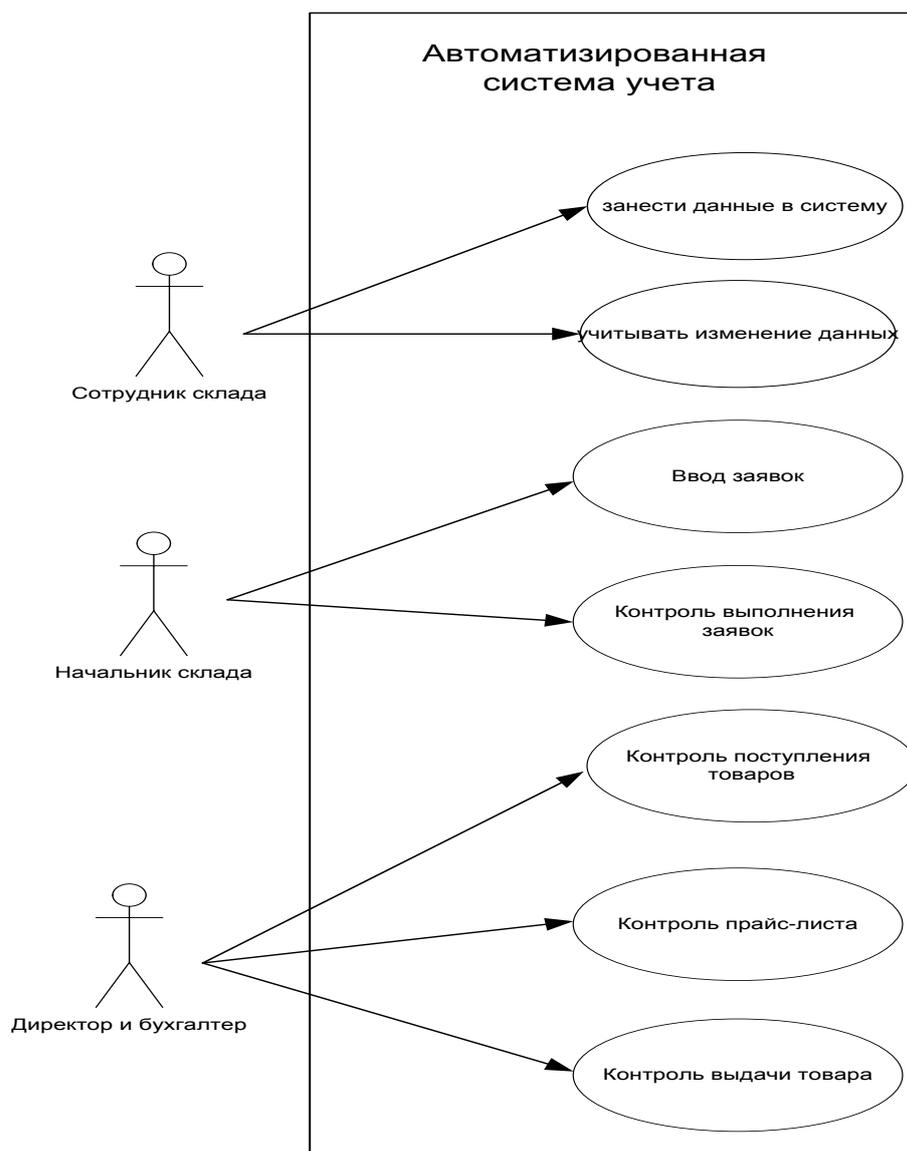


Рисунок 2.2 – Диаграмма вариантов использования системы в нотации UML

В основе использования автоматизированной системы лежит следующий алгоритм действий:

- внесение необходимых данных в систему (делается сотрудником предприятия)
- корректировка данных начальником подразделения;
- получение отчетов администрацией, на основе внесенных сведений;
- организация контроля и учета на всех уровнях работы с товаром.

Диаграмма была разработана для анализа требуемых функций пользователя.

2.2.2 Логическая модель базы данных

В системе используется СУБД MSSQL Server 2017. Разработанная база данных состоит из 33 таблиц, 173 полей и 29 реляционных связей. Фрагмент логической модели изображен на рисунке 2.3.

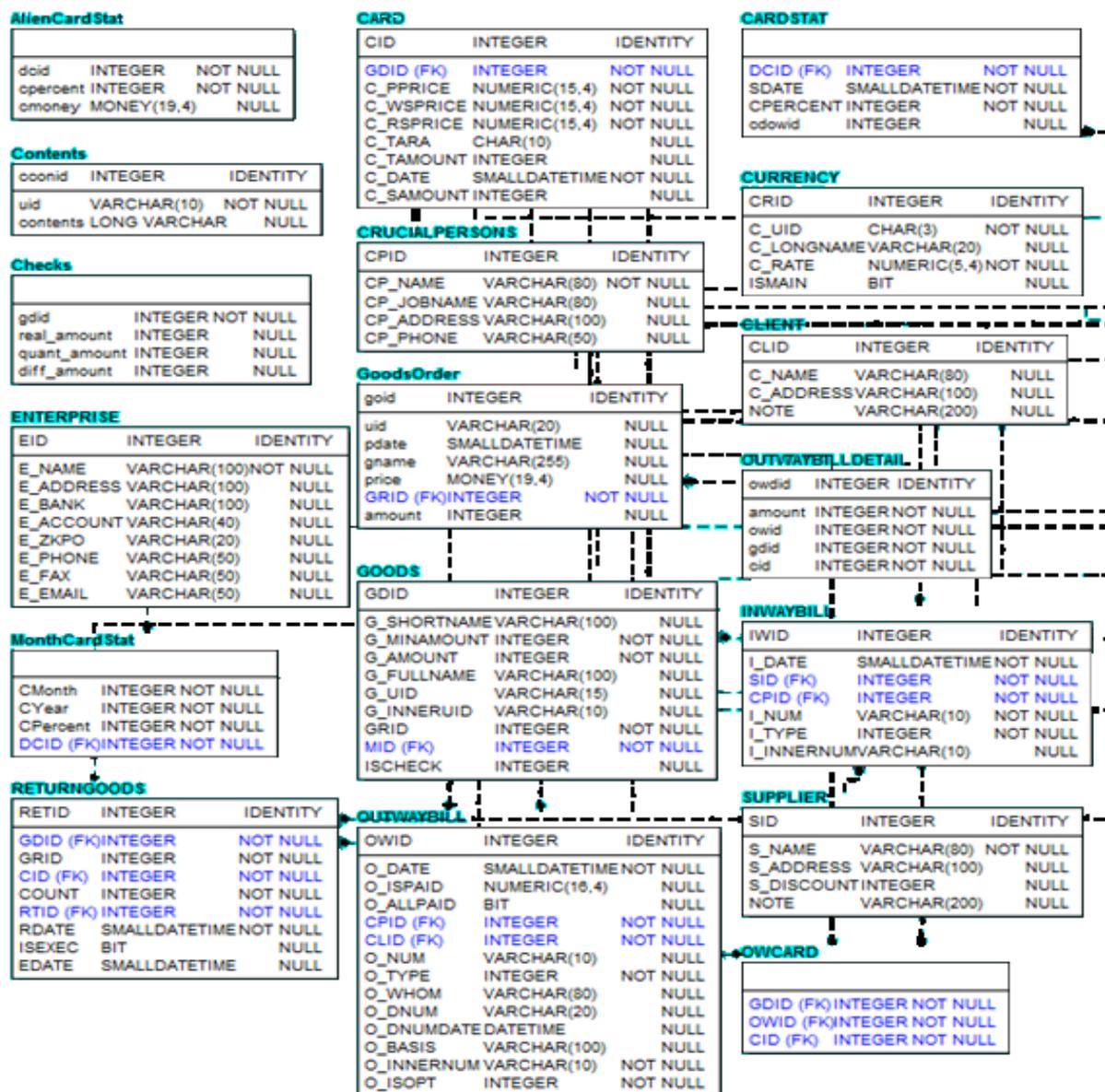


Рисунок 2.3 – Фрагмент логической модели базы данных

Описание таблиц базы данных:

- Goods - Товар;
- GroupGoods - Группы товаров;
- Client - Клиенты;
- Supplier - Поставщики;
- ClientDiscount - Скидки клиентам;

- Inwaybill - Приходные документы;
- InwaybillDetail - Состав приходных документов;
- Outwaybill - Расходные документы;
- OutwaybillDetail - Состав расходных документов;
- ReturnGoods - Возвращенный товар;
- CrucialPerson - Ответственные лица.

Логическая модель системы позволяет определить набор основных объектов базы и связи между ними.

2.3 Описание требования к аппаратно-программному обеспечению

Для функционирования приложения, реализующего учет работы с документами организации, необходимы следующие программные средства:

- ОС Windows 7 и выше (предназначенная для функционирования на клиентской машине).
- ОС Windows Server (или при наличии менее 10 сотрудников, работающих с «Автоматизированной системой интернет провайдера» - ОС Windows клиентского типа, предназначенная для функционирования на сервере и обеспечивающая многопользовательское подключение к базе данных).
- СУБД MS SQL Server 2017 (для обеспечения локального и сетевого функционирования системы учета на каждой машине). Данная СУБД выбрана, так как планируется относительно небольшая по объёму информационная система.

Автоматизированная система учета, предназначена для функционирования в ОС Windows, поэтому для эффективной работы системы необходимо обеспечить требования к архитектуре и параметрам технических средств, которые позволят более эффективно функционировать пользователям операционной системе.

Выбор ОС Windows объясняется удобным графическим пользовательским интерфейсом данного семейства ОС, а также тем, что

компания-заказчик использует именно эту ОС. «Многие фирмы-производители программного обеспечения, в том числе систем разработки приложений, также ориентированы на это семейство операционных систем, что привело к наличию на рынке большого количества систем программирования, позволяющих разрабатывать программное обеспечение для Windows» [4].

Сервер баз данных обслуживает базы данных и обеспечивает целостность и сохранность данных при их хранении, а также операциях ввода-вывода при доступе клиента к информации.

Сервер баз данных (СБД) под управлением серверной операционной системы работает и получает и запросы на языке SQL и обеспечивает выполнение следующих операций:

- создание новых данных и структур данных;
- сравнение, сортировку данных в табличном виде;
- изменение данных;
- удаление данных из базы;
- добавление данных в базу;
- передачу сообщения другим клиентам, подключенным к серверу.

Наиболее популярные среди СБД: Oracle, MS SQL Server, MySQL, и многие другие.

MS SQL – это сервер баз данных. Он ставится на любую машину Windows, а остальные рабочие станции подключаться к нему по сети.

Причиной выбора именно MS SQL является то, что СУБД MS SQL Server 2017 разработана компанией Microsoft, а, следовательно, хорошо сочетается с ОС Windows, и по стоимости приближается к стоимости отдельной базы данных. Также использование СУБД MS SQL Server 2017 в качестве СБД является требованием заказчика.

2.3.1 Обоснование выбора среды разработки

В ходе работы был проведен сравнительный анализ конкретных реализаций подходов к программированию. Согласно требованиям заказчика, к

функционалу и особенностям системы был выбран языка программирования Delphi версии 10.3.1 Rio.

«Delphi. Последняя на сегодняшний день версия Delphi является мощным универсальным средством разработки приложений, RAD-оболочкой. Оболочка основана на библиотеке VCL. В сравнении с C++ Builder производительность Pascal-кода, сгенерированного Delphi, всего на 4-5% меньше чем кода C++» [28].

К несомненным достоинствам языка программирования Delphi можно отнести простоту, быстроту и эффективность разработки. Delphi имеет один из самых быстрых компиляторов.

Недостатки языка программирования Delphi: статическое присоединение (linking) библиотеки VCL и компонентов к исполняемому файлу; в используемой Delphi парадигме (Forms) вся информация о форме включая свойства, настройки компонентов, значения по умолчанию, хранится в exe-файле; малое число параметров оптимизации кода. Кроме того, заметна тенденция к «разрастанию» exe-файлов, генерируемых Delphi.

В связи с вышесказанным можно выделить по каким критерием выбирается язык программирования Delphi.

«Система Delphi позволяет решать множество задач, в частности:

- создавать законченные приложения для Windows самой различной направленности: от чисто вычислительных и логических, до графических и мультимедиа.

- создавать (даже начинающим программистам) профессионально выглядящий оконный интерфейс для любых приложений.

- создавать мощные системы работы с локальными и удаленными базами данных

- создавать справочные системы (файлы. hlp) для своих приложений и мн. др.

Delphi - это комбинация нескольких важнейших технологий:

- визуальное (а, следовательно, и скоростное) построение приложений из программных прототипов
- масштабируемые средства для построения баз данных.
- высокопроизводительный компилятор в машинный код» [28].

2.3.2 Архитектура сети ОАО «ВольтМастер»

Разработанный программный продукт является клиент-серверным приложением, что позволяет подключать к базе данных одновременно большое количество клиентских машин.

На рисунке 2.4 изображена схема организации рабочих мест с использованием разработанной системы.

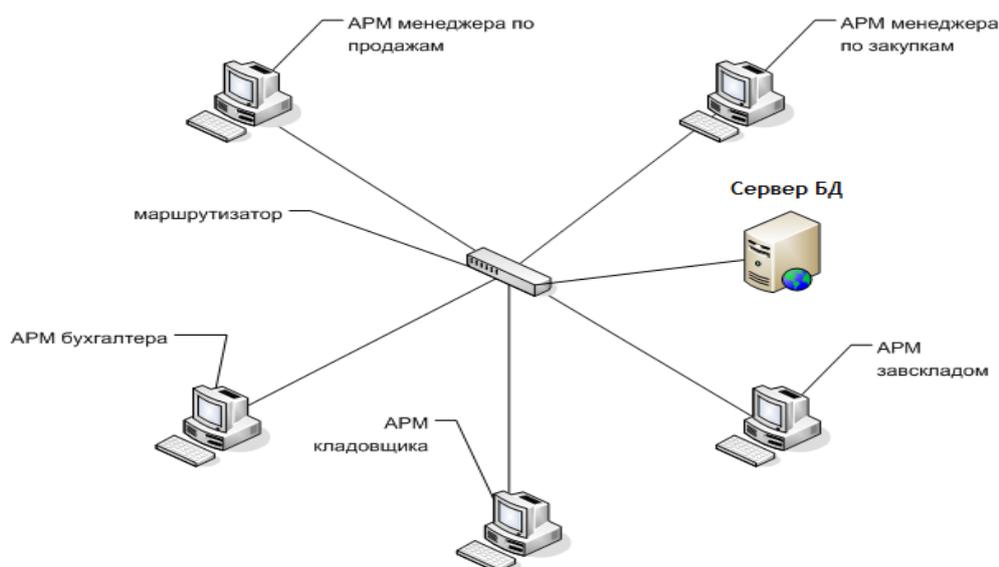


Рисунок 2.4 - Организация локальной сети

Данная схема подключения позволяет одновременно обрабатывать большие объемы информации. Данные, введенные одним пользователем, мгновенно становятся доступны другому пользователю, как для ознакомления, так и для редактирования.

2.4 Технологические процессы на предприятии

На уровне сотрудника работа с автоматизированной системой начинается с запуска программы и перехода в меню в состоянии «Ввод данных». Далее,

для ввода необходимых данных, выбирается необходимое поле и производится ручной ввод.

Заполнение полей реквизитов позволяет получить актуальный отчет на любой момент времени. В соответствии с запрашиваемыми параметрами производится отбор сведений и базы данных.

Отдельные таблицы базы данных содержат информацию о менеджерах, покупателях и заявках на товар или материал. Все эти сведения доступны сотруднику склада для формирования заказа. На основании заявки сотрудник формирует прайс-лист, содержащий наименование товары или материала, количество данной продукции в наличии на складе с указанием актуальной цены.

Схема, представленная на рисунке 2.5 указывает точные алгоритмы последовательности процессора сбора информации, ее передачи и обработки информации, а затем выдачи.

На начальном этапе происходит подготовка склада к приему товара, проверка сопутствующей документации и проведение разгрузочных работ. На каждом этапе формируется свой блок документов, таких как накладная на получение товара или материала, составление акта приемки, оформление приходских документов и формирование отчетов.

На схеме продемонстрировано место базы данных в движение потоков документов на каждом отдельном этапе.

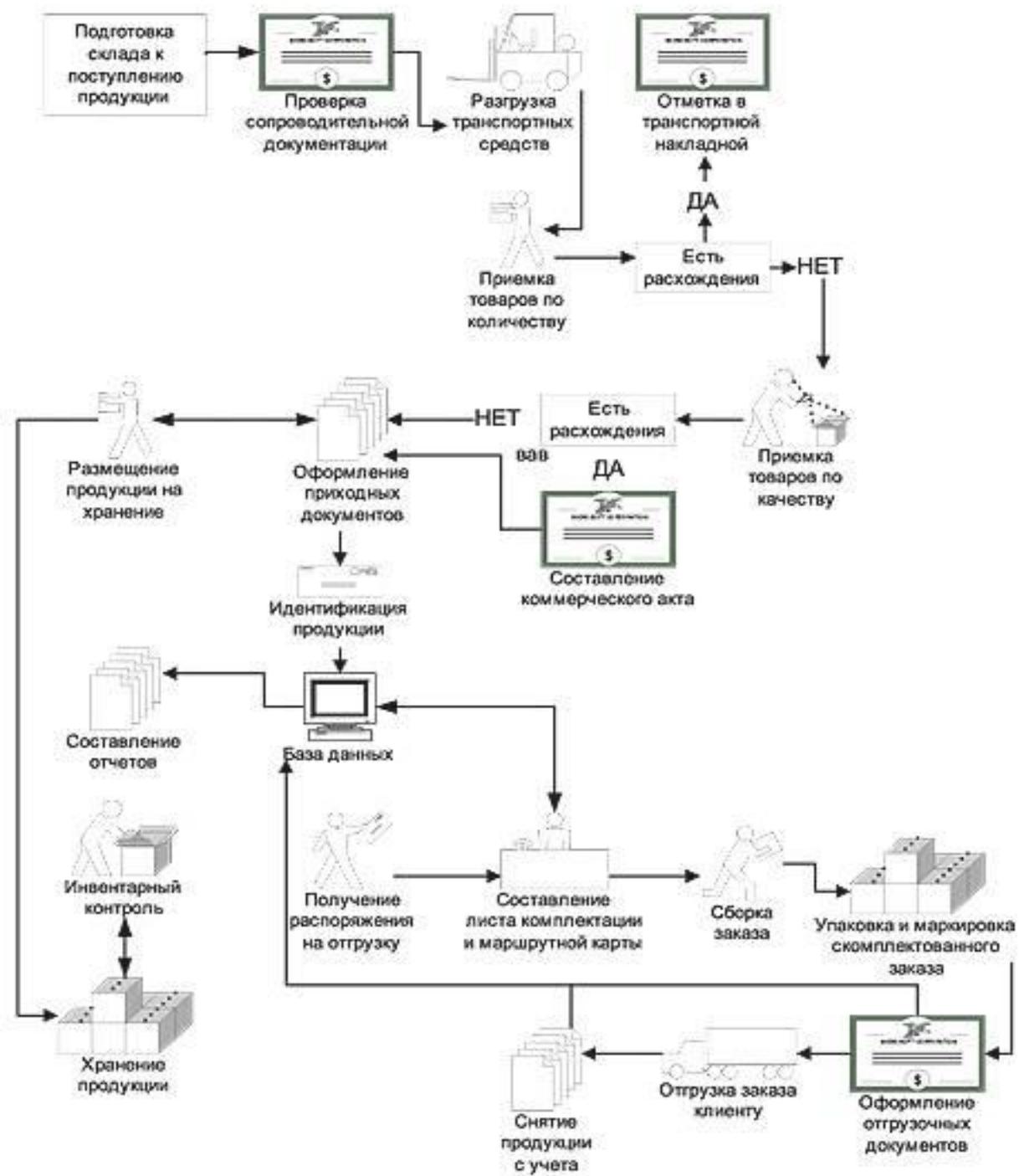


Рисунок 2.5 - Схема технологического процесса

Сотрудник склада, при необходимости внести какие-либо данные, открывает систему и переходит в меню «Ввод данных». В этом пункте он выбирает нужный подпункт, где последовательно вводит необходимые данные.

Кроме того, сотруднику склада доступен ввод информации о менеджерах, покупателях и заявках на какой-либо продукт. Эти данные сохраняются в отдельных таблицах базы данных.

Также сотрудник склада имеет возможность сформировать прайс-лист, который состоит из наименования и количества имеющихся на складе товаров с указанием их цены» [8].

Схема технологических процессов сбора, передачи, обработки и выдачи информации представлена на рисунке 2.6.

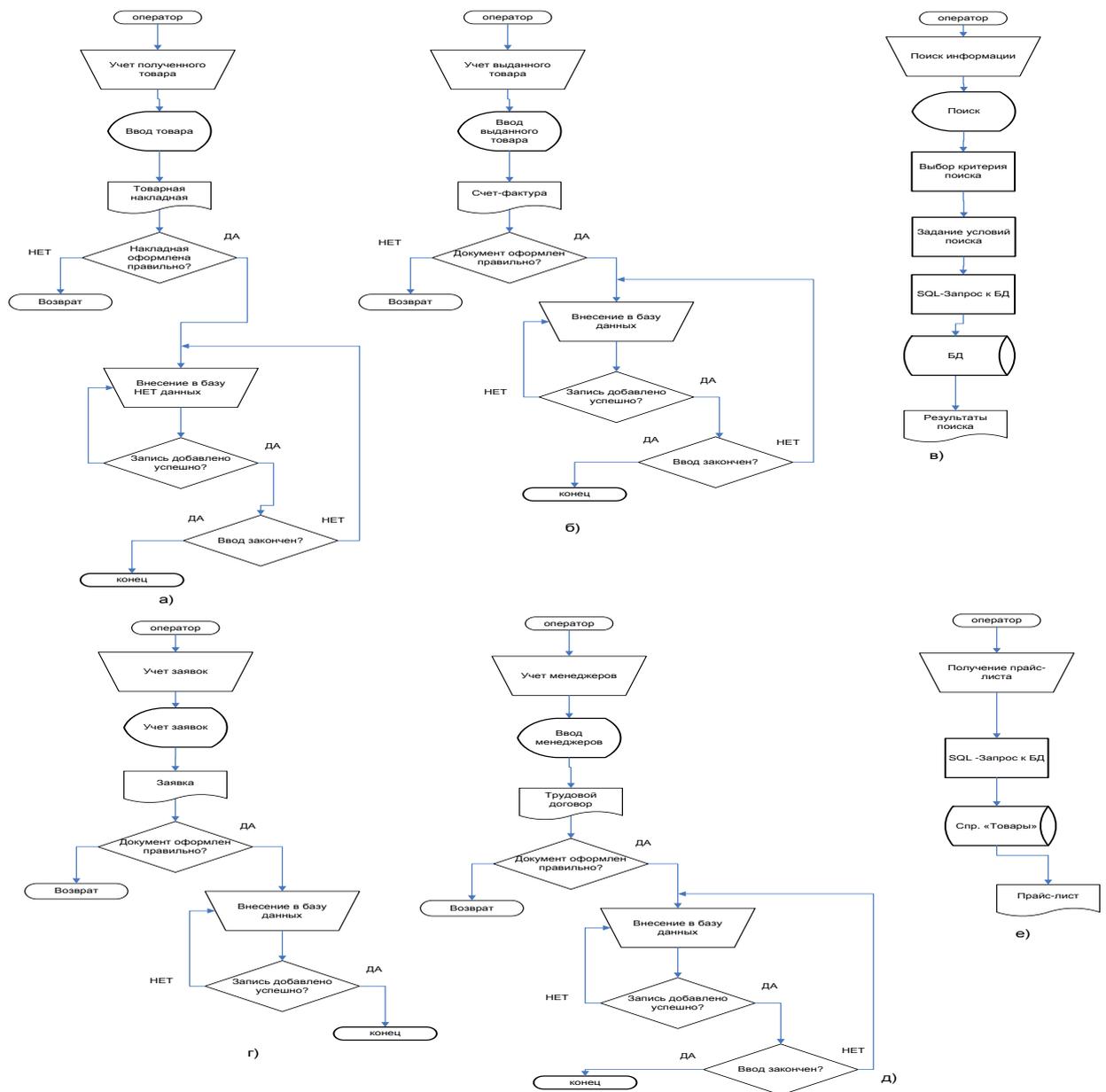


Рисунок 2.6 – Схема технологических процессов: а) учет полученного товара, б) учет выданного товара, в) поиск информации, г) учет заявок, д) учет менеджеров, е) получение прайс-листа.

Описанный технологический процесс демонстрирует нагрузку на сотрудников предприятия, а также их функционал и задачи.

2.5 Пример реализации проекта и его описание

Программа предназначена для автоматизации учета работы с документами для ОАО «ВольтМастер».

Информационная система выполняет следующие функции:

- ведение информации в собственной базе данных о:
 - поставщиках;
 - клиентах;
 - товарах;
 - учете движения товара;
 - учете заказов;

Работу с программой осуществляют пользователи следующих категорий:

- «Продавцы»;
- «Менеджеры»;
- «Администраторы».

На рисунках 2.7 – 2.10 поэтапно представлен процесс работы.

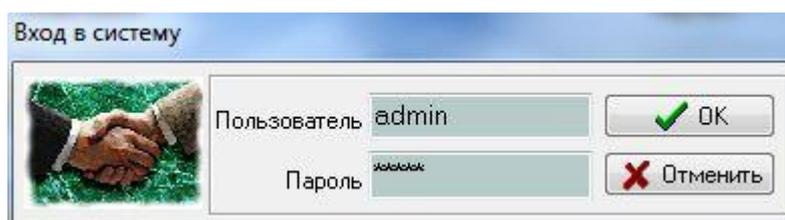


Рисунок 2.7 – Форма Авторизации

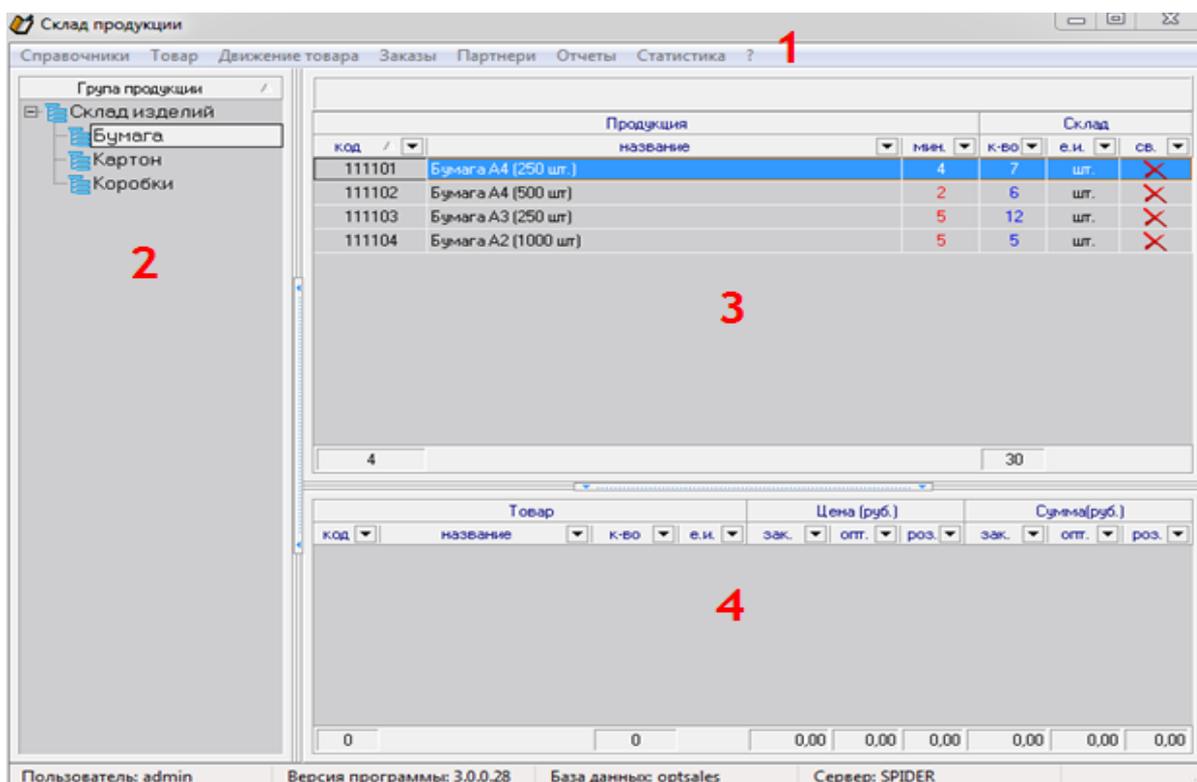


Рисунок 2.8 – Главная форма автоматизированной системы

Элементы главной формы:

1. Меню программы
2. Панель группы товаров
3. Список товаров
4. Рабочая панель

В левой панели системы (2) отображен список складов и групп товаров на складе. В правой отображается товар (3), находящийся на складе.

В меню «Справочники» находится перечень всех справочников системы:

- поставщики;
- данные о предприятии;
- клиенты;
- единицы измерения;
- ответственные лица;
- заказчики;
- причины возврата товара.

Вкладка меню «Товар»:

- добавить группу товаров;
- удалить группу товаров;
- добавить товар;
- изменить данные о товаре;
- переместить товар;
- поиск товара;
- возврат товара.

«Поле ввода текста. Предназначено для ввода строк неформатированного текста. Текст можно ввести с помощью клавиатуры или скопировать из буфера обмена. Следует нажать кнопкой мыши на это поле, после чего при помощи клавиатуры ввести новый или исправить старый текст. Клавишами *Del* и *Backspace* можно удалять ненужный текст» [10].

 A screenshot of a software interface showing a form with four text input fields. The fields are labeled: 'Название' (Name) with the value 'Поставщик №', 'Адрес' (Address), 'Скидка (%)' (Discount (%)), and 'Примечание' (Remarks). The 'Примечание' field has a small upward-pointing arrow on its right side, indicating it is a scrollable area.

Рисунок 2.9 – Поле ввода текста

«Выпадающие списки. Предназначены для выбора значения из фиксированного списка значений. Выпадающие списки с редактированием позволяют добавлять новые значения. При вводе первых символов в поле списка осуществляется переход к соответствующему значению» [10].

 A screenshot of a dropdown menu. The label 'Единицы измерения' (Units of measurement) is on the left. The dropdown list is open, showing four options: 'Коробки' (Boxes), 'Упаковки' (Packaging), 'Штуки' (Pieces), and 'Ящики' (Cases). The 'Коробки' option is currently selected and highlighted in blue.

Рисунок 2.10 – Выпадающие списки

«Поле ввода даты. Предназначено для указания даты. Используйте выпадающие списки для определения числа, месяца и года.

При нажатии на кнопку  открывается специальное окно, в котором необходимо выбрать нужное значение. Выбор дня производится после нажатия кнопкой мыши на соответствующем элементе календаря. Для изменения значений года и месяца используйте кнопки «» и «» [10].

2.5.1 Работа со списками

«Система позволяет регулировать ширину столбцов в списках. Сделанные изменения запоминаются и восстанавливаются при следующем сеансе работы. Изменение ширины столбца производится следующим образом. Подведите указатель мыши к правой границе требуемого столбца так, чтобы указатель изменил свою форму. После того как указатель изменит свою форму, нажмите левую кнопку мыши и, не отпуская её, передвиньте указатель в нужную сторону. При этом та граница, по которой нажата кнопка мыши, будет двигаться вслед за перемещениями указателя. После того как получен нужный размер столбца, отпустите кнопку мыши» [10].

Формирование полученных информационных списков делается с учетом пользовательских требований. Так, в частности, предусмотрена возможность сортировки в алфавитном порядке или по возрастанию/убыванию. Направление упорядочивания списка указывается в заголовке специальным значком . Повторное нажатие по заголовку меняет направление сортировки на противоположное - .

В столбцах, содержащих символьную или текстовую информацию, данные упорядочиваются в прямом или обратном алфавитном порядке, а в столбцах, содержащих целочисленные данные или дату, по возрастанию или убыванию.

2.5.2 Работа с товаром

На рисунке 2.11 изображена форма добавления товара.

Новый товар

Сокращенное название

Полное название

Код товара

Внутренний код товара

Минимальное количество: 0

Единицы измерения: Коробки

Согласен Отменить

Рисунок 2.11 – Форма добавления товара

На рисунке 2.12 изображена форма движения товара

Вкладка меню «Движение товара» содержит вкладки:

- Получить товар;
- Отпустить товар;

Товар покупателю

Товар

Бумага А4 (250 шт.) -/з.ц./: 500.0000 -/о.ц./: 585.0000 -/р.ц./: 800.0000

1

Бумага А4 (250 шт.)

Оптовая цена: 585

Розничная цена: 800

Упаковка: шт.

В упаковке: 1

Остаток на складе: 7

Реализовать покупателю: 0

Согласен Отмена

Рисунок 2.12 – Форма движения товара

На рисунке 2.13. показаны отчеты

Вкладка меню «Партнеры» содержит вкладки:

- Данные о поставщиках;
- Данные о клиентах (отпущенных товарах);
- Формирование отчета об отгрузке товара за период.

Приход товара					
№ п/п	Поставщик	Получено	Сумма (зак.)	Сумма (опт.)	Сумма (роз.)
1	Бумага	43	23460,00	27448,20	37536,00
	Всего	43	23460,00	27448,20	37536,00
	Плановый доход	14076,00			

Данные о продажах по видам прихода		
№ п/п	Название	Сумма
1	по розничным ценам	11 136,00
	Итого	11 136,00

Данные о продажах по группам			
№ п/п	Название	Сумма зак.	Сумма реал.
1	Бумага	6 960,00	11 136,00
	Итого	6 960,00	11 136,00
	Доход	4 176,00	

Ответственный _____

Рисунок 2.13 – Отчет об отгрузке товара

Вкладка меню «Отчеты» содержит вкладки формирования отчетов, показанные на рисунках 2.14 – 2.19.

Прайс:

Предприятие	Назв. банка
Адреса	Номер р/с
Тел./Факс	Эл. адрес

Прайс-лист

Группа товара Код товара	Название товара	Цена (руб.)	
		опт.	розница
Бумага			
111101	Бумага А4 (250 шт.)	585,00	800,00
111102	Бумага А4 (500 шт.)	528,50	720,00
111103	Бумага А3 (250 шт.)	702,00	980,00
111104	Бумага А2 (1000 шт.)	725,40	992,00

Рисунок 2.14 – Прайс-лист

Наличие на складе:

Наличие на складе

Код товара	Название товара	Кол-во
<i>Бумага</i>		
111101	Бумага А4 (250 шт.)	7
111102	Бумага А4 (500 шт)	6
111103	Бумага А3 (250 шт)	12
111104	Бумага А2 (1000 шт)	5

Рисунок 2.15 – Наличие на складе

Остаток товара по поставщику:

Остаток по товару
поставщик **Деревообрабатывающий комбинат №1**

№ п/п	Код товара	Принято	Продано	Остаток	Цена	Сумма
1	111101	10	3	7	500,00	3500,00
2	111102	10	4	6	450,00	2700,00
3	111103	15	3	12	600,00	7200,00
4	111104	8	3	5	620,00	3100,00
Итого		43	13	30		16500,00

Рисунок 2.16 – Остаток товара

Чек:

Чек №		000001				
Кому выдан		Магазин №1				
№ п/п	Код	Название	Ед. изм.	К-во	Цена за единицу	Сумма
1	111104	Бумага А2 (1000 шт)	шт.	3	992,00	2 976,00
2	111103	Бумага А3 (250 шт)	шт.	3	960,00	2 880,00
3	111102	Бумага А4 (500 шт)	шт.	4	720,00	2 880,00
4	111101	Бумага А4 (250 шт.)	шт.	3	800,00	2 400,00
Итого						11 136,00
Скидка						0,00
К оплате						11 136,00
<i>Сумма прописью одиннадцать тысяч сто тридцать шесть руб. 0 коп.</i>						

Выдал _____

Рисунок 2.17 – Чек на перемещение товара

Приходная накладная:

Накладная № 000001

№ п/п	Код	Название	Ед. изм.	К-во	Цена за единицу			Сумма(зак.)
					зак.	опт.	роз.	
1	111104	Бумага А2 (1000 шт)	шт.	8	620,00	725,40	992,00	4 960,00
2	111103	Бумага А3 (250 шт)	шт.	15	600,00	702,00	960,00	9 000,00
3	111101	Бумага А4 (250 шт.)	шт.	10	500,00	585,00	800,00	5 000,00
4	111102	Бумага А4 (500 шт)	шт.	10	450,00	526,50	720,00	4 500,00
					Всего			23 460,00

Рисунок 2.18 – Приходная накладная на склад

Аналитика движения товара:



Рисунок 2.19 – Аналитика движения товара

На рисунке 2.19 представлена динамика движение товара за месяцы.

В данном пункте был рассмотрен функционал и основные формы автоматизированной системы.

Выводы по главе 2.

В данной главе была разработана информационная система для автоматизации учета. Данный процесс на рассматриваемом предприятии наиболее сложен в связи с большим ассортиментом продукции, а также ее малыми (как правило) размерами, что накладывает свои требования к разрабатываемой системе.

Система, разработанная в результате данной выпускной квалификационной работы, позволяет автоматизировать процесс учета документов, а именно: поступление товара на предприятие, учет наличия товаров на складе, что существенно повышает производительность труда работников. Система обладает возможностью использования в многопользовательском режиме.

Потребность в получении актуальной информации и создание автоматизированной системы учета — первейшая задача автоматизации торгового предприятия. Информационная система позволяет в реальном времени следить за наличие или отсутствием товаров на складе, распределение и резервирование их у покупателей. Важной составляющей автоматизированной системы является возможность получения данных о качественных показателях работы персонала.

Глава 3 ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА

3.1 Выбор и обоснование методики расчёта экономической эффективности

Качественная оценка эффективности автоматизированной системы позволяет получить объективные данные о результатах внедрения системы.

Среди этапов оценки экономической эффективности информационной системы можно выделить:

- оценка по результатам соотношения затрат и результатов;
- расчет стоимости использования автоматизированной системы;
- инвестиционная оценка внедрения системы;

«Показатели коммерческой эффективности проекта в целом отражают финансовые последствия внедрения информационной системы. В качестве основных показателей для расчета коммерческой эффективности проекта рекомендуется использовать следующие:

- чистый доход;
- чистый дисконтированный доход;
- внутренняя норма доходности;
- индексы доходности затрат и инвестиций;
- срок окупаемости.

Таким образом, исходя из всего выше сказанного, можно сделать вывод, что процесс оценки экономической эффективности информационных систем сложен и неоднозначен. Подходить следует индивидуально в каждом конкретном случае, но опираясь на определенные методики, что позволит исключить «человеческий фактор» и снизить погрешности ввиду отсутствия каких-либо данных» [24].

В рассматриваемом случае, так как система не планируется к продаже, а будет внедрена только на одном предприятии, необходимо рассчитать экономическую эффективность исходя из снижения издержек на производственную деятельность.

При расчете экономической эффективности проводится сравнение результатов обработки информации при существующем бизнес-процессе и после внедрения автоматизированной системы. При оценке прямой эффективности рассчитываются две группы показателей – показатель снижения трудовых затрат на предприятии и показатель снижения стоимостных затрат.

При расчете изменения трудовых затрат на обработку информации используется следующая система показателей:

1. Абсолютный показатель снижения трудовых затрат на обработку информации

$$\Delta T = T_0 - T_1 \quad (3.1)$$

где T_0 – годовая трудоемкость обработки информации при базисном варианте;

T_1 – годовая стоимость обработки информации при проектируемом варианте.

2. Коэффициент снижения трудовых затрат

$$K_T = (\Delta T / T_0) * 100 (\%) \quad (3.2)$$

3. Индекс снижения трудовых затрат, который показывает рост производительности труда при обработке информации.

$$Y_T = T_0 / T_{1в} \quad (3.3)$$

К стоимостным показателям относятся: абсолютное снижение стоимостных затрат (ΔC), коэффициент относительного снижения стоимостных затрат (K_C) индекс снижения стоимостных затрат (Y_C):

1. «Показатель снижения стоимостных затрат

$$\Delta C = C_0 - C_1 \quad (3.4)$$

где C_0 – годовая стоимость обработки информации при базисном варианте;

C_1 – годовая стоимость обработки информации при проектируемом варианте.

2. Коэффициент эффективности по затратам:

$$K_c = (\Delta C / C_0) * 100 (\%) \quad (3.5)$$

3. Индекс изменения стоимостных затрат

$$Y_c = C_0 / C_1 \quad (3.6)$$

Помимо рассмотренных показателей целесообразно также рассчитать срок окупаемости затрат на внедрение проекта машинной обработки информации ($T_{ок}$), рассчитываемые в годах, долях года или в месяцах года:

$$T_{ок} = K_{п} / \Delta C \quad (3.7),$$

где $K_{п}$ - затраты на создание проекта (проектирование и внедрение)» [2].

Вывод: Была рассмотрена экономическая эффективность информационной системы, проведен расчет изменения трудовых затрат, рассчитан срок окупаемости.

3.2 Расчёт показателей экономической эффективности проекта

Стоимость ПК равна 0, так компьютер уже закуплен и используется на рабочем месте заведующего склада. Для наглядного представления данные представлены в таблице 3.1.

Таблица 3.1 - Исходные данные для расчета

Данные по проекту	Количество	Ед.измерения
Подсистемы	1	шт.
Функции	11	шт.
Версии	1	шт.
Запросы на спецификацию (КТР1)	1	запрос
Запросы к серверу базы данных (КТР2)	53	запрос

Сотрудник, ведущий базу данных (КЧ2)	1	чел.
Сотрудник, сопровождающий сервис ИТ (КЧ3)	1	чел.
Запросы в службу поддержки (КТР4)	3	запрос
Сотрудник службы поддержки (КЧ4)	1	чел.
Запросы на получение доступа к файлам (КТР5)	51	запрос
Число потребляемых за период (год) единиц сервиса ИТ (q)	15546	документов
Ставка сотрудника склада	67	руб/час
Время на единичное сопровождения ИТ сервиса (Вр)	24	час

Этапы создания ИТ-проекта:

На стадии разработки технического задания происходит: сбор материалов, анализ объектов автоматизации, определение задач автоматизации, оформление технико-экономического обоснования и технического задания.

На стадии разработки проекта информационной системы рассматривается состав автоматизируемых функций и обеспечивающих подсистем. Проводится разработка и настройка программного обеспечения, разработка структуры и регистрация базы данных (БД), наполнение и сопровождение БД.

Следующий этап – внедрение автоматизированной системы – заключается в технологическом сопровождении и отладке модулей системы, проведении обучения персонала и устранения возникающих в ходе тестирования, а в дальнейшем рабочего процесса ошибок и неполадок.

В ходе оценки экономической эффективности проекта необходимо учитывать наличие страхового взноса, которые с 1 января 2010 года уплачивают организации и индивидуальные предприниматели в России.

Таблица 3.2 - Ставки страховых взносов в 2019 г.

База для начисления страховых взносов	Тариф страхового взноса		
	ПФР, %	ФСС, %	ФМС, %
В пределах установленного лимита базы	22,0	2,9	5,1
Свыше установленного лимита базы	10,0	5,1	0,0

Расчеты проводятся на двух этапах - до внедрения автоматизированной информационной системы и Проектный материал, чтобы наглядно показать, какой наиболее эффективный для работы.

Расчет трудовых затрат по существующему и проектируемому варианту:

1. До внедрения ИС

Материальные затраты.

В материальные затраты на обработку информации до внедрения автоматизированной системы включаются затраты на расходные материалы (канцтовары): скоросшиватели, ручки, карандаши, линейки, ластик, штрих, бумага формата А4. Ежемесячные траты на данную продукцию оцениваются в 350,00 руб.

Картриджи: 1 шт. x 700 руб. = 700,00 руб.;

Итого: 1050 руб. в месяц

Соответственно на год потребуется:

$1050 \times 12 = 12600,00$ руб.

Трудовые затраты.

Количество работников - 1 человек с фондом оплаты труда сотрудника = 19200 руб. в месяц.

Отчисления в ПФ РФ (22% от ЗП) составят 4224,0 руб. в месяц.

Отчисления в ФСС РФ (2,9% от ЗП) составят 556,8 руб. в месяц.

Отчисления в ОМС (5,1% от ЗП) составят 972,2 руб.

Фонд заработной платы за год составляет:

$19200 \times 12 = 230400,00$ руб.

Накладные расходы.

$2500\text{Кв/ч} * 1,50\text{руб.} = 37500,00\text{ руб.}$

Всего затрат на существующий проект.

$\text{Сб} = 230400 + 12600 = 243000,00\text{ руб.}$

Таблица 3.3 - Расчет себестоимости внедрения базового варианта

Статьи затрат	Сумма затрат (руб.)	Удельный вес, %
1. Основная зарплата	19200	29,8
2. Дополнительная зарплата (10 % от основной)	1920	3
3. Отчисления в ПФР (22% от ФОТ)	4224	6,6
4. Отчисления в ФСС РФ (2,9% от ФОТ)	556,8	0,9
5. Отчисления в ФФОМС (5,1% от ФОТ)	972,2	1,5
6. Накладные расходы	37500	58,2
ИТОГО	64373	100

2. Проектный вариант

Материальные затраты.

В материальные затраты на обработку информации включаются следующие расходные материалы:

картриджи 1 шт. x 700 руб. = 700,00 руб.;

бумага 1 шт. x 300 руб. = 300,00 руб.;

Итого: 1000,00 руб. в месяц

Соответственно на год потребуется:

$1000 \times 12 = 12000,00\text{ руб.}$

Трудовые затраты.

Количество работников - 1 человек. ФОТ (Фонд оплаты труда) сотрудника = 3000 руб. в месяц.

Отчисления в ПФ РФ (22% от ЗП) составляет 660,00 руб. в месяц.

Отчисления в ФСС РФ (2,9% от ЗП) составляет 87,00 руб. в месяц.

Отчисления в ОМС (5,1% от ЗП) составляет 153,00 руб.

Фонд заработной платы за год составляет:

$3000 \times 12 = 36000,00$ руб.

Накладные расходы.

$2500 \text{Кв/ч} \times 1,50 \text{руб.} = 37500,00$ руб.

Сопровождение 25000 руб.

Всего затрат.

$Сб = 36000 + 12000 = 48000$ руб.

Таблица 3.4 - Расчет себестоимости внедрения проектного варианта

Статьи затрат	Сумма затрат (руб.)	Удельный вес, %
1. Основная зарплата	3000	4,5
2. Дополнительная зарплата (10 % от основной)	300	0,4
3. Отчисления в ПФР (22% от ФОТ)	660	0,9
4. Отчисления в ФСС РФ (2,9% от ФОТ)	87	0,1
5. Отчисления в ОМС (5,1% от ФОТ)	153	0,2
4. Накладные расходы	62500	93,9
ИТОГО	66700	100

Срок, в который планируется провести внедрение автоматизированной системы проекта рассчитан на 5 лет. При этом можно оценить затраты на амортизацию в 20,00% (100% /5) от средств, вложенных в проект.

Инвестиции разрабатываемого проекта составили:

Суммарная стоимость проекта $= 140000,00$ руб.

Расходы на амортизацию: $140000,00 \times 0,20 = 28000,00$ руб.

Затраты в целом: $Сп = 135000,00$ руб.

Рассчитаем значение стоимостных показателей:

Абсолютное снижение стоимостных затрат (ΔC):

$$\Delta C = C_{\text{б}} - C_{\text{п}} \quad (3.8)$$

$$\Delta C = 230400 - 135000 = 95400,00 \text{ руб.}$$

Коэффициента относительного снижения стоимостных затрат (K_c):

$$K_c = \Delta C / C_{\text{б}} * 100\% \quad (3.9)$$

$$K_c = 95400 / 230400 * 100\% = 41,4\%$$

Индекс изменения стоимостных затрат (Y_c):

$$Y_c = C_{\text{б}} / C_{\text{п}} \quad (3.10)$$

$$Y_c = 230400 / 135000 = 1,70$$

Экономическая эффективность системы - это отношение экономического эффекта, полученного за определённый период времени от вложения капитала, к объёму капиталовложений, обеспечившего внедрение мероприятий и получение экономически выгодного эффекта.

Экономический эффект представляет собой чистый доход (прибыль), т. е. цена минус себестоимость.

Формула экономической эффективности имеет вид:

$$E = \Delta C / K_z = 1/T \quad (3.11)$$

где: ΔC - чистый доход (прибыль), полученный в течение года от эксплуатации внедрённого проекта;

K_z - объём инвестиций, за счёт которого обеспечен доход.

$$E = 230400 / 140000 = 1,64$$

Эффективность капитальных затрат определяется сроком окупаемости (T) дополнительных капитальных затрат на модернизацию информационной системы:

Экономические показатели эффективности внедрения АИС представлены в таблице 3.5.

Таблица 3.5 - Экономические показатели проекта

Наименование показателей эффективности	Результат
Необходимые инвестиции на разработку системы	140000,00 руб.
Годовые затраты при существующем варианте	230400,00 руб.
Годовой экономический эффект	95400,00 руб.
Экономическая эффективность проекта	1,64
Срок окупаемости проекта	0,6 лет

Срок окупаемости определяется, как обратная величина E

$$T = 1/E$$

$$T = 1/1,64 = 0,6$$

В данном пункте был проведен расчет себестоимости внедрения и экономическая эффективность проекта.

Выводы по главе 3.

По итогам работы над третьей главой был проведен анализ экономической эффективности, из которого следует, что разработанная система учета окупается за 0,6 года, что является допустимым сроком для подобных систем.

ЗАКЛЮЧЕНИЕ

В данной работе была разработана информационная система для автоматизации учета работы с документами. Данный процесс на рассматриваемом предприятии наиболее сложен в связи с большим ассортиментом продукции, а также ее малыми (как правило) размерами, что накладывает свои требования к разрабатываемой системе.

Система, разработанная в результате данной выпускной квалификационной работы, позволяет автоматизировать процесс учета, а именно: поступление товара, учет товаров на складе, повышает производительность труда работников, обладает возможностью использования в многопользовательском режиме.

В результате внедрения разработки повысилась скорость выполнения рутинных операций, увеличилась точность предоставления данных учета, значительно выросла эффективность труда сотрудников предприятия.

Среди полученных положительных результатов – контроль за наличием или отсутствием товара, материала на складе или на предприятии, получение актуальных отчетов и ведение документации о передвижении товаров, контроль за качеством работы сотрудников и поставщиков, значительные качественные показатели в экономии времени на стандартные операции.

Однако на внедрении данной системы автоматизации учета предприятия не должна останавливаться. Как вариант, следующими шагами должны стать внедрение штрих-кодовой подсистемы учета товаров либо использование RFID-технологий, интеграция с комплексными системами управления деятельностью предприятия, а также переход на полностью безбумажный документооборот, хотя последняя перспектива в свете сегодняшних реалий выглядит наиболее туманной.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Нормативно-правовые акты

1. ГОСТ 7.05–2008. Библиографическая ссылка. Общие требования и правила составления [Текст]. – введ. 2014–01–01. – 22с.
2. ГОСТ 7.05–2008. Библиографическая ссылка. Общие требования и правила составления [Текст]. – введ. 2014–01–01. – 22с.
3. Стандарт ИСО 12207 [Текст]: Процессы Жизненного Цикла Программных Средств.

Научная и методическая литература

4. SQL. Сборник рецептов. Энтони Молинаро Издательство: [Текст] Символ-Плюс Год: 2019.
5. Аляев, Ю. А. Алгоритмизация и языки программирования Pascal, C++, Visual Basic: учебно-справ. пособие. - М. [Текст]: Финансы и статистика, 2014. - 319 с. - Библиогр.: с. 318-319.
6. Безручко, В. Т. Компьютерный практикум по курсу "Информатика" [Текст]: Работа в Windows XP, Word 2003, Excel 2003, Power Point 2003, Outlook 2003, Promt Family 7.0, Интернет [Комплект] : учеб. пособие . Ч. 1. - 3-е изд., перераб. и доп. - М. : ИД ФОРУМ : ИНФРА-М, 2015. - 367с. + 1 эл. опт диск -CD. - (Высшее образование). - Библиогр.: с. 266.
7. Библиографические базы данных ИНИОН по социальным и гуманитарным наукам ведутся с начала 1980-х гг. [Электронный источник] Общий объем массивов составляет более 3,5 млн записей (данные на 01.01.2016 г.)
8. Болушевский, С. В. Эффективный Интернет. [Текст] - СПб. : Питер, 2015. - 299 с. - (Трюки и эффекты).
9. Гольцман, В. И. MySQL 5.0. - СПб. : Питер, 2014. [Текст] - 253 с. - (Библиотека программиста).
10. Зверев, Г. Н. Теоретическая информатика и её основания: в 2 т. Т. 1. - М. : [Текст] Физматлит, 2017. - 591 с. - Библиогр.: с. 556-564

11. Зрюмов, Е.А. Базы данных для инженеров: Учебное пособие. - Барнаул: [Текст] Изд-во АлтГТУ, 2014. – 131 с.
12. Кудряшов, Б. Д. Теория информации : учеб. пособие для студентов вузов. - СПб. : Питер, 2015. - 314 с. - (Учебник для вузов). [Текст] - Библиогр. : с. 304-307.
13. Лаптев, В. В. С ++. Объектно-ориентированное программирование: учеб. пособие. - СПб : ПИТЕР, 2014. - 457 с. -[Текст] (Учебное пособие). - Библиогр.: с. 453-457.
14. Михеева, Е. В. Информационные технологии в профессиональной деятельности : учеб. пособие. - М. : Проспект, 2018. [Текст] - 448 с. - Библиогр.: с. 441-442.
15. Моделирование систем. Объектно-ориентированный подход: [Текст] Ю. Б. Колесов, Ю. Б. Сениченков — Москва, БХВ-Петербург, 2017 г.- 192 с.
16. Немцова, Т. И. Практикум по информатике [Комплект] : учеб. пособие для студентов сред. спец. образования. Ч. 1 /[Текст] под ред Л. Г. Гагариной. - М. : ИД ФОРУМ : ИНФРА-М, 2018. - 319 с. + 1 эл.опт. диск - CD. - (Профессиональное образование). - Библиогр.: с. 314.
17. Олифер В. Г. Компьютерные сети : принципы, технологии, протоколы : учеб. пособие. - 3-е изд. -[Текст] СПб : ПИТЕР, 2018. - 958 с. - (Учебник для вузов). - Библиогр.: с. 919-921
18. Олифер В.Г., Олифер Н.А. Компьютерные сети. [текст] Принципы, технологии, протоколы: Учебник для вузов.3-е изд. - СПб.: Питер, 2017, - 958 с.: ил.
19. Олифер, В. Г. Сетевые операционные системы : [учеб. пособие для студентов вузов]. - 2-е изд. -[Текст] СПб. : Питер, 2019. - 668 с. - (Учебник для вузов). - Библиогр. : с. 650-651
20. Прикладная информатика: справочник / под ред. В. Н. Волковой, В. Н. Юрьевой. - М. : Финансы и статистика : [Текст] Инфра-М., 2018. - 766 с. - Библиогр.: с. 740-761

21. Рефакторинг SQL-приложений. Стефан Фаро, Паскаль Лерми
Издательство: [Текст] Символ-Плюс Год: 2019.
22. Рунов, А. В. Социальная информатика : учеб. пособие. - М. [Текст]:
КноРус, 2019. - 427 с. - Библиогр.: с. 397-407. - ISBN 978-5-390-00358-9 : 220-
00.
23. Свердлов, С. З. Языки программирования и методы трансляции
[Комплект] : учеб. пособие для студентов вузов. - СПб[Текст] : ПИТЕР, 2017. -
637 с. + 1 эл. опт. диск (CD). - ISBN 978-5-469-00378-6 : 364-10.
24. Сергеева, И. И. Информатика : учеб. для студентов уч-ий ср. проф.
образ. - М. [Текст] : ИД ФОРУМ : ИНФРА-М, 2018. - 335 с. -
(Профессиональное образование). - Библиогр. : с. 295-297. - ISBN 978-5-8199-
0250-9. - ISBN 978-5-16-002545-2 : 134-86.
25. Степанов, А. Н. Информатика : учеб. пособие. - 5-е изд. - СПб. :
Питер, 2018. - 764 с. - (Учебник для вузов). - Библиогр. [Текст] : с. 754. - Алф.
указ.: с. 754-764. - ISBN 978-5-469-01348-8 : 297-90.
26. Стоян, А. Настройка и обслуживание компьютера: самоучитель;
[Текст] Питер, 2017. -340 с.
27. Сухарев, М.В. Основы Delphi. Профессиональный подход, [Текст]
Наука и техника, 2018.-420 с.
28. Фаронов, В. В. Delphi. Программирование на языке высокого
уровня : учебник для студентов вузов. - СПб. [Текст]: Питер, 2014. - 639 с. -
(Учебник для вузов). - Библиогр. : с. 628. - ISBN 978-5-8046-0008-3 : 269-70
29. Фаронов, В. В. Turbo Pascal 7.0 : практика программирования :
учеб. пособие. - М. [Текст]: КНОРУС, 2015. - 415 с. - Библиогр.: с. 413-415. -
ISBN 978-5-390-00165-3 : 154-00.
30. Хорев, П. Б. Методы и средства защиты информации в
компьютерных системах : учеб. пособие для студентов вузов. - 4-е изд., стер. -
М. [Текст] : Академия, 2016. - 255 с. - Библиогр.: с. 251-252. - ISBN 978-5-7695-
5118-5 : 224-40.

31. Шаньгин, В. Ф. Информационная безопасность компьютерных систем и сетей : учеб. пособие для студентов уч-ий ср. проф. образ. - М. [Текст]: ИД ФОРУМ : ИНФРА-М, 2018. - 415 с. - (Профессиональное образование). - Библиогр. : с. 401-408. - ISBN 978-5-8199-0331-5. - ISBN 978-5-16-003132-3 : 169-95.

32. Шапиро, Л. Дж. Компьютерное зрение : учеб. пособие / пер. с англ. А. А. Богуславского, М. М. Соколова. - М. [Текст]: БИНОМ. Лаборатория знаний, 2016. - 752 с. - (Лучший зарубежный учебник). - Библиогр. в конце глав. - ISBN 5-94774-384-1 : 396-00.

ПРИЛОЖЕНИЕ А

Листинг автоматизированной системы учета работы с документами

```
unit Main;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, RzPanel, RzSplit, ExtCtrls, SpeedBar, DB, TMDDataSource,

IBCCustomDataSet, IBQuery, dxDBGrid, dxTL, dxDBCtrI, dxCntner, dxDBTL,

ComCtrls, Menus, Placemnt, ActnList, dxDBTLCl, dxGrClms, IBStoredProc,

RxMemDS, ADODB, FR_Class, FR_Desgn, FR_DSset, FR_DBSet, FR_View, dxBar,

dxBarExtItems, FR_E_CSV,

FR_E_RTf, FR_E_TXT, StdCtrls, Mask, ToolEdit, FR_E_HTM, DBTables;

const

WM_UIS_MESSAGE = 5000;

WM_GDID_MESSAGE = 5001;

flsZero: Boolean = False;

flsReceive: Boolean = False;

flsGive: Boolean = False;

type

TfrmMain = class(TForm)

RzSplitter1: TRzSplitter;

StatusBar: TStatusBar;

RzSplitter2: TRzSplitter;

dsSelGroupGood: TMDDataSource;

ActionList: TActionList;

FormStorage: TFormStorage;

MainMenu: TMainMenu;

N11: TMenuItem;

dsSelGoods: TMDDataSource;

aEnterprise: TAction;

N1: TMenuItem;

pmGroupEdit: TPopupMenu;

pmGoods: TPopupMenu;

pmPaymentIns: TPopupMenu;

aGroupInsert: TAction;

aGroupDelete: TAction;

aGoodsInsert: TAction;

aGoodsDelete: TAction;

aGoodsUpdate: TAction;

aGoodsMove: TAction;
```

aGoodsReceive: TAction;
aGoodsGive: TAction;
aClose: TAction;
N2: TMenuItem;
N3: TMenuItem;
N4: TMenuItem;
aClose1: TMenuItem;
N5: TMenuItem;
N6: TMenuItem;
N7: TMenuItem;
N8: TMenuItem;
N9: TMenuItem;
N10: TMenuItem;
N12: TMenuItem;
N13: TMenuItem;
N14: TMenuItem;
N15: TMenuItem;
N16: TMenuItem;
N17: TMenuItem;
N18: TMenuItem;
N19: TMenuItem;
N20: TMenuItem;
N21: TMenuItem;
N22: TMenuItem;
mdfGoodsOld: TRxMemoryData;
N25: TMenuItem;
mdfGoodsOldSHORTNAME: TStringField;
mdfGoodsOldAMOUNT: TIntegerField;
mdfGoodsOldMEASURES: TStringField;
dsGoods: TTMDDataSource;
mdfGoodsOldGDDID: TIntegerField;
mdfGoodsOldUID: TStringField;
mdfGoodsOldINNERUID: TStringField;
aDelete: TAction;
N27: TMenuItem;
aEmpty: TAction;
N28: TMenuItem;
aLoadGoods: TAction;
N29: TMenuItem;
pmPaymentOut: TPopupMenu;
N30: TMenuItem;
N31: TMenuItem;
aMoveGoods: TAction;
aMoveGoods1: TMenuItem;

mdfGoodsOldGSUMM: TCurrencyField;
mdfGoodsOldGRID: TIntegerField;
qSelGroupGood: TADOQuery;
qSelGroupGoodG_NAME: TStringField;
qSelGroupGoodGRID: TAutoIncField;
qSelGroupGoodPGID: TIntegerField;
spGroupDel: TADOStoredProc;
spDelGoods: TADOStoredProc;
qSelGoods: TADOQuery;
qSelGoodsG_UID: TStringField;
qSelGoodsG_INNERUID: TStringField;
qSelGoodsG_SHORTNAME: TStringField;
qSelGoodsG_FULLNAME: TStringField;
qSelGoodsG_MINAMOUNT: TSmallintField;
qSelGoodsGDID: TAutoIncField;
qSelGoodsM_SHORTNAME: TStringField;
qSelGoodsG_AMOUNT: TSmallintField;
aClientDetail: TAction;
N32: TMenuItem;
N33: TMenuItem;
aSupplierDetail: TAction;
N34: TMenuItem;
N35: TMenuItem;
N36: TMenuItem;
aAbout: TAction;
N37: TMenuItem;
frPrise: TfrReport;
qSelPrise: TADOQuery;
qSelPriseG_NAME: TStringField;
qSelPriseG_UID: TStringField;
qSelPriseG_SHORTNAME: TStringField;
qSelPriseG_FULLNAME: TStringField;
qSelEnterprise: TADOQuery;
qSelEnterpriseEID: TAutoIncField;
qSelEnterpriseE_NAME: TStringField;
qSelEnterpriseE_ADDRESS: TStringField;
qSelEnterpriseE_BANK: TStringField;
qSelEnterpriseE_ACCOUNT: TStringField;
qSelEnterpriseE_ZKPO: TStringField;
qSelEnterpriseE_PHONE: TStringField;
qSelEnterpriseE_FAX: TStringField;
qSelEnterpriseE_EMAIL: TStringField;
aPrise: TAction;
N38: TMenuItem;

frDBdsPrise: TfrDBDataSet;
aOrderForm: TAction;
N39: TMenuItem;
N40: TMenuItem;
frWarehouse: TfrReport;
frDBWarehouse: TfrDBDataSet;
qSelWarehouse: TADOQuery;
aWarehouse: TAction;
N41: TMenuItem;
aSupplier: TAction;
N42: TMenuItem;
aClient: TAction;
N43: TMenuItem;
aMeasures: TAction;
N44: TMenuItem;
qSelAll: TADOQuery;
dsSelAll: TTMDDataSource;
qSelAllG_UID: TStringField;
qSelAllG_INNERUID: TStringField;
qSelAllG_SHORTNAME: TStringField;
qSelAllG_FULLNAME: TStringField;
qSelAllG_MINAMOUNT: TSmallintField;
qSelAllGDID: TAutoIncField;
qSelAllM_SHORTNAME: TStringField;
qSelAllG_AMOUNT: TSmallintField;
aCrucPerson: TAction;
N45: TMenuItem;
aClientVerif: TAction;
aVerifications: TAction;
mdfGoodsOldPPRICE: TCurrencyField;
mdfGoodsOldWSPRICE: TCurrencyField;
mdfGoodsOldRSPRICE: TCurrencyField;
mdfGoodsOldTARA: TStringField;
mdfGoodsOldTAMOUNT: TIntegerField;
aCard: TAction;
N49: TMenuItem;
qSelPriseC_WSPRICE: TBcdfField;
qSelPriseC_RSPRICE: TBcdfField;
mdfGoodsOldCID: TIntegerField;
mdfGoodsOldSUMM: TCurrencyField;
qDecGoods: TADOQuery;
qIncGoods: TADOQuery;
aPrise1: TAction;
N50: TMenuItem;

aDayPayment: TAction;
N46: TMenuItem;
aPrintGrid: TAction;
aRetType: TAction;
N47: TMenuItem;
aReturnGoods: TAction;
aFind: TAction;
aFind1: TMenuItem;
N51: TMenuItem;
qSelWarehouseG_NAME: TStringField;
qSelWarehouseG_UID: TStringField;
qSelWarehouseG_AMOUNT: TIntegerField;
qSelWarehouseG_SHORTNAME: TStringField;
aClSumm: TAction;
qSumOst: TADOQuery;
aSumOst: TAction;
pmSplitter: TPopupMenu;
N52: TMenuItem;
N53: TMenuItem;
qSelSumGood: TADOQuery;
qSelSumGoodg_shortname: TStringField;
qSelSumGoodg_amount: TIntegerField;
qSelSumGoodmaxsum: TBCDField;
aMaxSum: TAction;
qSelSumGoodminsum: TBCDField;
aRetAll: TAction;
N55: TMenuItem;
frTextExport1: TfrTextExport;
frRTFExport1: TfrRTFExport;
frCSVExport1: TfrCSVExport;
aDiscountCard: TAction;
N56: TMenuItem;
aRollUp: TAction;
aStatistics: TAction;
nStat: TMenuItem;
N58: TMenuItem;
aAllStat: TAction;
N59: TMenuItem;
aMaxSales: TAction;
N60: TMenuItem;
qSelGoodsISCHECK: TIntegerField;
qSelAllISCHECK: TIntegerField;
aChecked: TAction;
qIsCheck: TADOQuery;

N61: TMenuItem;
N62: TMenuItem;
aNotChecked: TAction;
N63: TMenuItem;
qNonChecked: TADOQuery;
Panel1: TPanel;
tlGroupGood: TdxDBTreeList;
tlcGroupGood: TdxDBTreeListColumn;
Panel2: TPanel;
dbgDocuments: TdxDBGrid;
cUID: TdxDBGridColumn;
cSHNAME: TdxDBGridColumn;
cMinAmount: TdxDBGridColumn;
cAmount: TdxDBGridColumn;
cMeasures: TdxDBGridColumn;
dbgDocumentsColumn7: TdxDBGridImageColumn;
Panel3: TPanel;
dbgPayment: TdxDBGrid;
dxDBGridColumn1: TdxDBGridColumn;
dxDBGridColumn3: TdxDBGridColumn;
cCount: TdxDBGridColumn;
dxDBGridColumn6: TdxDBGridColumn;
cZak: TdxDBGridCurrencyColumn;
cOpt: TdxDBGridCurrencyColumn;
cRoz: TdxDBGridCurrencyColumn;
cAll: TdxDBGridCurrencyColumn;
N23: TMenuItem;
N24: TMenuItem;
N26: TMenuItem;
N48: TMenuItem;
N54: TMenuItem;
N57: TMenuItem;
N64: TMenuItem;
N65: TMenuItem;
N66: TMenuItem;
aGoodsOrder: TAction;
N67: TMenuItem;
aGoodOrders: TAction;
N68: TMenuItem;
aOrderView: TAction;
N69: TMenuItem;
Panel4: TPanel;
qSumOstczak: TBCDField;
qSumOstcopt: TBCDField;

qSumOstcroz: TBCDField;
qSumOstsam: TIntegerField;
frGGOstat: TfrReport;
frDBGGOstat: TfrDBBDataSet;
spGroupGoodsOstat: TADOStoredProc;
aGGOstat: TAction;
N70: TMenuItem;
frHTMExport1: TfrHTMExport;
aHtml: TAction;
N71: TMenuItem;
aGoodsMovest: TAction;
N72: TMenuItem;
tCardExport: TTable;
qExCard: TADOQuery;
N73: TMenuItem;
N74: TMenuItem;
N75: TMenuItem;
qDelTable: TQuery;
tCardImport: TTable;
qImpCard: TADOQuery;
tCardExportDCID: TIntegerField;
tCardExportCPERCENT: TIntegerField;
tCardExportCMONEY: TCurrencyField;
tCardImportDCID: TIntegerField;
tCardImportCPERCENT: TIntegerField;
tCardImportCMONEY: TCurrencyField;
qDelACD: TADOQuery;
qExCarddcid: TIntegerField;
qExCardpercent: TIntegerField;
qExCardcmoney: TBCDField;
aPererach: TAction;
N76: TMenuItem;
N77: TMenuItem;
N78: TMenuItem;
mdGoods: TTable;
mdGoodsGRID: TIntegerField;
mdGoodsSHORTNAME: TStringField;
mdGoodsAMOUNT: TIntegerField;
mdGoodsMEASURES: TStringField;
mdGoodsGDID: TIntegerField;
mdGoodsUID: TStringField;
mdGoodsINNERUID: TStringField;
mdGoodsPPRICE: TCurrencyField;
mdGoodsWSPRICE: TCurrencyField;

mdGoodsRSPRICE: TCurrencyField;
mdGoodsTARA: TStringField;
mdGoodsTAMOUNT: TIntegerField;
mdGoodsCID: TIntegerField;
mdGoodsGSUMM: TCurrencyField;
mdGoodsSUMM: TCurrencyField;
N79: TMenuItem;
N80: TMenuItem;
spExAmount: TADOStoredProc;
spExAmountg_uid: TStringField;
spExAmountgdate: TDateTimeField;
spExAmountg_shortname: TStringField;
spExAmountsprice: TBCDField;
spExAmountg_amount: TIntegerField;
tExpAmount: TTable;
tExpAmountUID: TStringField;
tExpAmountEDATE: TDateField;
tExpAmountENAME: TStringField;
tExpAmountERSPRICE: TCurrencyField;
tExpAmountAMOUNT: TIntegerField;
spImpAmount: TADOStoredProc;
tImpAmount: TTable;
tImpAmountUID: TStringField;
tImpAmountEDATE: TDateField;
tImpAmountENAME: TStringField;
tImpAmountERSPRICE: TCurrencyField;
tImpAmountAMOUNT: TIntegerField;
qDelExp: TQuery;
aUpdateAmount: TAction;
aAbortPayment: TAction;
N81: TMenuItem;
aWaybillIn: TAction;
N82: TMenuItem;
qDelExpGoods: TADOQuery;
aGoodsOtchet: TAction;
N83: TMenuItem;
aEmpty1: TAction;
mdGoodsRSUMM: TCurrencyField;
cRAI1: TdxDBGridCurrencyColumn;
aSupplier: TAction;
N84: TMenuItem;
mdGoodsZSUMM: TCurrencyField;
cZSumm: TdxDBGridCurrencyColumn;
aUtils: TAction;

```

frRTFExport2: TfrRTFExport;

frCSVExport2: TfrCSVExport;

aSplsh: TAction;

N85: TMenuItem;

aGroupArhive: TAction;

N86: TMenuItem;

N87: TMenuItem;

qGroupInArh: TADOQuery;

qMovest: TADOQuery;

N88: TMenuItem;

Panel5: TPanel;

Memo: TMemo;

qMovestam: TIntegerField;

qMovestdd: TDateTimeField;

qMovestnn: TStringField;

qMovestcn: TStringField;

qMovestpr: TStringField;

procedure FormCreate(Sender: TObject);

procedure dsSelGroupGoodTMDDataChange(Sender: TObject; Field: TField);

procedure aEnterpriseExecute(Sender: TObject);

procedure aCloseExecute(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure aGroupInsertExecute(Sender: TObject);

procedure aGroupDeleteExecute(Sender: TObject);

procedure aGoodsInsertExecute(Sender: TObject);

procedure aGoodsDeleteExecute(Sender: TObject);

procedure ActionListUpdate(Action: TBasicAction; var Handled: Boolean);

procedure aGoodsUpdateExecute(Sender: TObject);

procedure aGoodsMoveExecute(Sender: TObject);

procedure aGoodsReceiveExecute(Sender: TObject);

procedure aGoodsGiveExecute(Sender: TObject);

procedure aDeleteExecute(Sender: TObject);

procedure aEmptyExecute(Sender: TObject);

procedure aLoadGoodsExecute(Sender: TObject);

procedure aMoveGoodsExecute(Sender: TObject);

procedure aClientDetailExecute(Sender: TObject);

procedure aSupplierDetailExecute(Sender: TObject);

procedure aAboutExecute(Sender: TObject);

procedure aPriseExecute(Sender: TObject);

procedure aOrderFormExecute(Sender: TObject);

procedure aWarehouseExecute(Sender: TObject);

procedure aSupplierExecute(Sender: TObject);

procedure aClientExecute(Sender: TObject);

procedure aMeasuresExecute(Sender: TObject);

```

```

procedure aCrucPersonExecute(Sender: TObject);

procedure aClientVerifExecute(Sender: TObject);

procedure aVerificationsExecute(Sender: TObject);

procedure aCardExecute(Sender: TObject);

procedure dbgDocumentsDbfClick(Sender: TObject);

procedure dsSelGoodsTMDDataChange(Sender: TObject; Field: TField);

procedure aPriseIExecute(Sender: TObject);

procedure aDayPaymentExecute(Sender: TObject);

procedure aRetTypeExecute(Sender: TObject);

procedure aReturnGoodsExecute(Sender: TObject);

procedure aFindExecute(Sender: TObject);

procedure aClSummExecute(Sender: TObject);

procedure aSumOstExecute(Sender: TObject);

procedure N52Click(Sender: TObject);

procedure N53Click(Sender: TObject);

procedure aMaxSumExecute(Sender: TObject);

procedure aRetAllExecute(Sender: TObject);

procedure aDiscountCardExecute(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure aRollUpExecute(Sender: TObject);

procedure aStatisticsExecute(Sender: TObject);

procedure aAllStatExecute(Sender: TObject);

procedure aMaxSalesExecute(Sender: TObject);

procedure aChekedExecute(Sender: TObject);

procedure aNotCheckedExecute(Sender: TObject);

procedure aGoodsOrderExecute(Sender: TObject);

procedure DateComboChange(Sender: TObject);

procedure aGoodOrdersExecute(Sender: TObject);

procedure aOrderViewExecute(Sender: TObject);

procedure MainMenuChange(Sender: TObject; Source: TMenuItem;
  Rebuild: Boolean);

procedure aGGOstatExecute(Sender: TObject);

procedure aHtmlExecute(Sender: TObject);

procedure aGoodsMovestExecute(Sender: TObject);

procedure N74Click(Sender: TObject);

procedure N75Click(Sender: TObject);

procedure aPererachExecute(Sender: TObject);

procedure mdGoodsCalcFields(DataSet: TDataSet);

procedure N79Click(Sender: TObject);

procedure N80Click(Sender: TObject);

procedure frGGOstatGetValue(const ParName: String;
  var ParValue: Variant);

procedure aUpdateAmountExecute(Sender: TObject);

procedure aAbortPaymentExecute(Sender: TObject);

```

```

procedure aWaybillInExecute(Sender: TObject);

procedure aGoodsOtchetExecute(Sender: TObject);

procedure aEmpty1Execute(Sender: TObject);

procedure aSupplierExecute(Sender: TObject);

procedure aUtilsExecute(Sender: TObject);

procedure aSplashExecute(Sender: TObject);

procedure aGroupArhiveExecute(Sender: TObject);

procedure N88Click(Sender: TObject);

procedure MemoDbClick(Sender: TObject);

private

{ Private declarations }

public

procedure RefLocEx(DS : TDataSet; const KeyField: string; Value: Variant);

procedure UIS_MESSAGE(var M: TMessage);message WM_UIS_MESSAGE;

procedure GDID_MESSAGE(var M: TMessage);message WM_GDID_MESSAGE;

{ Public declarations }

end;

var

frmMain: TfrmMain;

implementation

uses Data, NewGroup, NewGoods, GoodsMove, IWGoods, OWGoods, InPayment,
OutPayment, ClientDetail, SupplierDetail, Enterprise, About, Order,
Supplier, Client, Measures, GoodsEdit, CrucialPerson, ClientVerif,
Verifications, Card, Global, Math, RetType, ReturnGoods,
Filter, AllReturn, DiscountCard, GoodStatistics, SalesStatistics,
MaxSales, GoodsOrder, GoodOrder, OrderView, GoodsMovest, Pererach,
UpdateAmount, AbortPayment, WaybillIn, Splash, GoodsOtchet,
SupplierOtchet, PaymentDay, Ostatok;

{$R *.dfm}

procedure TfrmMain.FormCreate(Sender: TObject);

begin

with dmData do begin

if ADOConnection.Connected = False then begin

ADOConnection.Connected := True;

end;

end;

with qSelGroupGood do begin

Close;

```

```

Open;

end;

if mdGoods.Active then Close;

mdGoods.Open;

with dbgDocuments do begin

    sdxPanelText := 'Группировкаколоннок';

end;

if qSelGoods.Active = False then qSelGoods.Active := True;

if qSelAll.Active = False then qSelAll.Active := True;

Self.Caption := _Name;

StatusBar.Panels[0].Text := ' Пользователь: '+fCurrentUser;

StatusBar.Panels[1].Text := ' Версияпрограммы: '+GetFileVersion(Application.ExeName);

StatusBar.Panels[2].Text := ' Базаданных: '+dmData.ADOConnection.Properties.Item[0].Value;

StatusBar.Panels[3].Text := ' Сервер: '+dmData.ADOConnection.Properties.Item[15].Value;

end;

procedure TfrmMain.dsSelGroupGoodTMDDataChange(Sender: TObject;
Field: TField);

begin

case qSelGroupGoodPGID.AsInteger of

1: begin

with dbgDocuments do begin

DataSource := dsSelAll;

end;

with qSelAll do begin

Close;

Open;

end;

dbgDocuments.Invalidate;

end;

2..1000: begin

with dbgDocuments do begin

DataSource := dsSelGoods;

end;

with qSelGoods do begin

Close;

with Parameters do begin

ParamByName('GRID').Value := qSelGroupGoodGRID.AsInteger;

end;

Prepared;

Open;

end;

dbgDocuments.Invalidate;

end;

```

```

end;

end;

procedure TfrmMain.aEnterpriseExecute(Sender: TObject);
begin
with TfrmEnterprise.Create(nil) do begin
try
with tEnterprise do begin
if Active then Active := False;
Active := True;
end;
ShowModal;
finally
Free;
end;
end;
end;
end;

procedure TfrmMain.aCloseExecute(Sender: TObject);
begin
Close;
end;

procedure TfrmMain.FormClose(Sender: TObject; var Action: TCloseAction);
begin
if mdGoods.IsEmpty then begin
with dmData.ADOConnection do begin
Connected := False;
end;
Action := caFree;
frmMain := Nil;
end
else
if MessageDlg('Есть непроведенные данные'#13'Операция отменяется!!!'#13'Просьба быть внимательнее!',
mError,[mbYes],0) = mrYes then Action := caNone
else if MessageDlg('?',
mError,[mbYes],0) = mrYes then Action := caNone
else Action := caNone;
end;

procedure TfrmMain.aGroupInsertExecute(Sender: TObject);
begin
with TfrmNewGroup.Create(nil) do begin
try

```

```

with qSelGroupParent do begin

    Close;

    Open;

    Locate('GRID', 1, [loPartialKey]);

end;

fHandle:=Self.Handle;

fCtrls := 0;

lcbGroupParent.DisplayValue := qSelGroupParentG_NAME.AsString;

ShowModal;

finally

    Free;

end;

end;

end;

procedure TfrmMain.aGroupDeleteExecute(Sender: TObject);

begin

if MessageDlg('Вы действительно желаете удалить'+#13+#13+'запись'+#13, mtInformation, [mbYes, mbCancel], 0) = mrYes then begin

    with spGroupDel do begin

        try

            with Parameters do begin

                ParamByName('@GRID').Value := qSelGroupGoodGRID.AsInteger;

            end;

            Prepared;

            ExecProc;

        except

            on e: Exception do

                begin

                    MessageDlg('Ошибкаудаления'+#13+#13+e.Message,

                        mtError, [mbOk], 0);

                end;

            end;

        end;

    end;

    with qSelGroupGood do begin

        Close;

        Open;

        Locate('GRID',qSelGroupGoodGRID.AsInteger,[loPartialKey]);

    end;

end;

end;

procedure TfrmMain.UIS_MESSAGE(var M: TMessage);

begin

```

```
RefLocEx(qSelGroupGood.'GRID',PGroupInfo(M.LParam)^.GRID);
```

```
end;
```

```
procedure TfrmMain.RefLocEx(DS: TDataSet; const KeyField: string;
```

```
Value: Variant);
```

```
begin
```

```
Application.ProcessMessages;
```

```
try
```

```
if DS.State=dsInactive then
```

```
begin
```

```
DS.Open;
```

```
if Value<>0 then
```

```
DS.Locate(KeyField,Value,[]);
```

```
end
```

```
else
```

```
begin
```

```
if (Value=0) then
```

```
Value:=DS.FieldName(KeyField).Value;
```

```
DS.Close;
```

```
DS.Open;
```

```
if not VarIsEmpty(Value) and not VarIsNull(Value) then
```

```
DS.Locate(KeyField,Value,[]);
```

```
end;
```

```
finally
```

```
end;
```

```
end;
```

```
procedure TfrmMain.aGoodsInsertExecute(Sender: TObject);
```

```
begin
```

```
with TfrmNewGoods.Create(nil) do begin
```

```
try
```

```
with qSelMeasures do begin
```

```
Close;
```

```
Open;
```

```
Locate('MID', qSelMeasures.MID.AsInteger, [loPartialKey]);
```

```
end;
```

```
fHandle:=Self.Handle;
```

```
fCtrls := 0;
```

```
fGRID := qSelGroupGoodGRID.AsInteger;
```

```
IcbMeasures.DisplayValue := qSelMeasuresM_FULLNAME.AsString;
```

```
ShowModal;
```

```
finally
```

```
Free;
```

```
end;
```

```

end;

end;

procedure TfrmMain.aGoodsDeleteExecute(Sender: TObject);

begin

if MessageDlg('Вы действительно желаете удалить'+#13+#13+'запись'+#13, mInformation, [mbYes, mbCancel], 0) = mrYes then begin

with spDelGoods do begin

try

with Parameters do begin

ParamByName('@GDID').Value := qSelGoodsGDID.AsInteger;

end;

Prepared;

ExecProc;

except

on e: Exception do

begin

MessageDlg('Ошибка удаления'+#13+#13+e.Message,

mtError, [mbOk], 0);

end;

end;

end;

end;

with qSelGoods do begin

Close;

Open;

Locate('GDID',qSelGoodsGDID.AsInteger,[loPartialKey]);

end;

end;

end;

procedure TfrmMain.ActionListUpdate(Action: TBasicAction;

var Handled: Boolean);

begin

if dbgDocuments.DataSource.DataSet.IsEmpty then begin

aGoodsDelete.Enabled := False;

aGoodsUpdate.Enabled := False;

aGoodsMove.Enabled := False;

aCard.Enabled := False;

aGoodsReceive.Enabled := False;

aGoodsGive.Enabled := False;

end

else begin

aGoodsDelete.Enabled := True;

aGoodsUpdate.Enabled := True;

```

```

aGoodsMove.Enabled := True;

aCard.Enabled := True;

aGoodsReceive.Enabled := True;

aGoodsGive.Enabled := True;

end;

if (cAmount.Field.Value = 0) or (fUser = 'men410236') then begin

aGoodsGive.Enabled := False;

end

else begin

aGoodsGive.Enabled := True;

end;

if tlGroupGood.DataSource.DataSet.IsEmpty then begin

aGroupDelete.Enabled := False;

aGoodsInsert.Enabled := False;

end

else begin

aGroupDelete.Enabled := True;

aGoodsInsert.Enabled := True;

end;

aStatistics.Enabled := not qSelGoods.IsEmpty;

aNotChecked.Enabled := (fNo = 1);

//nStat.Enabled := (fNo = 1);

aMoveGoods.Enabled := aGoodsGive.Enabled;

aUtils.Enabled := (fUser = 'adminnimda');

end;

procedure TfrmMain.aGoodsUpdateExecute(Sender: TObject);

begin

with TfrmGoodsEdit.Create(nil) do begin

try

with tGoodsEdit do begin

if Active then Active := False;

Active := True;

end;

tGoodsEdit.Locate('GDID', qSelGoods.GDID.AsInteger, [loPartialKey]);

fHandle := Self.Handle;

fCtrls := 0;

fGRID := qSelGroupGood.GRID.AsInteger;

ShowModal;

finally

Free;

end;

end;

end;

end;

```

```

procedure TfrmMain.aGoodsMoveExecute(Sender: TObject);
begin
    with TfrmGoodsMove.Create(nil) do begin
        try
            with qSelGroup do begin
                Close;

                with Parameters do begin
                    ParamByName('GRID').Value := qSelGroupGoodGRID.AsInteger;
                end;

                Open;

                Locate('GRID', qSelGroupGRID.AsInteger, [loPartialKey]);
            end;

            fHandle:=Self.Handle;

            fCtrls := 0;

            fGDID := qSelGoodsGDID.AsInteger;

            lcbGoodsMove.DisplayValue := qSelGroupG_NAME.AsString;

            ShowModal;
        finally
            Free;
        end;
    end;
end;
end;
end;

```

```

procedure TfrmMain.aGoodsReceiveExecute(Sender: TObject);
begin
    with TfrmIWGoods.Create(nil) do begin
        try
            case qSelGroupGoodPGID.AsInteger of
                1: begin
                    fGDID := qSelAllGDID.AsInteger;
                    fUID := qSelAllG_UID.AsString;
                    fIUID := qSelAllG_INNERUID.AsString;
                    fName := qSelAllG_SHORTNAME.AsString;
                    fMName := qSelAllM_SHORTNAME.AsString;
                    fGRID := qSelGroupGoodGRID.AsInteger;

                    with qSelMeasures do begin
                        Close;

                        Open;

                        Locate('MID', qSelMeasuresMID.AsInteger, [loPartialKey]);
                    end;

                    lcbTara.DisplayValue := qSelMeasuresM_FULLNAME.AsString;
                end;
            end;
        finally
            Free;
        end;
    end;
end;
end;
end;

```

```

2..1000: begin

    fGDID := qSelGoodsGDID.AsInteger;

    fUID := qSelGoodsG_UID.AsString;

    fIUID := qSelGoodsG_INNERUID.AsString;

    fName := qSelGoodsG_SHORTNAME.AsString;

    fMName := qSelGoodsM_SHORTNAME.AsString;

    fGRID := qSelGroupGoodGRID.AsInteger;

    with qSelMeasures do begin

        Close;

        Open;

        Locate('MID', qSelMeasuresMID.AsInteger, [loPartialKey]);

    end;

    lcbTara.DisplayValue := qSelMeasuresM_FULLNAME.AsString;

end;

end;

ShowModal;

finally

    Free;

end;

end;

end;

end;

```

```

procedure TfrmMain.aGoodsGiveExecute(Sender: TObject);

```

```

begin

    with TfrmOWGoods.Create(nil) do begin

        try

            case qSelGroupGoodPGID.AsInteger of

                1: begin

                    fGDID := qSelAllGDID.AsInteger;

                    fUID := qSelAllG_UID.AsString;

                    fIUID := qSelAllG_INNERUID.AsString;

                    fName := qSelAllG_SHORTNAME.AsString;

                    fMName := qSelAllM_SHORTNAME.AsString;

                    fGRID := qSelGroupGoodGRID.AsInteger;

                    fHandle := Self.Handle;

                    with qSelSGoods do begin

                        Close;

                        with Parameters do begin

                            ParamByName('gdid').Value := qSelAllGDID.AsInteger;

                        end;

                        Prepared;

                        Open;

                        Locate('cid',qSelSGoodscid.AsInteger,[loPartialKey]);

                    end;

                end;
            end;
        end;
    end;
end;

```

```

GroupBox1.Caption := qSelAllG_SHORTNAME.AsString;

end;

2..1000: begin

fGDID := qSelGoodsGDID.AsInteger;

fUID := qSelGoodsG_UID.AsString;

fIUID := qSelGoodsG_INNERUID.AsString;

fName := qSelGoodsG_SHORTNAME.AsString;

fMName := qSelGoodsM_SHORTNAME.AsString;

fGRID := qSelGroupGoodGRID.AsInteger;

fHandle := Self.Handle;

with qSelSGoods do begin

Close;

with Parameters do begin

ParamByName('gdid').Value := qSelGoodsGDID.AsInteger;

end;

Prepared;

Open;

Locate('cid',qSelSGoodscid.AsInteger,[loPartialKey]);

end;

GroupBox1.Caption := qSelGoodsG_SHORTNAME.AsString;

end;

end;

ShowModal;

finally

Free;

end;

end;

end;

end;

procedure TfrmMain.aDeleteExecute(Sender: TObject);

begin

IncCount(mdGoodsAMOUNT.AsInteger,mdGoodsGDID.AsInteger, mdGoodsCID.AsInteger);

if mdGoods.State = dsBrowse then mdGoods.Edit;

with mdGoods do begin

Delete;

Refresh;

end;

if dbgPayment.DataSource.DataSet.RecordCount <= 0 then begin

dbgPayment.PopupMenu := nil;

end;

Caption := _Name;

end;

end;

```

```

procedure TfrmMain.aEmptyExecute(Sender: TObject);

begin
//
if MessageDlg('Вы действительно желаете очистить список?',
mtWarning,[mbYes],0) = mrYes then begin

with mdGoods do begin

First;

while not Eof do begin

IncCount(mdGoodsAMOUNT.AsInteger,mdGoodsGDID.AsInteger, mdGoodsCID.AsInteger);

Next;

end;

end;

if mdGoods.State = dsBrowse then mdGoods.Edit;

with mdGoods do begin

Close;

EmptyTable;

Open;

Refresh;

end;

dbgPayment.PopupMenu := nil;

dbgPayment.Invalidate;

Caption := _Name;

end

else Exit;

end;

```

```

procedure TfrmMain.aLoadGoodsExecute(Sender: TObject);

begin

with TfrmInPayment.Create(nil) do begin

try

delIW.Date := Now;

with qSelSupplier do begin

Close;

Open;

Locate('SID', qSelSupplierSID.AsInteger,[loPartialKey]);

end;

lcbSupplier.DisplayValue := qSelSupplierS_NAME.AsString;

with qSelCrPerson do begin

Close;

Open;

Locate('CPID', qSelCrPersonCPID.AsInteger,[loPartialKey]);

end;

lcbCrucialPerson.DisplayValue := qSelCrPersonCP_NAME.AsString;

```

```

fHandle:=Self.Handle;

fCtrls := 0;

fGRID := qSelGroupGoodGRID.AsInteger;

ShowModal;

finally

Free;

end;

end;

end;

procedure TfrmMain.aMoveGoodsExecute(Sender: TObject);

begin

with TfrmOutPaiment.Create(nil) do begin

try

deIW.Date := Now;

deDDate.Date := Now;

case fNo of

1: begin

with qSelSupplier do begin

Close;

Open;

Locate('CLID', qSelSupplierCLID.AsInteger,[loPartialKey]);

end;

lcbSupplier.DisplayValue := qSelSupplierC_NAME.AsString;

end;

2: begin

if fUser = 'prodavet235689' then begin

with qSelSupplier do begin

Close;

Filter := '(CLID <> 6) and (CLID <> 9) and (CLID <> 10) and (CLID <> 33) and (CLID <> 34)';

Filtered := True;

Open;

Locate('CLID', qSelSupplierCLID.AsInteger,[loPartialKey]);

end;

lcbSupplier.DisplayValue := qSelSupplierC_NAME.AsString;

end;

if fUser = 'nat1972' then begin

with qSelSupplier do begin

Close;

Filter := '(CLID = 6) or (CLID = 9) or (CLID = 10) or (CLID = 33) or (CLID = 34)';

Filtered := True;

Open;

Locate('CLID', qSelSupplierCLID.AsInteger,[loPartialKey]);

end;

```

```

lcbSupplier.DisplayValue := qSelSupplierC_NAME.AsString;

cbDiscountCard.Enabled := False;

end;

end;

end;

with qSelCrPerson do begin

Close;

Open;

Locate('CPID', qSelCrPersonCPID.AsInteger, [loPartialKey]);

end;

lcbCrucialPerson.DisplayValue := qSelCrPersonCP_NAME.AsString;

fHandle := Self.Handle;

fCtrls := 0;

fGRID := qSelGroupGoodGRID.AsInteger;

fOptSum := cRAAll.SummaryFooterValue;

fRozSum := cAll.SummaryFooterValue;

ShowModal;

finally

Free;

end;

end;

end;

procedure TfrmMain.aClientDetailExecute(Sender: TObject);

begin

with TfrmClientDetail.Create(nil) do begin

try

with qSelClient do begin

Close;

Open;

end;

ShowModal;

finally

Free;

end;

end;

end;

end;

procedure TfrmMain.aSupplierDetailExecute(Sender: TObject);

begin

with TfrmSupplierDetail.Create(nil) do begin

try

with qSelSupplier do begin

Close;

```

```

    Open;

end;

ShowModal;

finally

    Free;

end;

end;

end;

end;

procedure TfrmMain.aAboutExecute(Sender: TObject);

begin

    with TAboutDlg.Create(nil) do begin

        try

            ShowModal;

        finally

            Free;

        end;

    end;

end;

end;

procedure TfrmMain.aPriseExecute(Sender: TObject);

begin

    with qSelPrise do begin

        Close;

        Open;

    end;

    with qSelEnterprise do begin

        Close;

        Open;

    end;

    with frPrise do begin

        LoadFromFile('c:\optsales\report\prise.frf');

        ShowReport;

    end;

end;

end;

procedure TfrmMain.aOrderFormExecute(Sender: TObject);

begin

    with TfrmOrder.Create(nil) do begin

        try

            with qSelSupplier do begin

                Close;

                Open;

                Locate('sid', qSelSupplierSID.AsInteger, [loPartialKey]);

```

```

end;

lchSupplier.DisplayValue := qSelSupplierS_NAME.AsString;

ShowModal;

finally

Free;

end;

end;

end;

end;

```

```

procedure TfrmMain.aWarehouseExecute(Sender: TObject);

begin

with qSelWarehouse do begin

Close;

Open;

end;

with frWarehouse do begin

LoadFromFile('c:\optsales\report\warehouse.frf');

ShowReport;

end;

end;

```

```

procedure TfrmMain.aSupplierExecute(Sender: TObject);

begin

with TfrmSupplier.Create(nil) do begin

try

with tSupplier do begin

if Active then Active := False;

Active := True;

end;

ShowModal;

finally

Free;

end;

end;

end;

```

```

procedure TfrmMain.aClientExecute(Sender: TObject);

begin

with TfrmClient.Create(nil) do begin

try

with tClient do begin

if Active then Active := False;

Active := True;

end;

end;

```

```

    ShowModal;

finally

    Free;

end;

end;

end;

procedure TfrmMain.aMeasuresExecute(Sender: TObject);

begin

with TfrmMeasures.Create(nil) do begin

    try

with tMeasures do begin

    if Active then Active := False;

    Active := True;

end;

    ShowModal;

finally

    Free;

end;

end;

end;

end;

procedure TfrmMain.aCrucPersonExecute(Sender: TObject);

begin

with TfrmCrucialPerson.Create(nil) do begin

    try

with tCrucialPerson do begin

    if Active then Active := False;

    Active := True;

end;

    ShowModal;

finally

    Free;

end;

end;

end;

end;

procedure TfrmMain.aClientVerifExecute(Sender: TObject);

begin

with TfrmClientVerif.Create(nil) do begin

    try

with qSelClient do begin

    Close;

    Open;

end;

end;

end;

end;

```

```
Locate('CLID', qSelClientCLID.AsInteger, [loPartialKey]);  
  
end;  
  
lcbClient.DisplayValue := qSelClientC_NAME.AsString;  
  
ShowModal;  
  
finally  
  
Free;  
  
end;  
  
end;  
  
end;
```

```
procedure TfrmMain.aVerificationsExecute(Sender: TObject);
```

```
begin  
  
with TfrmVerifications.Create(nil) do begin  
  
try  
  
with qSelClient do begin  
  
Close;  
  
Open;  
  
Locate('CLID', qSelClientCLID.AsInteger, [loPartialKey]);  
  
end;  
  
ShowModal;  
  
finally  
  
Free;  
  
end;  
  
end;
```