ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК

Кафедра прикладной информатики и информационных технологий

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЁТА ТОВАРОВ МАГАЗИНА «МЕЛИССА»

Выпускная квалификационная работа

Студента очной формы обучения
направления подготовки 09.03.03 Прикладная информатика
4 курса группы 07001205
Демченко Евгения Николаевича

Научный руководитель:
к.т.н., доцент
Павлов В. Ф.

Рецензент:
к.т.н., доцент
Михелёв В. М.

БЕЛГОРОД 2016

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В современных требованиях, предъявляемых к качеству работы торгово-розничных предприятий, отмечается, что эффективная работа сотрудников таких предприятий полностью зависит от оснащения их информационными средствами и их успешного использования в системе автоматизированного учёта товаров.

Компьютерный учёт товара полностью отличается от классического, рукописного. Компьютерные программы упрощают учёт товаров, сокращают время, требуемое на оформление документов для анализа торговой деятельности, следовательно, при применении компьютерных программ, повышается эффективность работы персонала торгового предприятия, уменьшается время обучения персонала.

Результаты выполнения торговых операций заносятся в соответствующие журналы, что позволяет автоматически их сохранять и использовать в дальнейшем.

Основные преимущества автоматизации учёта: экономия времени, сжатие хранимых данных с экономией объёма памяти и снижение затрат на операции обновления данных. При этом информационная система автоматизирует и ведёт учёт товаров, поставку и отпуск товаров со склада предприятия. Всё это будет сопутствовать более качественному обслуживанию, повышению результативности работы предприятия, повышению точности учёта и снижению потерь товара.

Актуальность темы данной выпускной квалификационной работы связана с необходимостью автоматизирования процесса учёта товаров в организации, которая занимается розничной торговлей в сфере продуктовых магазинов.

Цель данной выпускной квалификационной работы состоит в разработке информационной системы учёта товаров, которая позволит

повысить производительность труда персонала организации, в виде программного приложения.

Для достижения поставленной цели выпускной квалификационной работы необходимо решить следующие задачи:

1) провести анализ предметной области, в которой требуется применение информационной системы;
2) спроектировать структуру будущей информационной системы;
3) спроектировать структуру базы данных;
4) выполнить программную реализацию информационной системы;
5) тестировать информационную систему;

Объектом исследования выпускной квалификационной работы является учёт товаров в магазине.

Выпускная квалификационная работа содержит введение, 3 главы, заключение, список использованных источников, приложения.

Введение раскрывает актуальность работы, объект, цель и задачи, раскрывает практическую значимость работы

В первой главе исследуется розничная торговля, и определяются требования к разрабатываемой системе. Также проводится анализ деятельности магазина «Мелисса» и уровень его автоматизации.

Во второй главе приводится обоснование проектных решений по техническому, информационному, программному, технологическому обеспечению задачи, а также обоснован выбор программных средств.

В третьей главе представлено информационное обеспечение, которое включает описание новой информационной модели учета на предприятии, используемых классификаторов, характеристику первичных и результатных документов, характеристику базы данных, программное обеспечение, которое содержит дерево функций системы и сценарий диалога, структурную схему пакета и описание программных модулей, и технологическое обеспечение задачи. Также описан контрольный пример реализации проекта.

В заключении подводятся итоги работы, формируются окончательные выводы по разработанной системе, и рассматриваются пути её совершенствования.

Выпускная квалификационная работа состоит из 77 страниц, 35 рисунков, 12 таблиц и приложений.

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Технико-экономическая характеристика предметной области

1.1.1 Характеристика предприятия

Розничная торговля – продажа товара конечному потребителю (частному лицу), является динамично развивающимся сектором экономики России.

Предметной областью является продуктовый магазин «Мелисса» (ИП Кореньков И.Н), зарегистрированный как субъект предпринимательской деятельности 12 марта 2001 года, и расположенный по адресу: Белгород, ул. Некрасова, 10.

Основной вид деятельности - магазин по продаже продовольственных и хоз. товаров. Ассортимент магазина насчитывает до 1000 наименований товаров, из которых около 75% составляют продукты.

В своей деятельности магазин руководствуется Уставом и законодательством Российской Федерации.

ИП Кореньков И.Н. является успешно развивающейся организацией, которая ведёт свою деятельность в сфере розничной торговли.

Большое внимание уделяется сотрудничеству с местными производителями и поставщиками, что позволяет оперативно решать вопросы обеспечения магазина свежими товарами местного производства.

Целью работы ИП Кореньков считается получение прибыли, как основной показатель предприятия.

К функциям предприятия ИП Кореньков можно отнести:

1) рекламирование товаров и услуг;
2) оказание торговых услуг покупателям;
3) составление заявок на завоз товаров;
4) формирование ассортимента товаров;

5) изучение покупательского спроса на товары.

Работая для максимального удовлетворения запросов покупателей, организация осуществляет свою деятельность по нескольким основным направлениям, которые ориентированы на розничную продажу.

1.1.2 Организационная структура управления предприятием и её характеристика

В данной организации (ИП Кореньков) используется линейная структура управления, которая является одной из самых простых организаций управленческой структуры.

Во главе любого отдела стоит руководитель, который обладает всеми полномочиями в организации, принятые им решения передаются соответствующим отделам. Руководитель отдела в свою очередь подчиняется руководителю вышестоящего в рамках организационной иерархии.

Организационная структура магазина «Мелисса» ИП Кореньков представлена на рисунке 1:

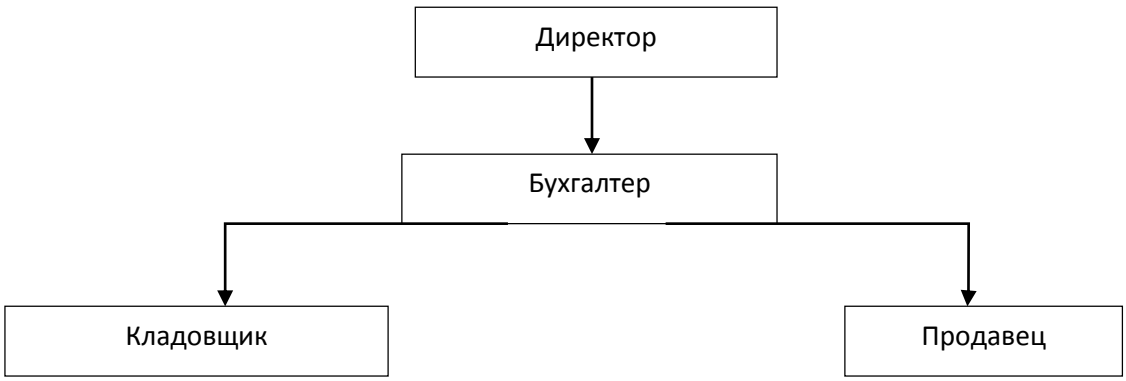


Рисунок 1 – Организационная структура магазина «Мелисса»

Директор руководит организацией в соответствии с законодательством, несёт всю полноту ответственности за все принимаемые решения, сохранность и использование имущества. Директору подчиняется бухгалтер магазина, который следит за приемом и обработкой первичной документации.

В подчинение к бухгалтеру входят кладовщик и продавец. Продавец выполняет операции по взаимодействию с клиентами, а кладовщик следит за товаром, поступающим в магазин.

1.2 Экономическая сущность задачи

Данный комплекс задач как автоматизация учета товаров очень важен для магазина, который занимается розничной торговлей товаров.

Учет товаров – это всегда работа с большим объемом данных. Автоматизация же учета позволяет экономить время, деньги и человеческий ресурс магазина.

Благодаря автоматизации учета товаров заметно снижается количество ошибок, которые делают в процессе работы сотрудники магазина. В автоматизированном магазине продуктивность каждого конкретного исполнителя заметно возрастает. С помощью автоматизации данного процесса можно увеличить количество обслуженных за день посетителей.

Автоматизация устраняет проблемы лишних трудозатрат, экономит время на ручной учет и формирование документации. Товар, который хранится в магазине, сам по себе риск, и чем больше товара, тем больше риск потерпеть убытки. Все зависит от вида товара.

При больших объемах одного вида товара существует риск, что товар потеряет свою актуальность, это приведет либо к потере вложенных средств, либо к получению низкого дохода.

Всех вышеперечисленных проблем можно избежать при регулярном формировании и анализе разнообразных отчетов:

- отчет по товарам в магазине;
- отчет по заказам поставщикам;
- отчет по приходу товара;
- отчет по расходу товара.

Опираясь на вышесказанное, можно сделать вывод о том, что автоматизация учета будет экономически выгодна для предприятия и позволит организовать работу магазина на новом уровне.

1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи

При ведении учёта в магазине, накапливается большой объём информации о поступающих данных, а также внутренних документах. В файлах компьютера хранить информацию намного дешевле, чем на бумаге. Использование техники даёт возможность быстро и, что важно, корректно решать задачи учёта товаров, и можно сделать вывод, что решение старым способом, работой с бумажными носителями, крайне неэфективно. Информация, которая хранится на бумажных носителях, со временем приходит в негодность, а электронные документы не ухудшают свои пользовательских качеств, могут храниться вечно, но иногда могут быть утеряны при случайных обстоятельствах.

Электронные базы данных хранят информацию структурированно, и позволяют извлекать её оптимальным для пользователя образом. Очень трудно вести и выполнять поиск данных о товарах, клиентах и продажах в бумажном виде. Поиск информации в информационной системе по сравнению с поиском документов на бумаге наиболее эффективен, так как осуществляется оперативно, удобно для пользователя.

Для выполнения структурно-функционального анализа ведения учета товаров в магазине была разработана структурно-функциональная диаграмма («КАК ЕСТЬ») по методологии SADT (IDEF0) с помощью CASE-средства AllFusion Process Modeler 7.

AllFusion Process Modeler 7 (ранее BPwin) - мощный инструмент моделирования, который используется для анализа, документирования и реорганизации сложных бизнес-процессов. Модель, созданная средствами

BPwin, позволяет четко документировать различные аспекты деятельности: действия, которые необходимо предпринять, способы их осуществления, требующиеся для этого ресурсы и другие. Таким образом, формируется целостная картина деятельности предприятия – от моделей организации работы в маленьких отделах до сложных иерархических структур.

Стандарт IDEF0 предназначен для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающие эти функции [5].

На рисунке 2 представлена контекстная диаграмма:

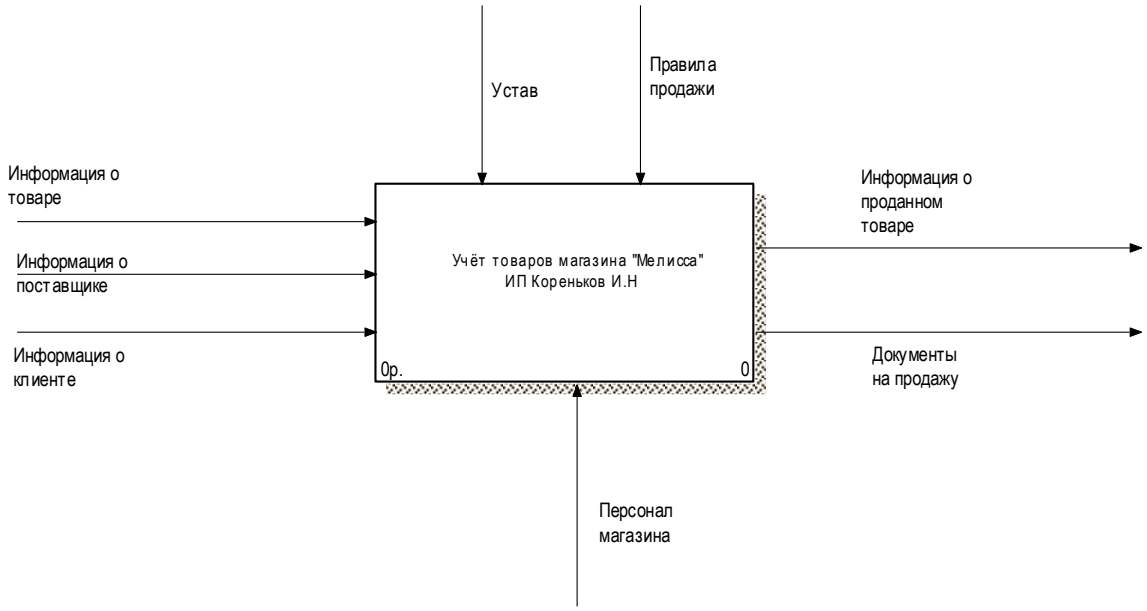


Рисунок 2 - Контекстная диаграмма «Учет товаров магазина «Мелисса» ИП Кореньков И.Н (IDEF0)

На рисунке 3 представлена декомпозиция контекстной диаграммы, в которой выделены три функциональных блока: «Приём товара», «Учёт товара», «Отпуск товара»:

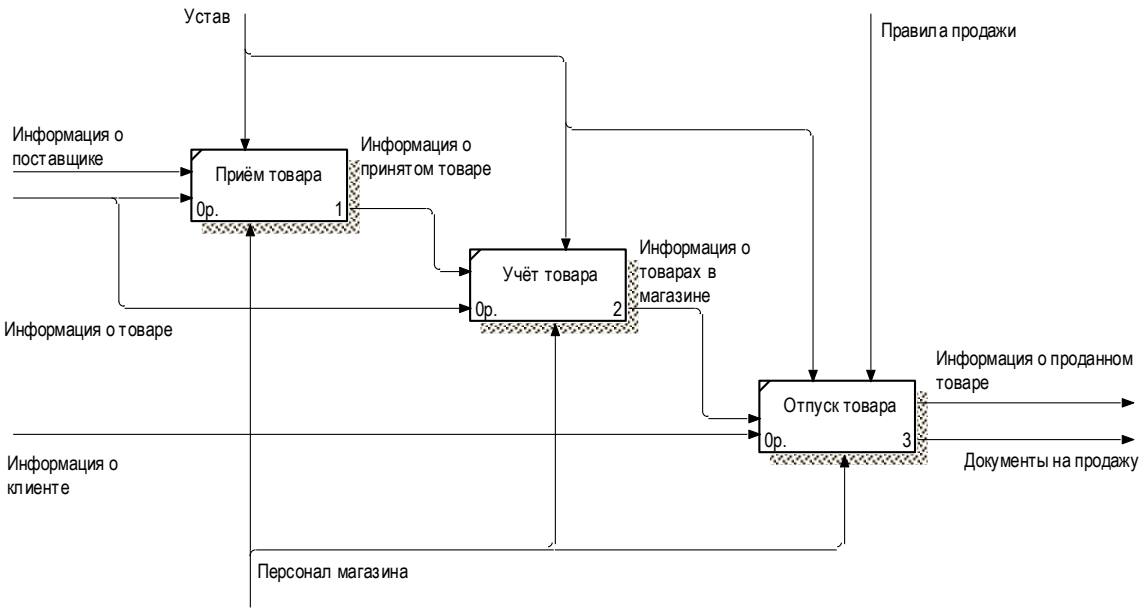


Рисунок 3 – Декомпозиция контекстной диаграммы (IDEF0)

В связи с этим, необходимо задуматься об автоматизации учета товаров в магазине, которая позволит быстро и качественно обрабатывать информацию при помощи вычислительной техники, что, в свою очередь, приведет к снижению трудовых и временных затрат на обработку данных, а также к повышению достоверности обрабатываемых данных. Следует отметить, что информация, хранящаяся в электронном виде более надежна и безопасна при правильном обеспечении современных средств безопасности.

Использование информационной системы системы в организации с большим количеством бумажных документов даст возможность снизить количество выполняемых ручных операций сотрудником магазина и вероятность ошибок, что в сумме увеличит его производительность.

Также при использовании автоматизированной информационной системы можно получать различные отчеты, на основе которых магазин может эффективно анализировать свою деятельность.

1.4 Постановка задачи

1.4.1 Цель и назначение автоматизированного варианта решения задачи

Данная выпускная квалификационная работа направлена на разработку информационной системы учёта товаров для магазина.

В результате решения данной задачи произойдёт сокращение времени выполнения работы, улучшение качества выполняемой работы, и повышение производительности труда сотрудника.

Разрабатываемая система должна решать задачу автоматизации учета товаров магазина и выполнять следующие функции:

- оперативный доступ к имеющимся данным;
- ввод, хранение, изменение и удаление информации о товарах, поставщиках и клиентах;
- ввод, хранение, изменение и удаление информации о приходе и расходе товаров;
- фильтрация информации;
- сортировка информации по убыванию и возрастанию;
- поиск информации;
- формирование отчетов.

Информационная система будет выполнять все функции, необходимые для ведения учета товаров в магазине «Мелисса» ИП Кореньков И.Н..

1.4.2 Общая характеристика организации решения задачи на ЭВМ

Для реализации проектируемой системы был выбран вариант «клиент-серверной» архитектуры, так как она отвечает всем необходимым условиям

проектируемой информационной системы. Клиент-серверная архитектура позволит распределить нагрузку на сеть, объединить рабочие места пользователей в локальную сеть, все пользователи будут использовать централизованное хранилище данных, что позволит избежать устаревания и неточности информации. Клиент-серверная архитектура будет реализована на базе SQL-сервера. SQL-сервер будет выступать в роли координатора клиентских приложений, который сможет обрабатывать сложные запросы, в том числе вызов хранимых процедур и триггеров, что позволит снизить объем передаваемого трафика по сети и блокировку таблиц базы данных во время выборки информации.

Клиентская часть будет реализована в виде набора модулей и исполнимых файлов, реализованных на языке высокого уровня. Пользователь, используя клиентскую часть автоматизированной системы, сможет производить различные манипуляции с данными, находясь сколь угодно далеко от центрального серверного компьютера, при этом возможен вариант установки данной системы на одну пользовательскую машину, что будет отвечать потребностям пользователей, не имеющих в наличии локальной сети.

Входные данные будут вводиться при помощи экранных форм системы, которые предназначены для облегчения пользователю ввода данных. В справочниках базы данных хранится условно-постоянная информация. Далее представлены данные, выступающие в качестве условно-постоянной информации:

- список товаров;
- список категорий товаров;
- список единиц измерения товаров;
- список клиентов;
- список поставщиков;

Условно-постоянная информация поступает ежедневно, однако она может поступать и раз в месяц, и раз в квартал, и раз в год.

Первичная информация также вводится в базу данных. Первичные данные могут быть получены из различных источников устных или письменных, а также из печатных документов.

Для решения задачи автоматизации учета можно выделить следующие этапы:

- создание форм для ввода, редактирования и удаления, фильтрации, сортировки и поиска информации о товарах;
- создание форм для ввода, редактирования и удаления, фильтрации, сортировки и поиска категорий товаров;
- создание форм для ввода, редактирования и удаления информации о единицах измерения товаров;
- создание форм для ввода, редактирования и удаления, фильтрации, сортировки и поиска информации о покупателях;
- создание форм для ввода, редактирования и удаления, фильтрации, сортировки и поиска информации о приходе товаров;
- создание форм для ввода, редактирования и удаления, фильтрации, сортировки и поиска сведений о поставщиках;
- создание форм для ввода, редактирования и удаления, фильтрации, сортировки и поиска информации о продаже товаров клиентам, расчет итоговой суммы;
- создание форм для ввода, редактирования и удаления, фильтрации, сортировки и поиска данных о клиентах;
- создание форм для ввода, редактирования и удаления, фильтрации, сортировки и поиска информации о расходе товаров;
- создание различных отчетов.

Решение задачи автоматизации учета товаров реализовано при помощи диалогового режима, при котором существует возможность пользователя взаимодействовать с вычислительной системой в процессе работы.

Пользовательский интерфейс представлен в виде меню. Главное меню выводится на экран при открытии программы и представляет собой список команд, которые позволяют управлять работой программы.

Результатными документами являются различные отчеты. Для каждого отчета есть возможность вывода на экран и печати на принтере.

1.5 Анализ существующих разработок и обоснование выбора технологии проектирования

На сегодняшний день существует множество информационных систем для ведения учета товаров, однако не все могут подходить под специфику данного магазина. Рассмотрим наиболее известные и распространенные программные продукты.

Среди представленных на российском рынке систем автоматизирования торговли можно отметить предложения фирм

- «Tandem-Soft» (Торговля и учет для Windows);
- «Эйс» («Гепард»);
- «1С» (1С: Торговля);
- Информационные системы и технологии (система «Аспект»);
- «КомТех», «Атлант-Информ» (система «Садко»);
- «Парус», «Галактика», «Мета» (комплекс автоматизирования в розничной торговле);
- «Интеллект-Сервис», «Инфин», «Веста-С» (Управление товарооборотом на базе системы StorS).

Как мы можем видеть, анализ перечня свидетельствует об активности фирм, разрабатывающих программы для автоматизирования.

Из проведенного списка можно сделать вывод, что ни одна из рассмотренных систем не подходит для внедрения в данном магазине.

Основной причиной такого заключения считается слишком обширная функциональность всех рассмотренных программных комплексов. Это

повлечет за собой, в случае внедрения, реализацию множества либо ненужных в деятельности магазина функций, либо их дублирование. А так как стоимость внедрения программных комплексов составляет, как правило, достаточно весомую сумму, это грозит еще и неэффективными вложениями капитала.

Выводы по первой главе

В данной главе была проанализирована деятельность магазина «Мелисса» ИП Кореньков И.Н., были рассмотрены характеристики магазина и виды его деятельности, описана организационная структура, представлены структурно-функциональные диаграммы «КАК ЕСТЬ». Были определены основные требования к разрабатываемой системе учета, которая будет полностью учитывать специфику предприятия, проанализированы существующие разработки и произведено обоснование выбора технологии проектирования.

2 ОБОСНОВАНИЕ ПРОЕКТНЫХ РЕШЕНИЙ

2.1 Обоснование проектных решений по техническому обеспечению

Техническое обеспечение – совокупность технических средств, компьютерной техники, средств передачи информации, используемых в автоматизированных системах управления и в информационных системах, а также документация на эти средства и технологические процессы.

Комплекс технических средств данного проекта составляют:

- ЭВМ (персональный компьютер);
- клавиатура;
- манипулятор-мышь;
- принтер;
- модем;
- эксплуатационные материалы – бумага.

При выборе компьютера нужно руководствоваться некоторыми характеристиками. К таким характеристикам относятся надежность, стоимостные затраты, производительность, простота использования и т.д.. От значения перечисленных параметров зависит возможность работы с требуемыми программными средствами, а, значит, и успех создания системы.

При осуществлении выбора технического обеспечения, наиболее важными критериями для каждого из элементов данной схемы, являются:

- производительность процессора;
- разрешение монитора;
- возможность подключения периферийных устройств;
- объём оперативной памяти;
- объём жёсткого диска.[29].

Чтобы решить поставленную задачу, нужно использовать персональный компьютер с высоким уровнем вычислительной мощности, объемом оперативной памяти от 2Гб. Система оптимизирована для работы в экранном разрешении 1024x768 на мониторе с диагональю 17” дюймов. Все эти технические средства обладают достаточной конфигурацией для решения задачи.

Технические характеристики рабочей станции приведены в таблице 1.

Таблица 1 - Характеристики рабочей станции

Производитель	ASUS
Процессор	Intel Core i5-2450M CPU @ 2.50 GHz
Объём оперативной памяти	4,00 Гб
Объём жёсткого диска	1 TB
Оптический привод	DVD-RW
Видеокарта	NVIDIA GeForce 610M
Разъемы на материнской плате	3 USB, выход S/PiDIF, 2xCOiM, D-Subi, DVIi, Etheirnet, PS клавиатура
Операционная система	Windows 7 Максимальная
Размеры (ШхГхВ)	273 x 282 x 85 (мм)

Таким образом, в существующее техническое обеспечение не нужно вносить изменения, и тем самым затраты на разработку информационной системы будут сокращены.

2.2 Обоснование проектных решений по информационному обеспечению

Информационное обеспечение системы – информация, необходимая для управления экономическими процессами, содержащаяся в базах данных информационных систем. Различают два вида обеспечения: вне машинное и внутри машинное.

Информационное обеспечение включает в себя: единую систему классификации и кодирования информации, унифицированную систему документации, схему информационных потоков, циркулирующих на предприятии, методологию построения баз данных..

Вне-машинное обеспечение – это вся совокупность информации, основную часть которой составляют документы. Различат первичные документы (входные) такие, как налоговые декларации и другая отчетность налогоплательщиков, и отчетные (выходные) документы, к которым относятся отчетность налогоплательщиков, обработанная и подготовленная для отдела по работе с налогоплательщиками и отдела камеральных проверок.

Внутри-машинное ИО – это представление данных на машинных носителях, в виде специальным образом организованных массивов (файлов), БД и их информационных связей. Внутри машинное информационное обеспечение подсистемы создает информационную среду, направленную на выполнение сотрудниками своих профессиональных обязанностей. Эта область характеризуется набором объектов, их свойств и взаимосвязей. Для каждого объекта выделяется набор его характеристик, свойств[39].

Для того чтобы приспособить экономическую информацию для эффективного поиска, обработки и передачи по каналам связи, ее необходимо представить в цифровом виде. С этой целью ее нужно сначала упорядочить (классифицировать), а затем формализовать (закодировать) с использованием классификатора.

Классификатор – это документ, с помощью которого осуществляется формализованное описание экономической информации в ЭИС, содержащей наименования объектов, наименования классификационных группировок и их кодовые обозначения.

По сфере действия выделяют следующие виды классификаторов: международные, общегосударственные (общесистемные), отраслевые и локальные классификаторы.

Основной компонентой вне-машинного информационного обеспечения считается система документации, применяемая в процессе управления экономическим объектом. Под документом понимается определенная совокупность сведений, используемая при решении экономических задач, расположенная на материальном носителе в соответствии с установленной формой.

Система документации – это совокупность взаимосвязанных форм документов, регулярно используемых в процессе управления экономическим объектом. Отличительной особенностью системы экономической документации считается большое разнообразие видов документов.

Существующие системы документации, характерные для неавтоматизированных ЭИС, отличаются большим количеством разных типов форм документов; большим объемом потоков документов и их запутанностью; дублированием информации в документах и работ по их обработке и, как следствие, низкой достоверностью получаемых результатов. Обработка документов в таких системах занимает почти половину времени работников. При необходимости упростить систему документации, используют следующие подходы:

- проведение унификации и стандартизации документов;
- введение безбумажной технологии, основанной на использовании электронных документов и новых информационных технологий их обработки.

Входные документы для решаемой задачи желательно получать через сеть, поэтому они должны быть представлены в файлах заранее согласованной структуры. Для упрощения использования таких файлов, они должны иметь табличную форму (что несложно осуществить, т. к. эти документы являются результатными в других задачах и выводятся программными средствами).

Итак, информационная система должна выполнять следующие функции:

1) работать с товарами магазина и выполнять требуемые операции по учёту движения товаров;

2) возможность поиска необходимого товара по ключевым словам для ускорения работы;

3) изменение уже имеющейся базы данных;

4) возможность настройки параметров работы (фильтр по товарам, суммирование и вывод итогов по всем продажам).

Требования к защите данных.

Для обеспечения защиты данных в информационной системе предусмотрено разграничение прав доступа, которое регулируется путем установления паролей для пользователей, которые в дальнейшем будут обслуживать ее работу. Для обеспечения защиты данных про сбои в сети питания ПК либо аварийном завершении работы программы будет предусмотрен режим автоматического сохранения.

Требования к надежности.

Программа должна нормально функционировать при бесперебойной работе компьютера. При возникновении сбоя в работе аппаратуры, восстановление нормальной работы информационной системы должно производиться после: перезагрузки операционной системы либо перезапуска исполняемого файла программы, а также повторного выполнения действий, потерянных до последнего сохранения информации.

2.3 Обоснование проектных решений по программному обеспечению

Программное обеспечение – это набор программных средств, обеспечивающих функционирование информационной системы.

Программное обеспечение, можно условно разделить на три категории:

− системное программное обеспечение (программы общего пользования) необходимо для функционирования компьютера, работы с файлами, защиты программ и данных, а также для разработки прикладного программного обеспечения;

− прикладное программное обеспечение - совокупность программ для решения общих универсальных задач;

− инструментальное программное обеспечение (системы программирования), обеспечивающее разработку новых программ для компьютера на языке программирования [8].

Операционная система предоставляет пользователю интерфейс по взаимодействию с ресурсами машины, а также обеспечивает оптимальное распределение ресурсов между различными процессами.

Операционные системы могут различаться особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера (процессорами, памятью, устройствами), особенностями использованных методов проектирования, типами аппаратных платформ, областями использования и многими другими свойствами [8].

В настоящее время имеется множество операционных систем, и перед пользователем встает задача выбора операционной системы, которая будет отличаться от других теми или иным критериями. Любой операционной системе присущи свои плюсы и минусы. При выборе операционной системы, пользователь должен понимать, насколько та или иная операционная система обеспечит решение необходимых ему задач.

На компьютере магазина «Мелисса» ИП Кореньков И.Н. установлена многопользовательская, многозадачная 64-разрядная операционная система с графическим пользовательским интерфейсом Windows 7 Максимальная. Эта версия раскрывает максимальный потенциал Windows 7 и является наиболее универсальным решением. В неё включены все возможности всех других версий, включая запуск многих приложений Windows XP в режиме Windows

XP, шифрование данных с помощью функций BitLocker и BitLocker To Go. Дополнительную гибкость версии придаёт возможность работы на 35 языках.

Функциональные возможности Windows 7 Максимальная:

− значительное повышение производительности: быстрый запуск и завершение работы, быстрое переключение режимов и сеансов;

− модули управления ресурсами системы, оптимизация распределения нагрузки, защита от непредвиденных сбоев и зависаний, незаметная для пользователя автоматическая диагностика системы;

− интеллектуальная технология ускорения поиска необходимых файлов и программ и удобная система просмотра результатов поиска;

− система родительского контроля для предотвращения несанкционированного использования и блокировки нежелательных данных;

− гибкая система настройки Windows 7 Максимальная;

− обеспечение совместимости устройств, предоставление всех необходимых драйверов через центр обновлений Windows;

− обеспечение полной совместимости приложений в рамках системы, встроенные средства совместимости;

− возможность легко переписать данные и параметры со старого ПК на новый.

− фундаментальные функции безопасности: защита ядра от изменений, усиление защиты служб, предотвращение несанкционированного исполнения данных, предохранение от случайных изменений структуры адресного пространства, сохранение обязательных уровней целостности;

− увеличение времени работы от аккумулятора за счёт снижения общего числа фоновых задач, диагностики причин перерасхода энергии, автоматическое сохранение питания во время простоя, снижение яркости дисплея, спящий режим и так далее;

− для IT-специалистов в Microsoft Windows 7 Максимальная предусмотрены работа через командную строку, поддержка WMI-сценариев

для просмотра данных средства анализа стабильности системы (RAC), мониторинг стабильности системы [40].

В организации используется прикладное программное обеспечение Microsoft Office 2007, которое содержит всё необходимое для работы с текстом, таблицами, диаграммами, презентациями, базами данных, электронной почтой и многим другим.

На компьютере установлен антивирусный продукт Kaspersky Internet Security 2015, необходимый для обеспечения безопасности операционной системы.

Для разработки информационной системы учета товаров будут использоваться система управления базами данных FireBird, утилита IBExpert и среда визуального программирования Borland C++ Builder 6.0.

FireBird - это мощная, компактная, реляционная СУБД, основанная на ядре Borland InterBase и поддерживающая архитектуру «клиент-сервер». Она предназначена для хранения и обработки больших объемов информации, в условиях работы нескольких пользователей. Для управления базой данных сервер FireBird использует домены, просмотры, хранимые процедуры, триггеры, генераторы, транзакции, а также пользовательские функции.

Firebird (InterBase) обладает такими преимуществами, как:

- многоверсионная архитектура, что обеспечивает параллельную обработку оперативных и аналитических запросов;
- компактность (дистрибутив 5Mb);
- высочайшая эффективность;
- самая полная языковая поддержка для хранимых процедур и триггеров.

Firebird (InterBase) является сервером баз данных (SQL сервер). Один SQL сервер Firebird сможет обрабатывать сразу несколько независимых баз данных, с множеством пользовательских соединений на каждой [5].

Firebird широко используется с 2001 года. Это коммерчески независимый проект C и C++ программистов, технических советников и

разработчиков мультиплатформенных систем управления базами данных, основанный на исходном тексте, выпущенном корпорацией Borland 25 июля 2000 года в виде свободной версии Interbase 6.0. Выгодно отличается от MS SQL компактностью, кросплатформенностью (поддерживает Windows, GNU/Linux, FreeBsd, Solaris, OS X, HP-UX). Удобен и прост в использовании. Любой начинающий, знающий SQL на уровне оператора Select, легко в нем разберется. И, что немаловажно, за него не нужно платить. Допускает подключение пользовательских функций (UDF) в виде dll-библиотек, разработанных на любом языке программирования с использованием любой среды разработки (Delphi, C++Builder, MS Visual Studio, C++, Pascal). Имеется достаточно большое количество средств доступа к базам данных Firebird (Interbase) из-под ODBC, ADO, ADO.NET, BDE, php, perl, python. Этот sql сервер практически не имеет ограничений к применению.

Firebird (InterBase) обладает несомненными преимуществами перед другими СУБД этого уровня, такими как: mysql, msql, postgresql. От таких СУБД (sql server) как MS SQL и Oracle его выгодно отличает:

- компактный размер;
- простота установки и администрирования;
- бесплатное распространение.

Для работы с FireBird используют утилиту IBExpert, которая позволяет не только полностью управлять структурами баз данных, но также создавать механизмы управления базой данных и отлаживать их [22].

IBExpert – инструмент для разработки FireBird баз данных на основе технологии InterBase 6.0. IBExpert позволяет осуществлять проектирование с заметной легкостью, быстротой, надежностью и удобством для разработчика. IBExpert включает много инструментов и особенностей кодирования: визуальные редакторы для всех типов базы данных, SQL- редакторы и сценарии, отладчик для хранимых процедур, генераторов и триггеров, исключения, домены и многое другое [22].

Borland C++ Builder 6.0 - выпущенное компанией Borland средство быстрой разработки приложений, позволяющее создавать приложения на языке C++, используя при этом среду разработки и библиотеку компонентов Delphi. C++Builder включает обширный набор средств, которые повышают производительность труда программистов и сокращают продолжительность цикла разработки, а также поставляется ряд компонентов InterBase eXpress (IBX), позволяющих работать с сервером Firebird [28].

Borland C++ Builder 6.0 обеспечивает непревзойденную производительность и все преимущества визуальной разработки на основе легконастраиваемой среды AppBrowser IDE, предоставляет средства параллельной разработки, позволяющие параллельно визуально редактировать текст программы и изменять внешний вид используемых форм, специальные средства для повышения скорости программирования, такие как Code Insight, CodeBrowser, ClassExplorer, ClassCompletion и ParameterCompletion [1].

2.4 Обоснование проектных решений по технологическому обеспечению

Технологическое обеспечение определяет последовательность процедур и задач по работе с информационной системой. Технологическое обеспечение включает в себя процедуры, реализуемые конкретной информационной технологией.

Весь технологический процесс можно подразделить на процессы сбора и ввода исходных данных в вычислительную систему, процессы размещения и хранения данных в памяти системы, процессы обработки данных с целью получения результатов и, процессы выдачи данных в виде, удобном для восприятия пользователем [34].

Операции сбора и регистрации данных осуществляются с помощью различных средств. Различают механизированный, автоматизированный и автоматический способы сбора и регистрации информации и данных.

− механизированный способ представляет собой сбор и регистрацию информации непосредственно человеком с использованием простейших приборов (весы, счетчики и так далее);

− автоматизированный способ предполагает использование машиночитаемых документов, регистрирующих автоматов, универсальных систем сбора и регистрации, обеспечивающих совмещение операций формирования первичных документов и получения машинных носителей;

− автоматический способ используется в основном при обработке данных в режиме реального времени. Информация с датчиков, учитывающих ход производства: выпуск продукции, затраты сырья, простои оборудования и так далее – поступает непосредственно в ЭВМ [34].

На данный момент времени в магазине используется механизированный способ сбора и регистрации информации. С помощью пишущих средств, вся поступившая информация приводится к читаемому виду. Однако этот способ имеет множество недостатков по сравнению с автоматизированным способом, например, низкая производительность и качество выполнения персоналом организации рабочих задач.

В разрабатываемой информационной системе технологический процесс сбора информации будет представлять собой ввод информации с первичных документов в базу данных, при котором будет контролироваться допустимость введенных значений и обеспечиваться ввод данных путем выбора из списка. Все внесенные данные будут храниться в базе данных. Поступившая информация будет подвержена обработке и на ее основе можно будет получить результатную информацию в виде отчетов и выходных документов.

Информационная база (база данных) хранится в файле DB.FDB, который управляется с помощью клиент-серверной СУБД FireBird и утилиты IBExpert.

Для исключения возникающих в ходе работы ошибок будет проведено обучение персонала функциональным возможностям разработанной системы. Таким образом, будет усовершенствован учет товаров в данной торговой организации.

2.5 Обоснование выбора программных средств

Для проектирования информационных систем широко используются CASE-технологии. В настоящее время существует достаточное количество CASE-средств, например: Oracle Designer, BPwin, ERwin, Rational Rose, Silverrun и другие. Однако проанализировав все преимущества и недостатки данных программных средств, были выбраны CASE-средства BPwin и Erwin, так как эти программные продукты имеют широкие функциональные возможности для проектирования информационной системы, низкую стоимость, а также ранее был получен опыт работы с ними.

Для разработки функциональных моделей информационной системы учета товаров будем использовать программное средство AllFusion Process Modeler 7. Данный продукт представляет собой CASE-средство для моделирования, анализа, документирования и оптимизации бизнес-процессов. Поддерживается использование следующих видов диаграмм:

− Buisiness Process (методология IDEF0). Данный вид диаграмм предназначен для описания бизнес процессов путем разбиения его на более мелкие процессы, которые взаимодействуют друг с другом. Каждый процесс осуществляет преобразование входящей информации в исходящую путем использования какого-то механизма и под каким-то управляющим воздействием.

− Process Flow (методология IDEF3) предполагает представление процесса как упорядоченной последовательности событий.

− Data Flow (методология DFD), или диаграммы потоков данных. В данной методологии предполагается разбиение процесса на более мелкие подпроцессы, которые путем взаимодействия друг с другом образуют желаемый результат [20].

Для проектирования структуры базы данных информационной системы учета товаров используется программное средство AllFusion Erwin Data Modeler 7, который позволяет визуализировать структуру информационной базы, а также взаимосвязи между различными блоками данных. Данный продукт позволяет не только «нарисовать» модель базы данных, но и на основании построенной модели предусмотренное автоматическое создание структуры базы данных, то есть создание таблиц базы, их полей и установление связей между различными таблицами.

AllFusion ERwin Data Modeler позволяет осуществлять автоматизированное проектирование реляционных баз данных в визуальном редакторе. Для этого используется три основных типа объектов:

1) Сущности, которые представляют собой таблицу базы данных. Сущности обычно состоят из ряда атрибутов.

2) Атрибуты, представляющие конкретное поле таблицы базы данных. Атрибут представляет конкретное свойство, которое может принимать различные значения.

3) Связи, которые отражают взаимосвязи между таблицами базы [20].

Выводы по второй главе

В данной главе были обоснованы проектные решения по техническому, информационному, программному, технологическому обеспечению задачи, а также был обоснован выбор программных средств, используемых для проектирования информационной системы.

3 ПРОЕКТНАЯ ЧАСТЬ

3.1.Информационное обеспечение задачи (комплекса задач, АРМ)

3.1.1 Информационная модель и её описание

Важным этапом в проектировании информационной системы является построение информационной модели.

Информационная модель — модель объекта, представленная в виде информации, описывающей существенные для данного рассмотрения параметры и переменные величины объекта, связи между ними, входы и выходы объекта и позволяющая путём подачи на модель информации об изменениях входных величин моделировать возможные состояния объекта [6].

В данной выпускной квалификационной работе информационная модель будет построена в двух формах:

- схема данных, представленная в приложении А;
- структурно – функциональная модель, построенная с использованием CASE-средства AllFusion Process Modeler.

Новая информационная модель «КАК ДОЛЖНО БЫТЬ» претерпит некоторые изменения, то есть вся поступившая информация будет фиксироваться, храниться и обрабатываться в разработанной автоматизированной системе, а также появиться возможность получать различную отчетность и анализировать ее.

Рассмотрим структурно – функциональную модель складского учета в организации в нотации IDEF0. Контекстная диаграмма представлена на рисунке 4:

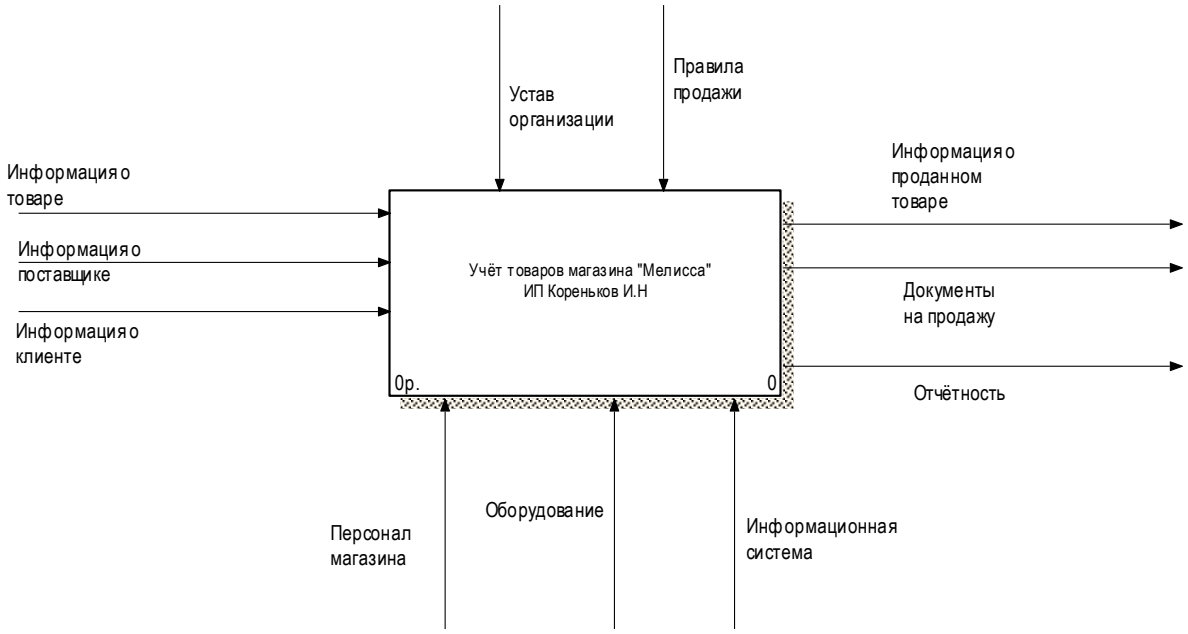


Рисунок 4 - Контекстная диаграмма (IDEF0)

Из контекстной диаграммы видно, что входной информацией являются информация о товаре, информация о поставщике, информация о клиенте.

Выходная информация представляет собой информацию о проданном товаре, документы на продажу и отчетность.

В качестве управляющих объектов выступают устав организации, правила и процедуры.

К механизмам исполнения относятся оборудование, персонал и информационная система.

После построения контекстная диаграмма детализируется с помощью диаграммы декомпозиции первого уровня. На этой диаграмме отображаются функции системы, которые должны быть реализованы в рамках основной функции. Декомпозиция контекстной диаграммы представлена на рисунке 5:

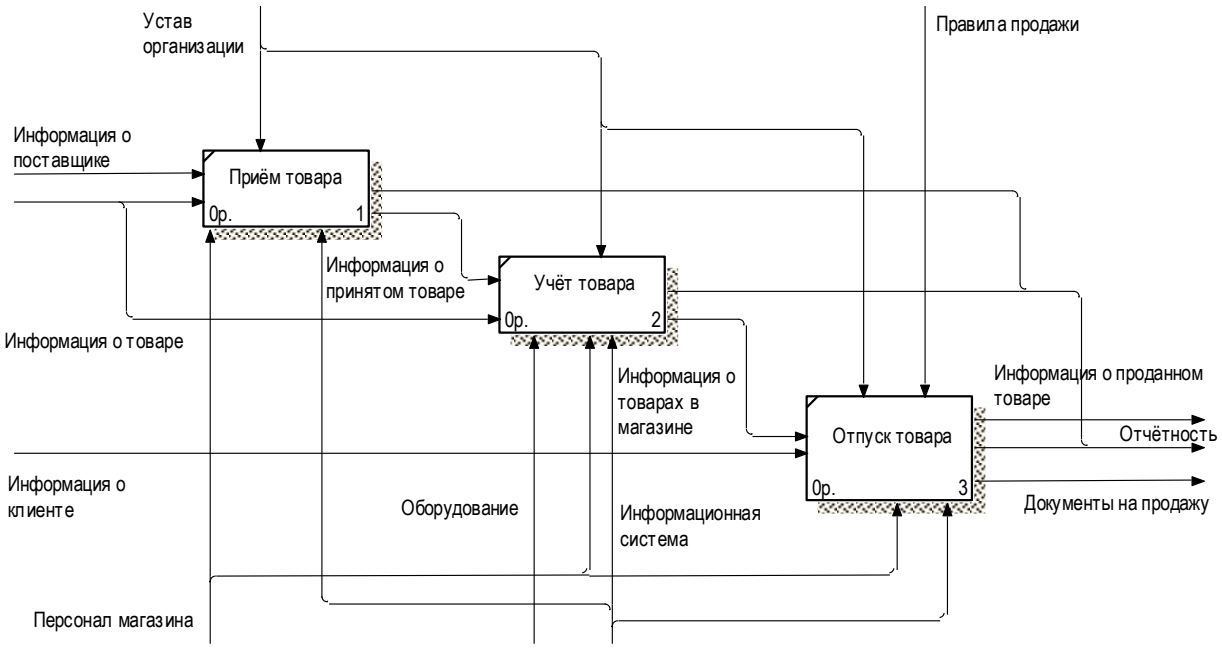


Рисунок 5 –Декомпозиция контекстной диаграммы (IDEF0)

На рисунке видно, что диаграмма декомпозиции содержит три функциональных блока: «Приём товара», «Учёт товара», «Отпуск товара»:

− блок «Приём товара», в котором происходит прием товара от поставщика в магазин на основании сведений от поставщика, документов от поставщика и информации о товаре. В результате выполнения данного блока будет получена информация о принятом товаре и отчетность;

− блок «Учёт товара», в котором осуществляется хранение товара в магазине на основании информации о принятом товаре. В результате преобразования этого блока будет получена информация о товарах в магазине и соответствующая отчетность.

− блок «Отпуск товара», в котором осуществляется отпуск товаров клиентам на основании информации о товарах и данных о клиенте. В результате преобразования данного блока будет получена информация о проданном товаре, документы на продажу и соответствующая отчетность.

Диаграмма декомпозиции функционального блока «Отпуск товара» показана на рисунке 6:

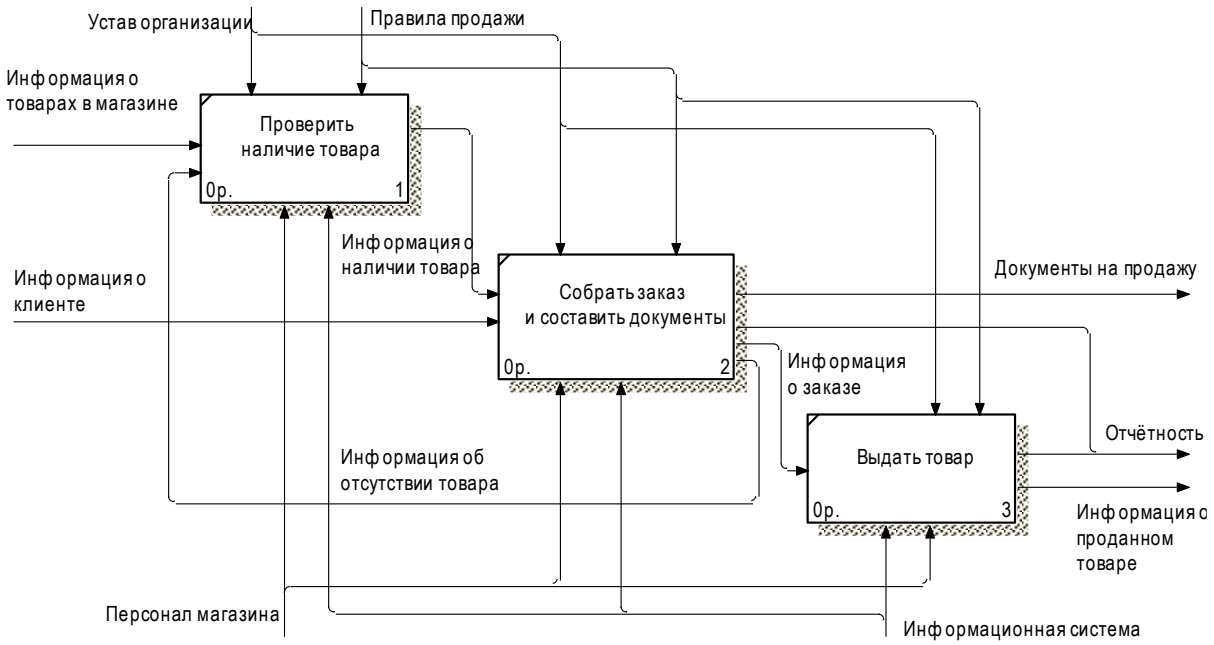


Рисунок 6 – Диаграмма декомпозиции блока «Отпуск товара» (IDEF0)

Диаграммы декомпозиции остальных блоков вынесены в приложение Б.

3.1.2 Используемые классификаторы и системы кодирования

Классификатором называется систематизированный перечень наименованных объектов, каждому из которых в соответствие дан уникальный код. Классификация объектов производится согласно правилам распределения заданного множества объектов на подмножества (классификационные группировки) в соответствии с установленными признаками их различия или сходства.

Кодирование представляет собой присвоение кодов классификационной группировке или объекту классификации. Кодирование предназначено для формализованного описания наименований объектов, характеристик. группировок. Обычно кодирование представляет собой процесс обозначения исходного множества объектов или сообщений набором

символов заданного алфавита на основе совокупности определенных правил [26].

Существуют порядковый, серийно-порядковый, последовательный и параллельный методы кодирования.

При порядковом методе кодирования кодовым обозначением служат числа натурального ряда и каждый объект кодируется с помощью текущего номера, который однозначно идентифицирует этот объект.

При серийно-порядковом методе кодами служат числа натурального ряда с закрепленной отдельной серией этих чисел за объектами классификации с одинаковыми признаками.

При последовательном методе в кодовом обозначении знаки на каждой ступени деления зависят от результатов разбиения на предыдущих ступенях. В результате кодовое обозначение группировки дает информацию о последовательности признаков, характеризующих эту группировку.

При параллельном методе признаки классификации кодируются независимо друг от друга определенными разрядами или группой разрядов кодового обозначения [26].

В таблице 2 представлены используемые классификаторы.

Таблица 2 – Используемые классификаторы

Наименование кодируемого множества объектов	Значность кода	Система кодирования	Система классификации	Вид классификатора
1	2	3	4	5
Код товара	6	Порядковая	Отсутствует	Локальный
Код документа	2	Порядковая	Отсутствует	Локальный
Код содержания документа	4	Порядковая	Отсутствует	Локальный

Продолжение таблицы 2

1	2	3	4	5
Код контрагента	3	Порядковая	Отсутствует	Локальный
Код типа контрагента	3	Порядковая	Отсутствует	Локальный
Код единицы измерения	1	Порядковая	Отсутствует	Локальный
Код номенклатуры	3	Порядковая	Отсутствует	Локальный
Код категории номенклатуры	6	Порядковая	Отсутствует	Локальный
Код типа документа	6	Порядковая	Отсутствует	Локальный
Код состояния документа	6	Порядковая	Отсутствует	Локальный

3.1.3 Характеристика первичных документов с нормативно-справочной и входной оперативной информацией

Входная информация представляет собой всю информацию, необходимую для решения задачи и расположенную на различных носителях, например, в первичных документах, на машинных носителях, в памяти персонального компьютера.

Вся нормативно-справочная информация хранится в справочниках, содержащих сведения о товарах, категориях товара, накладных и содержании накладных, поставщиках и клиентах. В справочниках хранится информация многократного использования, предусмотрена возможность добавления, удаления и редактирования содержащихся данных.

Справочник «Номенклатура» содержит всю необходимую информацию о товаре. Пример заполнения справочника представлен на рисунке 7:

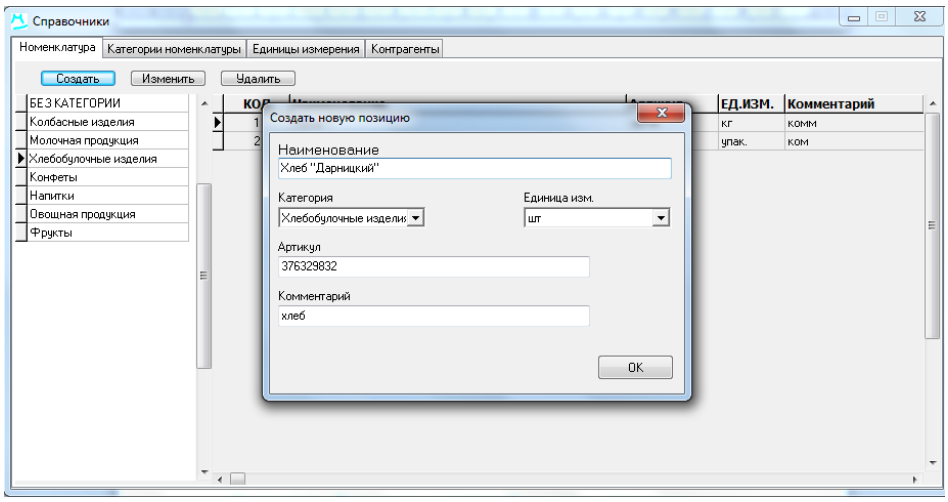


Рисунок 7 – Экранная форма справочника «Номенклатура»

Справочник «Категория» хранит информацию о категориях товаров.

Справочники «Единицы измерения» содержит сведения о единицах измерения товаров.

В справочнике «Контрагенты – Моя организация» хранятся данные о данном магазине. На рисунке 8 представлена форма заполнения справочника:

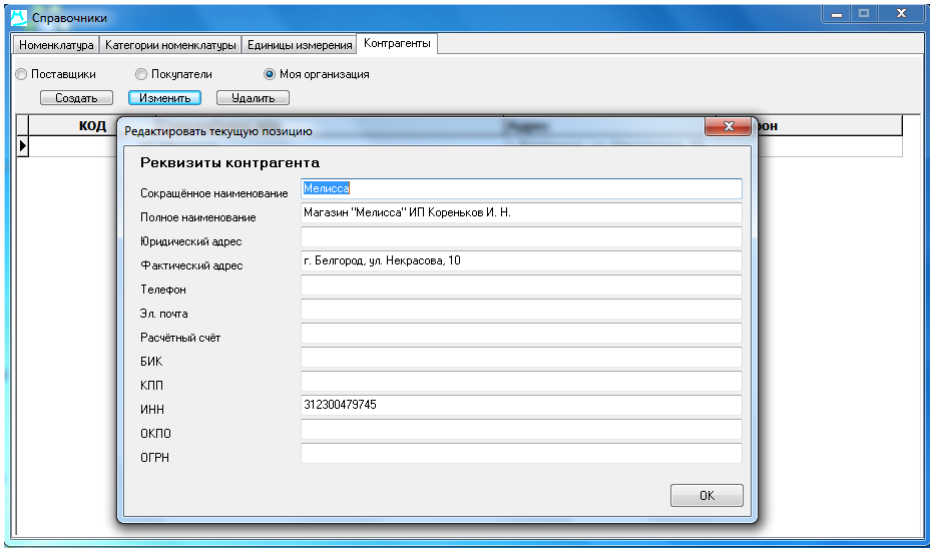


Рисунок 8 - Экранная форма справочника «Контрагенты – Моя организация»

Справочник «Контрагенты - Покупатели» содержит сведения о клиентах магазина. Пример заполненного справочника продемонстрирован на рисунке 9:

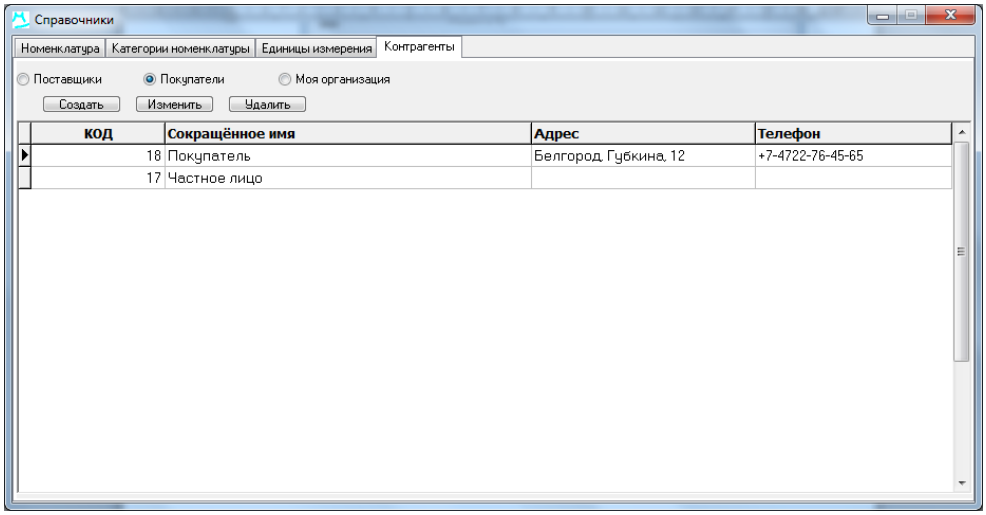


Рисунок 9 - Экранная форма справочника «Контрагенты - Покупатели»

В справочнике «Поставщики» хранятся данные о поставщиках. На рисунке 10 представлена форма заполнения справочника:

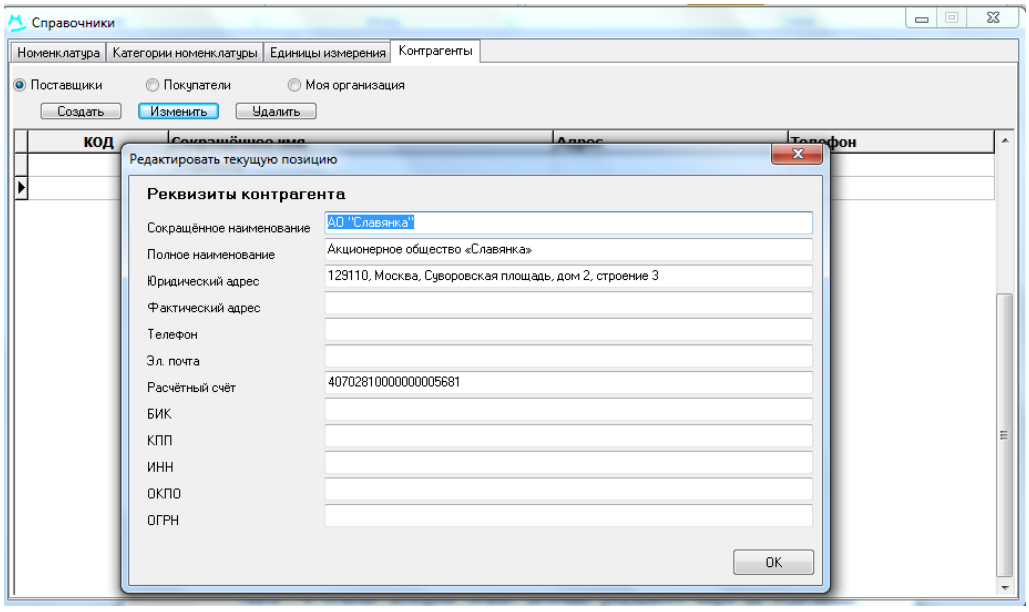


Рисунок 10 - Экранная форма справочника «Поставщики»

Таким образом, были рассмотрены все справочники разработанной информационной системы учета товаров.

3.1.4 Характеристика базы данных

3.1.4.1 Характеристика инфологической модели БД

Инфологическая модель представляет собой модель предметной области, определяющая совокупности информационных объектов, их атрибутов и отношений между объектами, динамику изменений предметной области, а также характер информационных потребностей пользователей. Инфологическая модель предметной области описана моделью "сущность—связь", в основе которой лежит деление реального мира на отдельные различимые сущности, находящиеся в определенных связях друг с другом, с использованием программного средства AllFusion Erwin Data Modeler 7.

Возможны две точки зрения на информационную модель и, соответственно, два уровня модели. Первый - логический (точка зрения пользователя) - описывает данные, задействованные в бизнесе предприятия. Второй - физический - определяет представление информации в базе данных. ERwin объединяет их в единую диаграмму, имеющую несколько уровней представления [14].

Логическая модель базы данных представлена на рисунке 11:

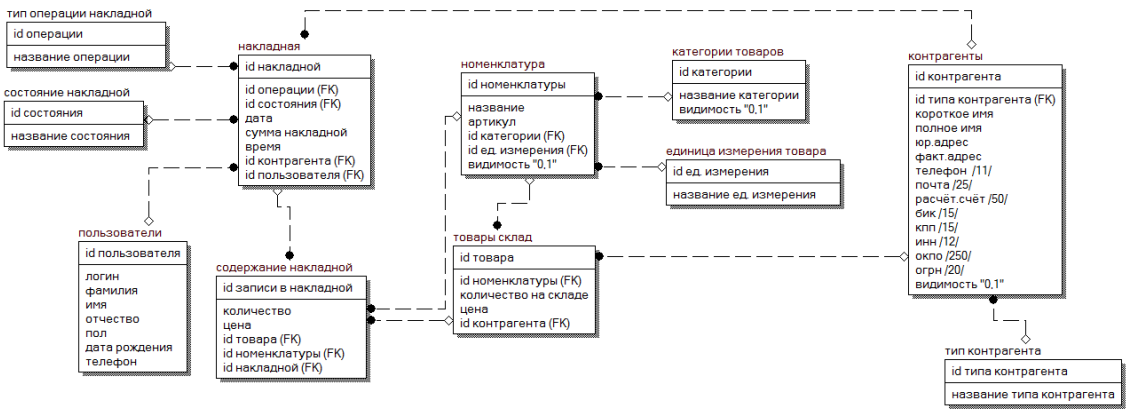


Рисунок 11 – Логическая модель базы данных

На основе логической модели базы данных была спроектирована физическая модель, показанная на рисунке 12:

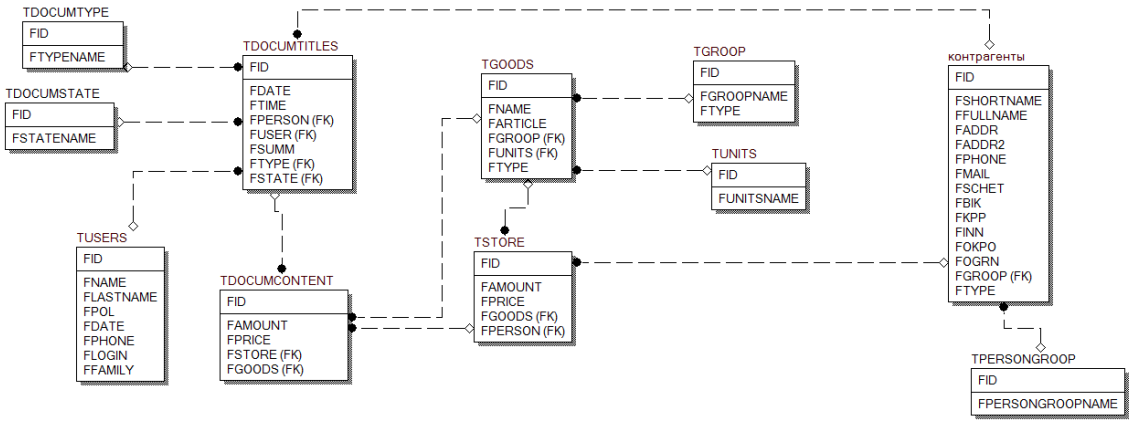


Рисунок 12 – Физическая модель базы данных

3.1.4.2 Характеристика даталогической модели БД

Даталогическая модель - это модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации. При этом даталогическая модель разрабатывается с учётом конкретной реализации СУБД, а также с учётом специфики конкретной предметной области на основе ее инфологической модели [14].

В таблице 3 представлено описание даталогической модели базы данных:

Таблица 3 – Описание даталогической модели базы данных

Сущность	Идентификатор таблицы	Атрибут	Идентификатор поля	Тип поля
1	2	3	4	5
Тип операции накладной	TDOCUMTYPE	Код операции	FID	integer
		Название операции	FTYPE	varchar
Состояние накладной	TDOCUMSTATE	Код состояния	FID	integer
		Название состояния	FSTATE	varchar

Продолжение таблицы 3

1	2	3	4	5
Накладная	TDOCUMTITLES	Код накладной	FID	bigint
		Тип накладной	FTYPE	smallint
		Состояние накладной	FSTATE	smallint
		Дата накладной	FDATE	date
		Время накладной	FTIME	time
		Сумма накладной	FSUMM	decimal
		Код пользователя	FUSERID	smallint
		Код контрагента	FPERSONID	smallint
Пользователи	TUSERS	Код пользователя	FID	integer
		Логин	FLOGIN	varchar
		Фамилия	FLASTNAME	varchar
		Имя	FNAME	varchar
		Пол	FMALE	varchar
		Дата рождения	FDATE	date
		Телефон	FPHONE	varchar
Содержание накладной	TDOCUMCONTENT	Код записи в накладной	FID	integer
		Код накладной	FDOCID	integer
		Код товара	FSTOREID	integer
		Код номенклатуры	FGOODSID	integer
		Количество	FAMOUNT	numeric
		Цена	FPRICE	decimal
Номенклатура	TGOODS	Код номенклатуры	FID	integer
		Название	FGOODSNAME	varchar
		Код категории	FGROOP	integer
		Код единицы измерения	FUNIT	integer
		Комментарий	FCOMMENT	varchar
		Артикул	FARTICLE	varchar

Продолжение таблицы 3

1	2	3	4	5
Контрагенты	TPERSON	Код контрагента	FID	integer
		Код типа контрагента	FTYPE	integer
		Короткое имя	FSHORTNAME	varchar
		Полное имя	FFULLNAME	varchar
		Юридический адрес	FADDR	varchar
		Фактический адрес	FADDR2	varchar
		Телефон	FPHONE	varchar
		Расчётный счёт	FSCHET	varchar
		БИК	FBIK	varchar
		КПП	FKPP	varchar
		ИНН	FINN	varchar
		ОГРН	FOGRN	varchar
		ОКПО	FOKPO	varchar
Тип контрагента	TPERSONGROOP	Код типа контрагента	FID	integer
		Название типа контрагента	FNAMEGROOP	varchar
Единицы измерения номенклатуры	TUNITS	Код единицы измерения	FID	integer
		Название единицы измерения	FUNITSNAME	varchar
Товары	TSTORE	Код товара	FID	integer
		Код номенклатуры	FGOODSID	integer
		Количество	FAMOUNT	numeric
		Цена	FPRICE	decimal
		Код контрагента	FPERSON	integer
Категории номенклатуры	TGROOP	Код категории	FID	integer
		Название категории	FNAME	varchar

3.1.5 Характеристика результатной информации

3.1.5.1 Характеристика таблиц с результатной информацией

Таблицы с результатной информацией формируются в результате запросов к объектам, хранящим входные данные - справочникам с условно-постоянными сведениями [28].

В разработанной информационной системе были созданы таблицы и процедуры для выборки. Например, на рисунке 13 показана работа процедуры, позволяющей просмотреть полную информацию обо всех документах магазина, созданной на основе таблиц «Тип документа», «Состояние документа», «Содержание документа», «Контрагенты», «Номенклатура» и «Категории номенклатуры». Информация, полученная при помощи процедуры, подлежит дальнейшему хранению в базе данных и при необходимости ее можно редактировать, удалять и обрабатывать, а также добавлять новую информацию.

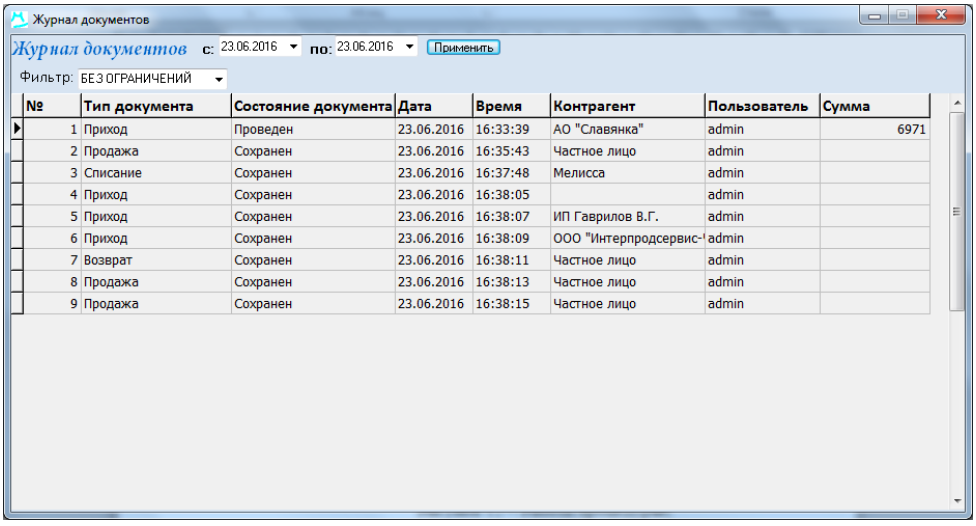


Рисунок 13 – Работа процедуры

Далее приведён SQL-код этой процедуры:

```
create or alter procedure PDOCUMTITLES_GETTITLE (
    VIN_SHOWDEL smallint,
```

```
    VIN_STARTDATE date,
    VIN_ENDDATE date,
    VIN_METHOD smallint,
    VIN_VALUE integer,
    VIN_USERID integer)
  returns (
    VID bigint,
    VTYPENAME varchar(20),
    VTYPEID smallint,
    VSTATEID smallint,
    VSTATENAME varchar(15),
    VDATE date,
    VTIME time,
    VPERSONID bigint,
    VPERSON varchar(30),
    VUSER varchar(20),
    VSUMM decimal(15,2))
  as
  BEGIN
   FOR SELECT D.FID, D.FTYPE,b.ftype, D.FSTATE,a.FSTATE, D.FDATE, D.FTIME,
D.FPERSON, P.FSHORTNAME, U.FLOGIN, D.FSUMM
      FROM ((((TDOCUMTITLES D
      LEFT JOIN TPERSON P ON D.FPERSON = P.FID)
      LEFT JOIN TUSERS U ON D.FUSER = U.FID)
      LEFT JOIN tdocumstate a ON D.fstate = a.fid)
      LEFT JOIN tdocumtype b ON D.ftype = b.fid)
      WHERE ((:VIN_SHOWDEL = 0 AND
          D.FSTATE IN (1, 2) /*ОТОБРАЖАТЬ ТОЛЬКО НЕ УДАЛЕННЫЕ*/
          OR (:VIN_SHOWDEL = 1)) /*ОТОБРАЖАТЬ ВСЕ*/
          AND
          ((:VIN_METHOD = 0 AND D.FUSER = :VIN_USERID)
          OR (:VIN_METHOD IN (1, 2) AND D.FPERSON = :VIN_VALUE AND D.FUSER =
:VIN_USERID)
          OR (:VIN_METHOD = 3 AND D.FTYPE = :VIN_VALUE AND D.FUSER =
:VIN_USERID)
          OR (:VIN_METHOD = 5 AND (:VIN_VALUE = 0 OR (:VIN_VALUE > 0 AND
D.FUSER = :VIN_VALUE))))
          AND (D.FDATE BETWEEN :VIN_STARTDATE AND :VIN_ENDDATE)
          OR (:VIN_METHOD = 0 AND D.FSTATE = 0 AND D.FUSER = :VIN_USERID))
      INTO :VID, :VTYPEid, :VTYPEname, :VSTATEID, :VSTATENAME, :VDATE,
:VTIME, :VPERSONID, :VPERSON, :VUSER, :VSUMM
   DO
   BEGIN
    SUSPEND;
   END
```

Документ «Приход товара» предназначен для регистрации поступления товаров в магазин от поставщиков. Форма документа представлена на рисунках 14-15:

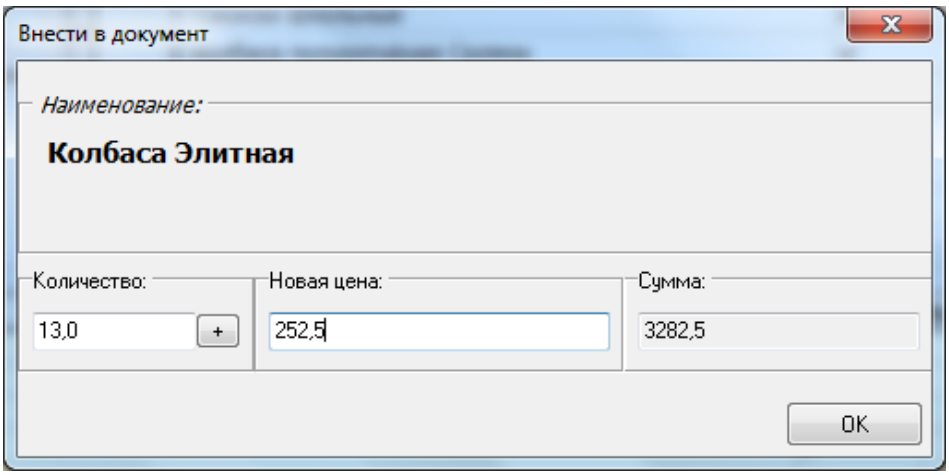


Рисунок 14 – Экранная форма добавления товара в документ

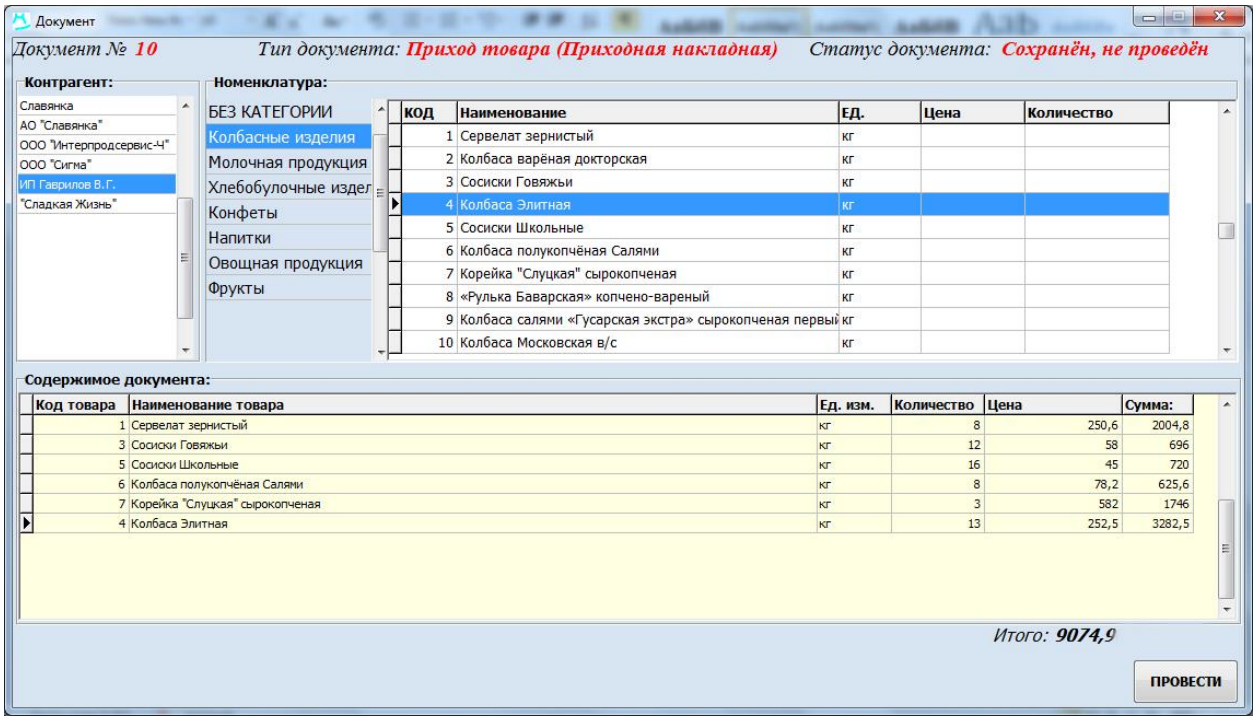


Рисунок 15 – Экранная форма документа «Приход товара»

Далее был создан документ «Продажа товара», который предназначен для фиксации проданных товаров клиентам магазина. На рисунках 16-17 показана экранная форма документа «Продажа товара»:

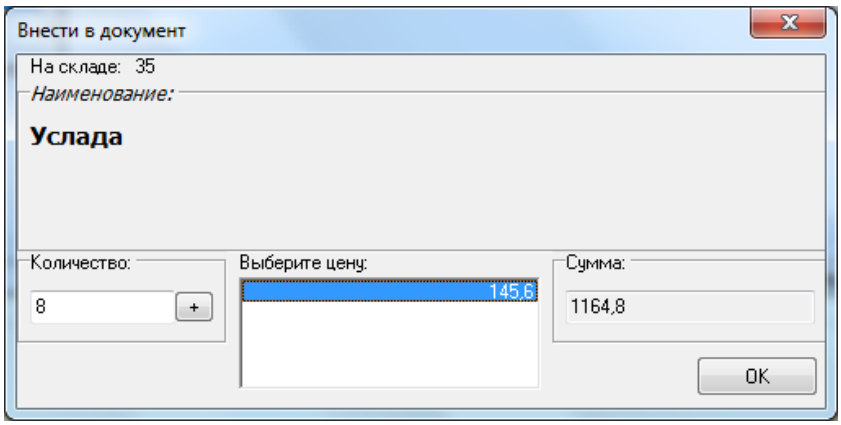


Рисунок 16 – Экранная форма добавления товара в документ

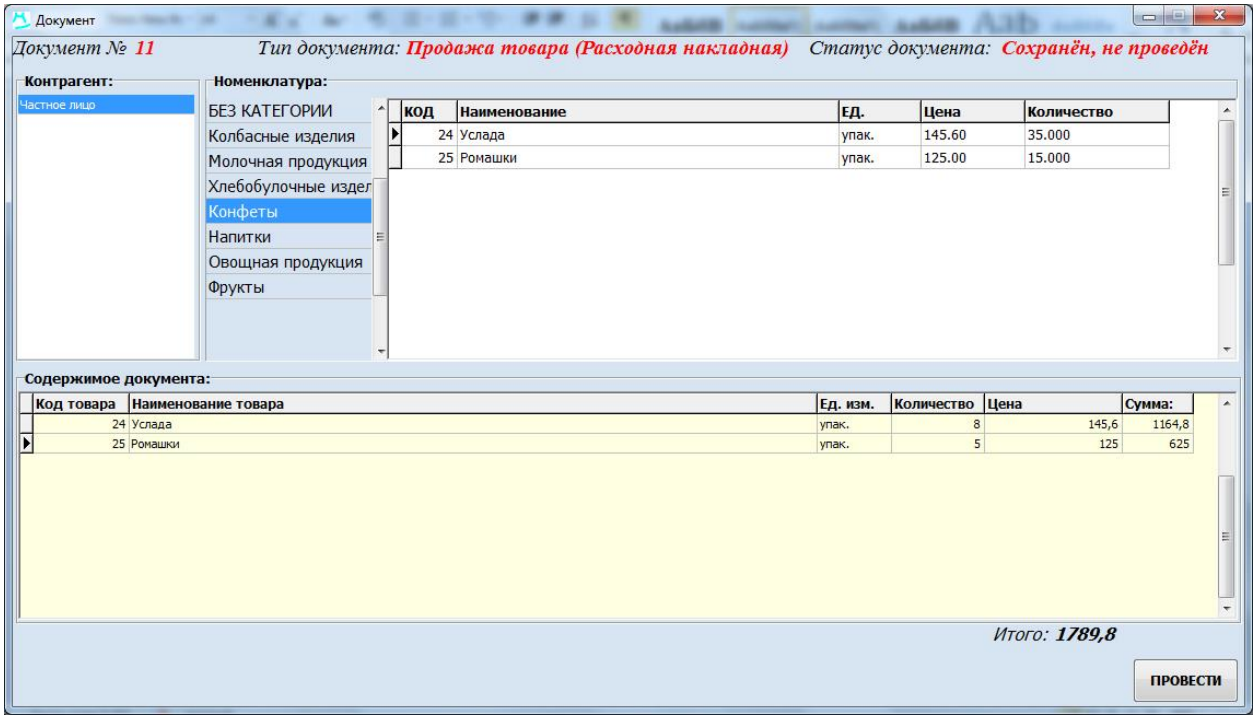


Рисунок 17 –Экранная форма документа «Продажа товара»

Документ «Списание товара» предназначен для регистрации списания товаров в магазине. Форма документа представлена на рисунках18-19:

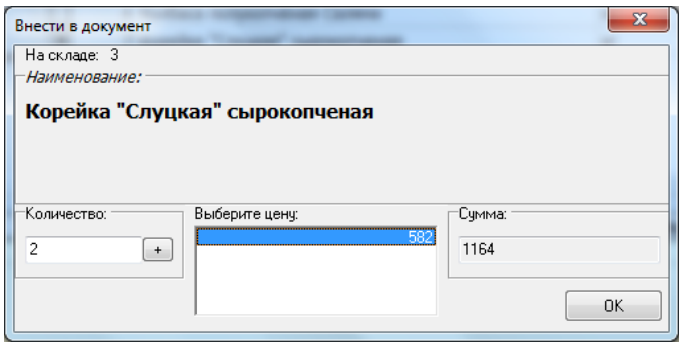


Рисунок 18 – Экранная форма добавления товара в документ

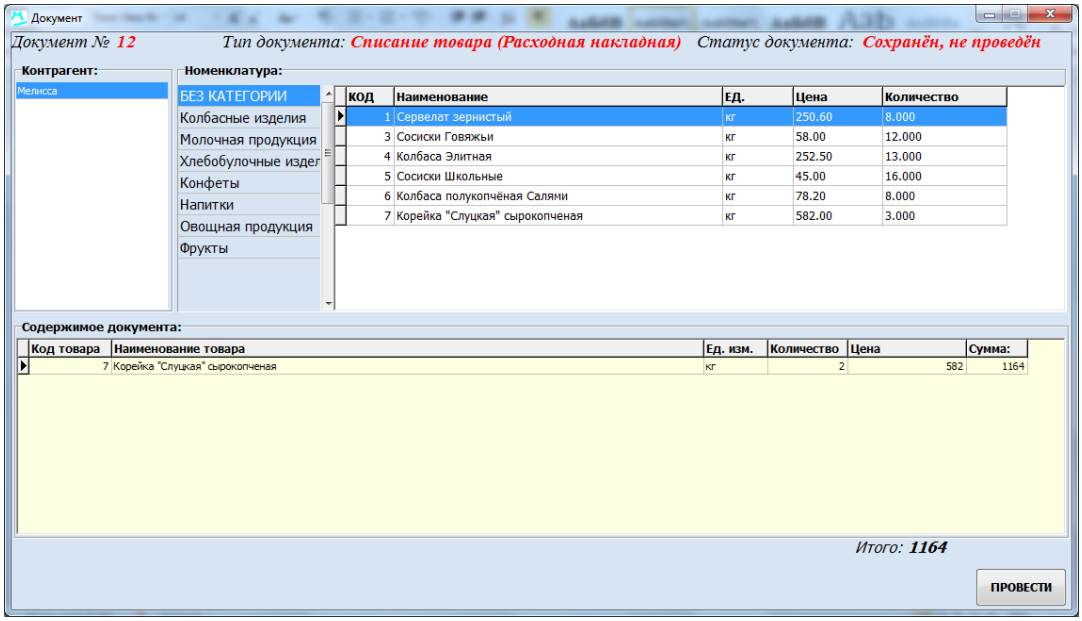


Рисунок 19 – Экранная форма документа «Списание товара»

Результатную информацию можно будет выгружать из информационной системы учета товаров в Microsoft Excel. Данные из базы данных будут экспортироваться в Microsoft Excel и после этого выгруженные данные можно будет импортировать в 1С:Бухгалтерию 8. Данная операция необходима для ведения бухгалтерской отчетности в 1С:Бухгалтерия 8. Для реализации этой операции необходимо использовать следующие компоненты: ExcelApplication, ExcelWorkbook, ExcelWorksheet, ExcelOLEObject.

3.1.5.2 Характеристика результатных документов

Результатными документами в информационной системе являются отчеты, предоставляющие возможность отслеживать и анализировать информацию о товарах, находящихся в магазине, о приходе и расходе товаров.

На рисунке 20 показан отчёт по прайс-листу магазина:

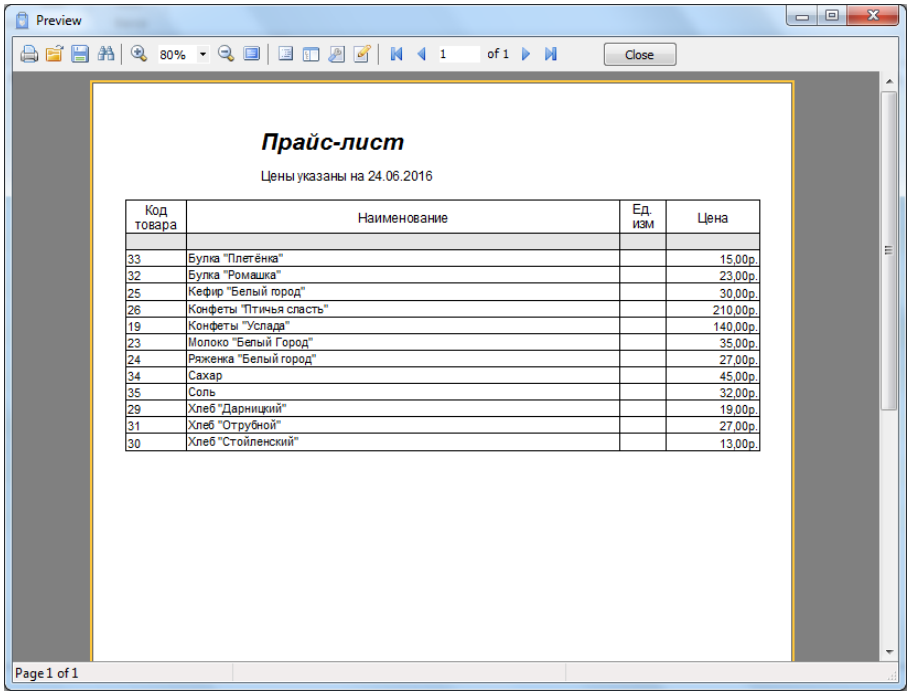
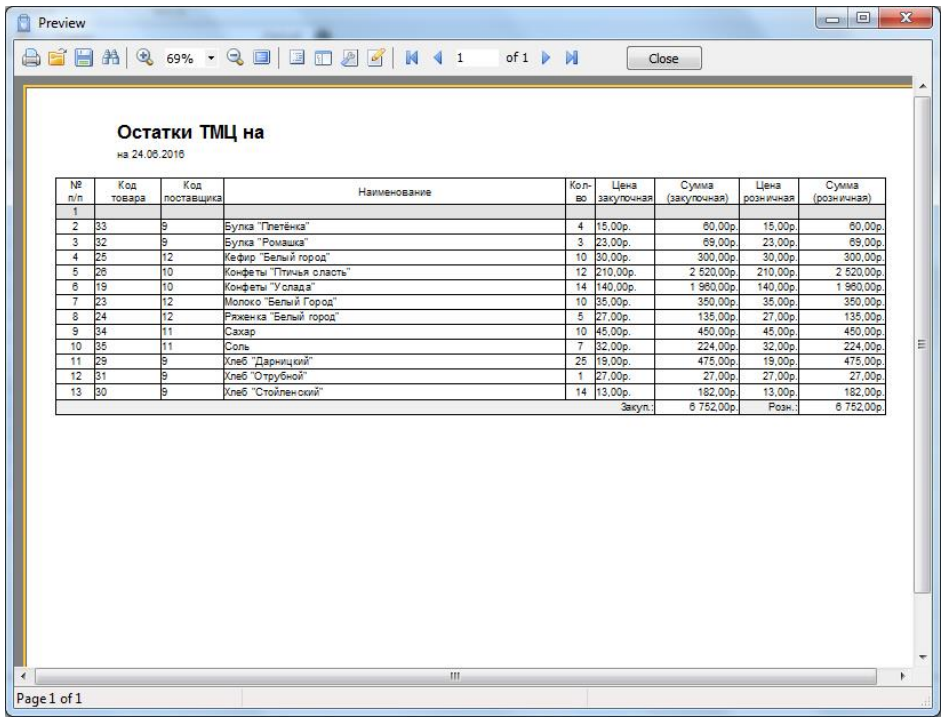


Рисунок 20 – Сформированный отчет «Прайс-лист»

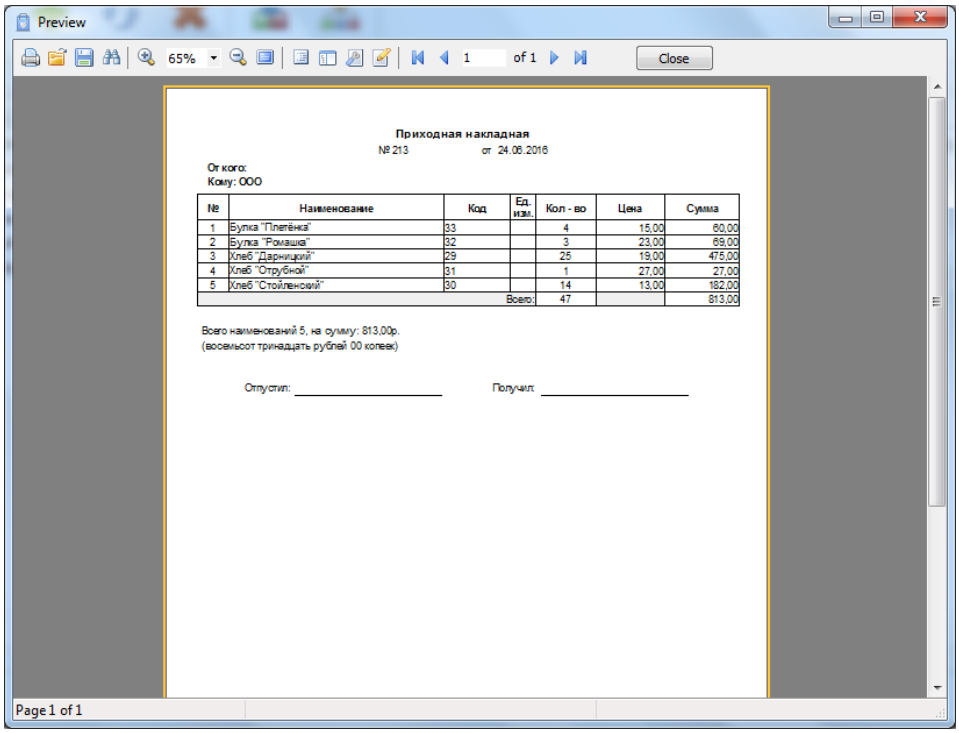
На рисунке 21 продемонстрирован сформированный отчет по остаткам продукции в магазине:



№ п/п	Код товара	Код поставщика	Наименование	Кол-во	Цена закупочная	Сумма (закупочная)	Цена розничная	Сумма (розничная)
1								
2	33	9	Булка "Плетёнка"	4	15,00р.	60,00р.	15,00р.	60,00р.
3	32	9	Булка "Ромашка"	3	23,00р.	69,00р.	23,00р.	69,00р.
4	25	12	Кефир "Белый город"	10	30,00р.	300,00р.	30,00р.	300,00р.
5	26	10	Конфеты "Птичья сласть"	12	210,00р.	2 520,00р.	210,00р.	2 520,00р.
6	19	10	Конфеты "Услада"	14	140,00р.	1 960,00р.	140,00р.	1 960,00р.
7	23	12	Молоко "Белый Город"	10	35,00р.	350,00р.	35,00р.	350,00р.
8	24	12	Ряженка "Белый город"	5	27,00р.	135,00р.	27,00р.	135,00р.
9	34	11	Сахар	10	45,00р.	450,00р.	45,00р.	450,00р.
10	35	11	Соль	7	32,00р.	224,00р.	32,00р.	224,00р.
11	29	9	Хлеб "Дарницкий"	25	19,00р.	475,00р.	19,00р.	475,00р.
12	31	9	Хлеб "Отрубной"	1	27,00р.	27,00р.	27,00р.	27,00р.
13	30	9	Хлеб "Стойленский"	14	13,00р.	182,00р.	13,00р.	182,00р.
					Закуп.:	6 752,00р.	Розн.:	6 752,00р.

Рисунок 21 – Сформированный отчет «Остаток ТМЦ»

Отчет по приходу товара показан на рисунке 22 и предназначен для отображения информации о поступивших товарах от поставщиков:



№	Наименование	Код	Ед. изм.	Кол - во	Цена	Сумма
1	Булка "Плетёнка"	33		4	15,00	60,00
2	Булка "Ромашка"	32		3	23,00	69,00
3	Хлеб "Дарницкий"	29		25	19,00	475,00
4	Хлеб "Отрубной"	31		1	27,00	27,00
5	Хлеб "Стойленский"	30		14	13,00	182,00
			Всего:	47		813,00

Рисунок 22 – Сформированный отчет по приходу товаров

В программе имеется возможность формирования отчетов с использованием фильтров, например, можно сформировать отчет за некоторый период времени и при этом необходимо ввести даты, как это показано на рисунках 23-24:

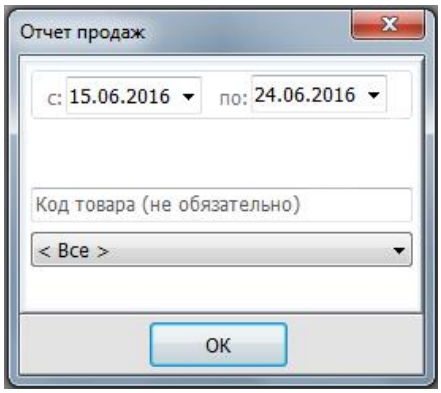
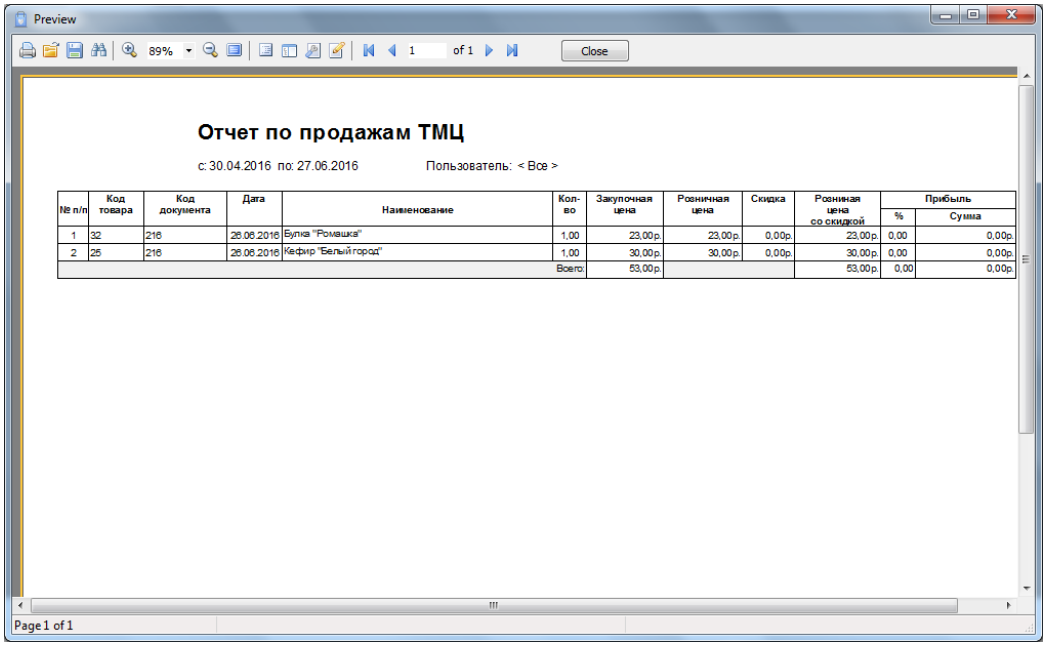


Рисунок 23 – Форма ввода даты для формирования отчета



№ п/п	Код товара	Код документа	Дата	Наименование	Кол-во	Закупочная цена	Розничная цена	Скидка	Розничная цена со скидкой	Прибыль	
										%	Сумма
1	32	216	26.06.2016	Булка "Ромашка"	1,00	23,00р.	23,00р.	0,00р.	23,00р.	0,00	0,00р.
2	25	216	26.06.2016	Кефир "Белый город"	1,00	30,00р.	30,00р.	0,00р.	30,00р.	0,00	0,00р.
					Всего:	53,00р.			53,00р.	0,00	0,00р.

Рисунок 24 – Сформированный отчет по продажам товара за период

На рисунках 25-26 представлена форма для формирования отчета по кассе. Для этого необходимо выбрать клиента из выпадающего списка и нажать на кнопку «ОК»:

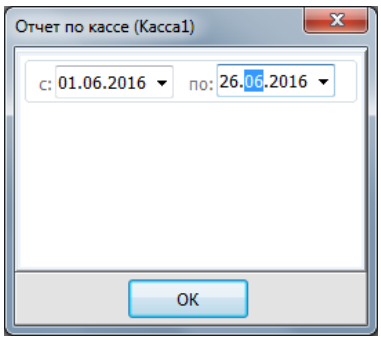


Рисунок 25 – Форма выбора даты для формирования отчета

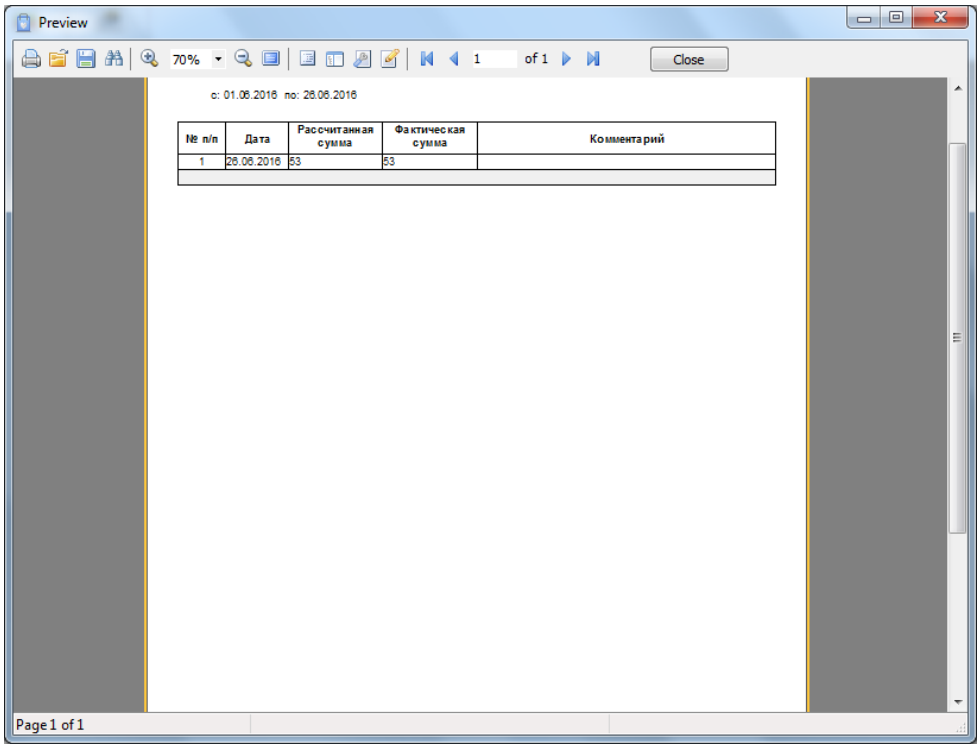


Рисунок 26 – Сформированный отчет по выбранной дате

3.2 Программное обеспечение задачи (комплекса задач, АРМ)

3.2.1 Общие положения (дерево функций и сценарий диалога)

Модель дерева функций относится к функциональному представлению и предназначена для описания иерархической структуры

функций (включая статические связи между ними) бизнес-процессов предприятия.

Функция - это задача, операция или действие, которые выполняются над объектом для достижения одной или нескольких целей.

Дерево функций информационной системы учета товаров показано на рисунке 27:

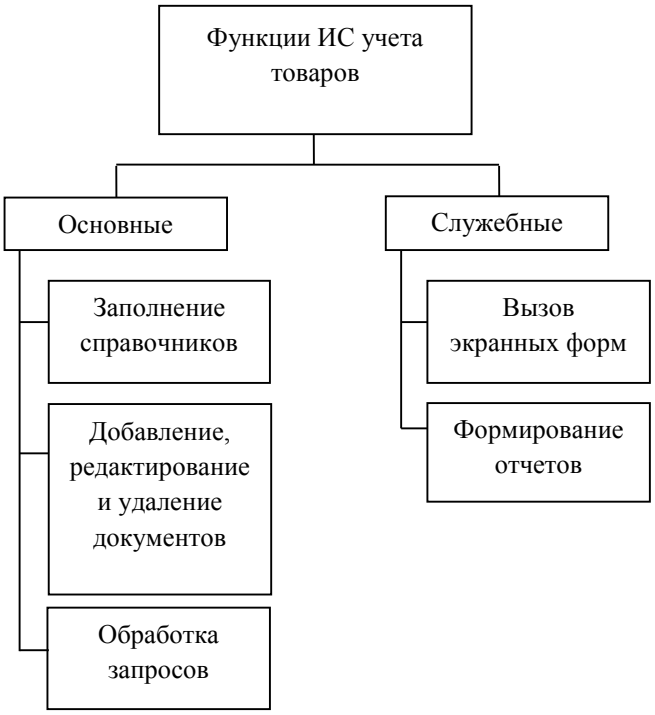


Рисунок 27 – Дерево функций

Были выделены следующие подмножества функций:

− основные функции, к которым относятся: заполнение справочников, добавление, редактирование и удаление документов и обработка запросов;

− служебные функции, к которым относятся: вызов экранных форм и формирование отчетов.

Так как при решении задачи используется технология обработки информации в режиме диалога, то взаимодействие пользователя с программой можно представить в виде схемы диалога, представленной на рисунке 28:

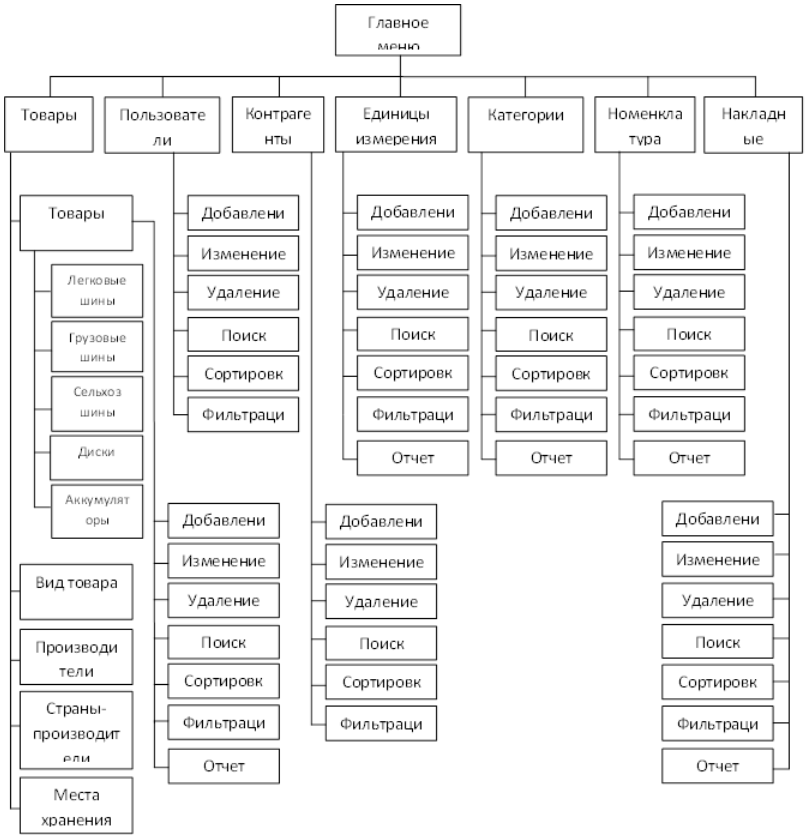


Рисунок 28 – Схема диалога взаимодействия пользователя с системой

Реализованный в программе диалог относится к типу меню ориентированных диалогов. Схема диалога - это общая конструкция диалога, то есть необходимая последовательность обмена данными между пользователем и системой. Главное меню, которое инициирует задачу, располагается на верхнем уровне схемы. Дальнейшая работа с автоматизированной системой производится по одному из разветвлений программы. Выбор разветвления зависит от количества вариантов ответа пользователя на запросы или от возможных реакций на конкретные сообщения пользователя [21].

3.2.2 Структурная схема пакета (дерево вызова процедур и программ)

Для наглядности представления работы разрабатываемой информационной системы необходимо построить структурную схему, которая содержит общий алгоритм переходов между формами системы.

На рисунке 29 представлена структурная схема автоматизированной системы. Главная форма состоит из ряда кнопок, которые, в свою очередь, связаны с базой данных по определенному алгоритму.

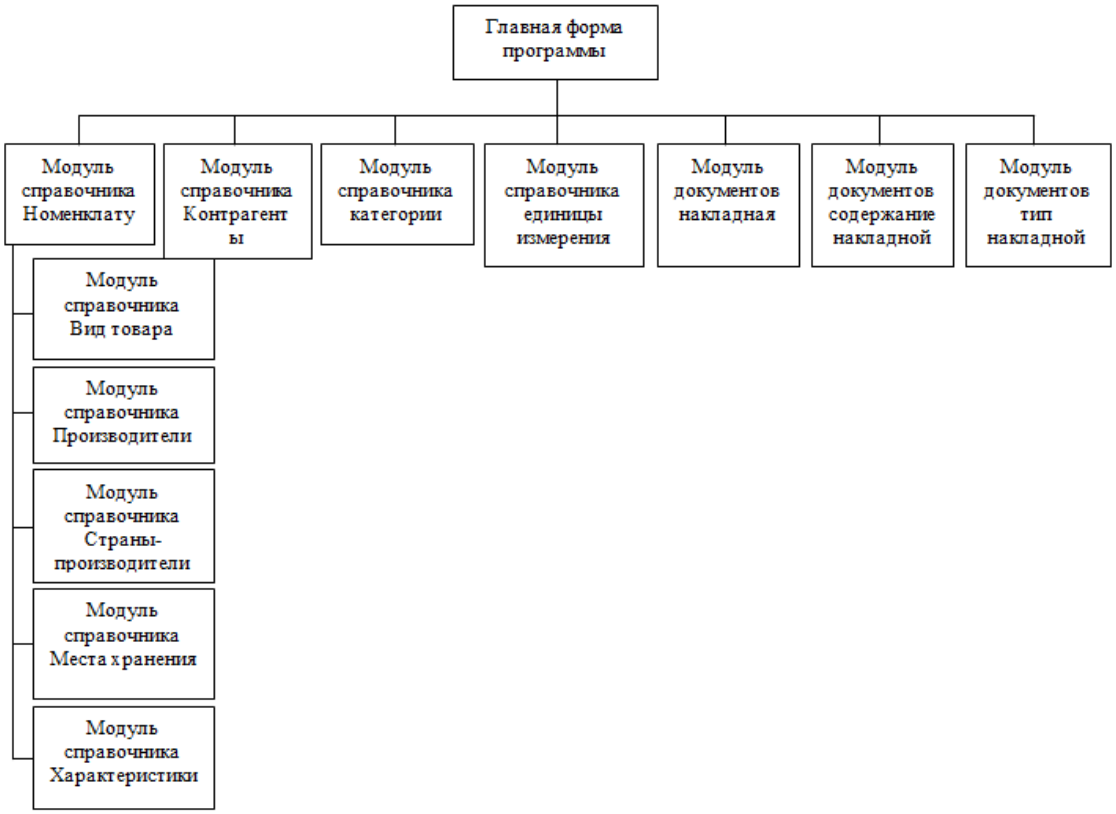


Рисунок 29 – Структурная схема автоматизированной системы

Структурная схема информационной системы содержит программные модули различных классов:

- модули, которые выполняют служебные функции;
- управляющие модули, которые предназначены для загрузки меню и передачи управления другому модулю;

− модули, которые связаны с вводом, хранением, обработкой и выдачей информации.

3.2.3 Описание программных модулей

Программный модуль – это «контейнер» для размещения текстов процедур и функций, вызываемых системой во время исполнения в определенные моменты времени.

В таблице 4 представлено описание программных модулей, которые схематично изображены на рисунке 13:

Таблица 4 – Описание программных модулей

№ п/п	Название модуля	Функции модуля
1	2	3
1	Модуль справочника номенклатура	Просмотр, добавление, редактирование и удаление, поиск, сортировка и фильтрация информации о номенклатуре, формирование отчета
2	Модуль справочника категории	Добавление, редактирование и удаление данных о категориях
3	Модуль справочника единицы измерения	Добавление, редактирование и удаление данных о единицах измерения
4	Модуль справочника контрагенты	Добавление, редактирование и удаление данных о контрагентах
5	Модуль справочника журнал документов	Добавление, редактирование и удаление данных о документах

Продолжение таблицы 4

1	2	3
7	Модуль документов Прайс-лист	Добавление, редактирование и удаление, поиск, сортировка и фильтрация информации о ценах на товары
8	Модуль документов Остатки в магазине	Добавление, редактирование и удаление, поиск, сортировка и фильтрация информации об остатках товаров
9	Модуль документов Отчёт продаж	Добавление, редактирование и удаление, поиск, сортировка и фильтрация информации о проданных товарах
10	Модуль документов Отчёт по кассе	Добавление, редактирование и удаление, поиск, сортировка и фильтрация информации о кассе

Таким образом, были перечислены и описаны все модули программы, которые выполняют определенные функции

3.3 Технологическое обеспечение задачи (комплекса задач, АРМ)

3.3.1 Организация технологии сбора, передачи, обработки и выдачи информации

Технологическое обеспечение отражает организацию технологии сбора, передачи, обработки и выдачи данных и описывает последовательность действий от получения первичной информации и до составления результатных документов [18].

Технологический процесс обработки данных - это упорядоченная последовательность взаимосвязанных действий по обработке информации до получения требуемого пользователю результата [18].

На технологический процесс обработки данных большое влияние оказывают характер выполняемых задач, применяемые технические средства, системы контроля, количество пользователей и иные аспекты.

Технологический процесс обработки информации состоит из таких операций, как:

− сбор данных, представляющий собой процесс регистрации, фиксации, записи информации о связях, событиях, объектах, действиях, признаках;

− обработка данных, к которой относятся следующие действия: расчеты, поиск, фильтрация, выборка, объединение, сортировка, слияние и другие;

− генерирование информации, которое заключается в организации, переорганизации и преобразовании данных в форму, необходимую пользователям;

− хранение данных и информации, включающее размещение, накопление, копирование и выработку данных и информации с целью их последующего применения;

− передача информации и данных, которая представляет собой распространение информации и данных между всеми пользователями при помощи средств и систем коммуникаций и путем перемещения данных от источника (отправителя) к приемнику (получателю) [18].

Технологический процесс начинается с загрузки операционной системы. После загрузки операционной системы необходимо запустить программу информационной системы учета товаров Project1.exe.

После запуска программы выводится информационное окно и активизируется система меню. Работа с программой осуществляется в диалоговом режиме. Главное меню программы содержит такие пункты, как: Журнал документов, Справочники, Отчёты, Настройки.

Результатом действия каждого пункта меню является активизация соответствующей формы – макета ввода/вывода данных по определенным условиям: либо необходимости ввести условия выбора данных вручную, либо выбрать из предлагаемого набора. Всю информацию можно вводить заново, редактировать, удалять, производить поиск, сортировку и

фильтрацию. Это касается как нормативно-справочной информации, так и первичной. Кроме того, предусмотрена возможность вывода на бумажный носитель (на печать) всех электронных документов [27].

3.3.2 Схема технологического процесса сбора, передачи, обработки и выдачи информации

На рисунке 30 представлен фрагмент схемы технологического процесса сбора, передачи, обработки и выдачи информации на примере работы с документом «Заказ поставщику»:

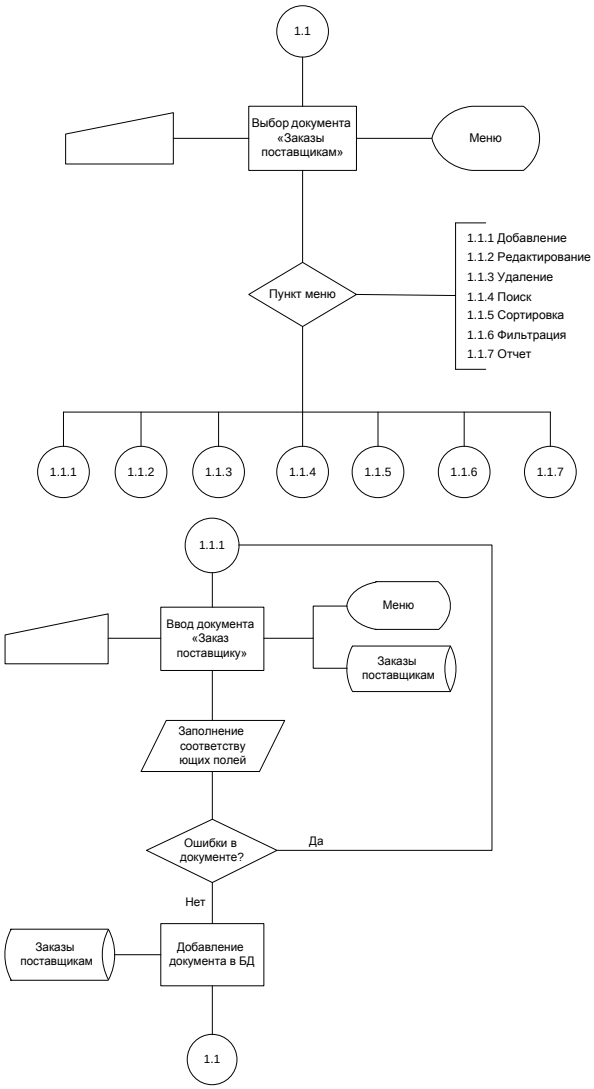


Рисунок 30 – Фрагмент схемы технологического процесса

При работе с остальными справочниками и документами, используемыми в информационной системе, технологические процессы сбора, передачи, обработки и выдачи информации осуществляются аналогичным образом.

3.4 Описание контрольного примера реализации проекта

Протестируем разработанную информационную систему учета. При открытии программы появляется главное окно, показанное на рисунке 31:

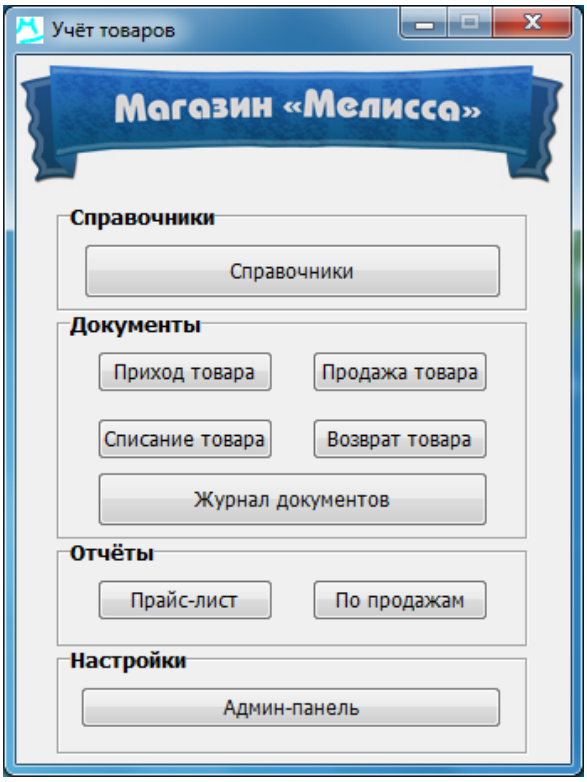


Рисунок 31 – Главное окно программы

В главном окне программы находится меню, состоящее из следующих разделов:

- справочники;
- документы;
- отчёты;

– настройки.

Для описания контрольного примера работы программы, возьмём раздел «Справочники». На рисунках 32, 33 и 34 представлено добавление информации о новой номенклатуре:

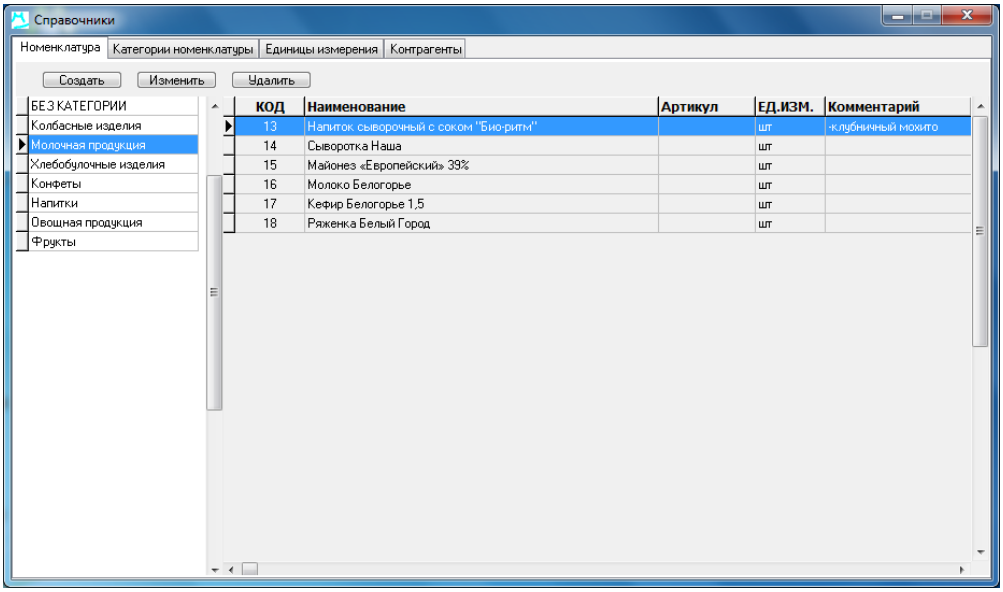


Рисунок 32 – Окно выбора операции

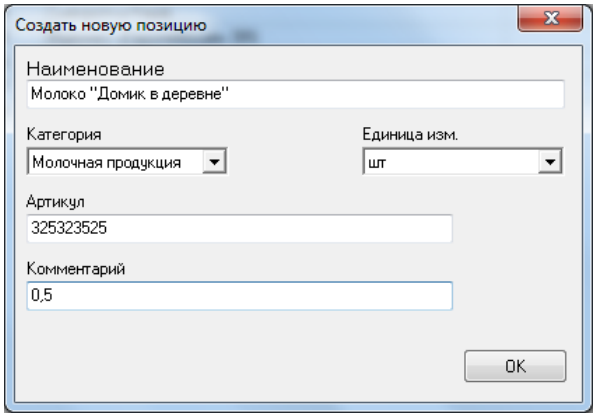


Рисунок 33 – Окно ввода информации о новом элементе

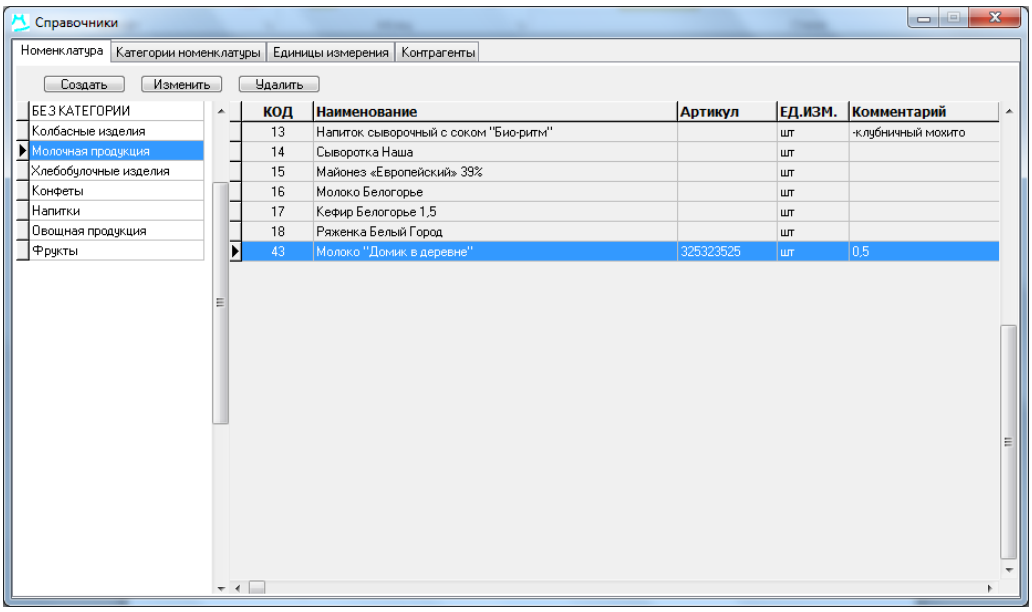


Рисунок 34 – Результат добавления нового элемента

Также для описания контрольного примера работы программы, возьмём раздел «Настройки», на рисунке 35 показан результат выбора кассира и его кассы:

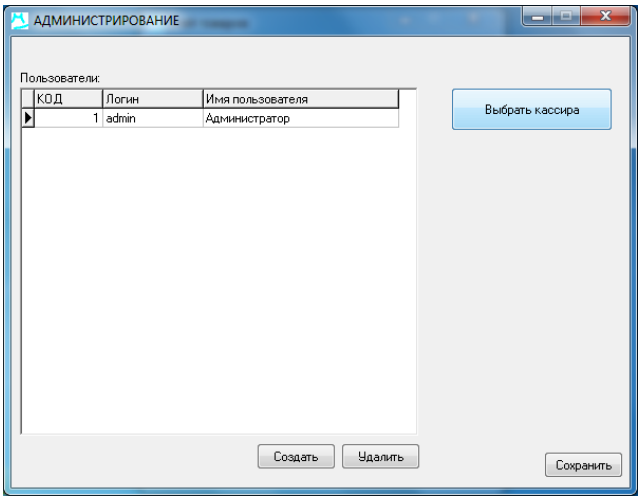


Рисунок 35 – Установка кассира

3.5 Целесообразность разработки с экономической точки зрения

Для данного магазина, ИП Кореньков И. Н., характерен большой объем документов и информации по учету товаров, тем самым, автоматизация позволит увеличить производительность труда, благодаря:

- ускорению обработки документов и информации;
- уменьшению трудозатрат;
- повышению достоверности данных;
- сокращению количества ручных операций и числа ошибок;
- повышению надежности;
- повышению качества выполняемых задач.

Информационная система позволит данной организации вести актуальную базу данных и проводить анализ деятельности по ведению учета товаров на основании различных отчетов.

Использование разработанного программного продукта позволит сократить затраты предприятия, отказавшись от покупки дорогостоящего оборудования, программного и аппаратного обеспечения.

В итоге разработка и внедрение информационной системы учета товаров является целесообразной.

3.6 Расчёт экономической эффективности

При проектировании данной информационной системы определяем этапы разработки (в скобках указываем продолжительность этапа в часах):

1) Знакомство с темой, анализ технического задания ТЗ (12)
2) Подбор и изучение справочной литературы (40)
3) Разработка алгоритма и структуры программы (25)
4) Программирование (50)
5) Тестирование программного обеспечения (ПО) (60)
6) Разработка инструкции и описаний ПО (40)

Для осуществления работ по каждому этапу определяем состав специалистов: специалист по информационным системам и программист.

Затраты на разработку системы подразделяем на капитальные или единовременные (разработка самой системы) и эксплуатационные или текущие. Расчет выполним по отдельным статьям:

1) Прямые материальные затраты
2) Фонд оплаты труда
3) Отчисления на социальные нужды
4) Амортизационные отчисления
5) Накладные расходы
6) Прочие расходы.

Расчет материальных затрат приведен в таблице 5.

Таблица 5 – Расчет прямых материальных затрат

Материалы	Ед. измер.	Кол- во	Цена за ед. (руб.)	Стоимость (руб.)
Бумага	лист	500	0.3	150
Картридж для принтера	шт.	1	360	360
Диск CD-R	шт.	1	45	45
Итого				555
Расходы на энергию	кв/ч	50	2.47	123.5
Итого				**678.5**

Прямые материальные затраты составили 678.5 рублей.

Для осуществления проекта необходимы единовременные или капитальные затраты, а именно покупка ПК на базе Intel Core i5в количестве 1 шт., по цене 25000 рублей.

Для расчета фонда оплаты труда необходимо рассчитать заработную плату разработчиков ИС, составить баланс рабочего времени. Данные сведем в таблице 6.

Таблица 6 – Баланс рабочего времени

П/п	Наименование показателей	ИТР
1	Число календарных дней в году	365
2	Число выходных и нерабочих дней в году	112
3	Число рабочих дней в году	253
4	Невыходы на работу:	
	А) по болезни	0
	Б) очередной отпуск	0
5	Фактическое число рабочих дней в году	253
6	Продолжительность рабочего дня	8
7	Годовой фонд рабочего времени (час.)	2024

Часовую ставку заработной платы (Чс) определяем по формуле (3.1):

$$Чс = (З * п * к) / Ф, \text{ где} \quad (3.1)$$

- Чс- месячная зарплата, руб.;
- П- число месяцев в году, исключая отпуск;
- К- коэффициент, учитывающий премии из фонда зарплаты;
- Ф - фактический годовой фонд рабочего времени, час.

Примем среднемесячную зарплату инженера – программиста и разработчика ИС равной: З = 11500 руб.,

При п = 11, а Ф = 2024 час., получим Чс = 71.8 руб./час.

Размер основной заработной платы определяем исходя из времени, затрачиваемого на выполнение работ и стоимости часа работы исполнителя.

Основная заработная плата определяется по графику основных этапов работ. Дополнительная заработная плата может составить до 15% от основной.

Расчет фонда оплаты труда приведен в таблице 7.

Таблица 7 – Расчет фонда оплаты труда

Этапы разработки	Время (час)	Часовая ставка (руб.)	Сумма (руб.)
Анализ ТЗ	12	71.8	861.6
Подбор, изучение литературы	40	71.8	2872
Разработка алгоритма и структуры программы	25	71.8	1795
Программирование	50	71.8	3590
Тестирование ПО	60	71.8	4308
Разработка инструкций ПО	40	71.8	2872
Основная заработная плата(итого)			16298.6
Дополнительная заработная плата			2444.79
Коэффициент			3259.72
ИТОГО			22003.11

Размер отчислений на социальные нужды определяется исходя из размера фонда оплаты труда. Расчет приведен в таблице 8.

Таблица 8 – Отчисления на социальные нужды

Отчисления	Доля от фонда оплаты труда (%)	Сумма (руб.)
В пенсионный фонд	20.6	4532.6
В фонд занятости	1.7	374
Медицинское страхование	2.6	572
Социальное страхование	2.9	638
ИТОГО	27.8	6116.8

Величина накладных расходов определяется в размере 80% от основной заработной платы и составит: НР= 0.8*16298.6= 13038.8

Амортизационные отчисления на оргтехнику допустимо производить из расчета 50% в год. По ПК, следовательно, амортизационные расходы составят: 25000* 0.5 (50%)= 12500 руб. в год

Полную смету затрат на разработку системы приведем в таблице 9.

Таблица 9 – Расчет затрат на разработку ИС

Статья расхода	Сумма (руб.)
Фонд оплаты труда	22003.11
Отчисления на социальные нужды	6116.8
Материальные затраты	678.5
Амортизационные отчисления	12500
Прочие расходы	-
Накладные расходы	13038.8
Итого	54337.21

Расчет ежемесячных затрат на эксплуатацию системы приведем в таблице 10.

Таблица 10 – Расчет затрат на эксплуатацию системы

Статья расхода	Сумма (руб.)
Зарплата администратора БД	2100
Отчисления на социальные нужды	583.8
Затраты на электроэнергию	150
ИТОГО	2883.8

Для оценки инвестиционного проекта информационной системы необходимо рассчитать планируемые поступления денежных средств от реализации товаров по месяцам. Расчет приведем в таблице 11.

Таблица 11 – Расчет планируемых поступлений

Месяц	Увеличение объема продаж, %	Сумма, руб.	Затраты, руб.	CF, руб.	$cF_t^{(\Sigma)}$ руб.
1	0	0	2883,8	-2883,8	2883,8
2	1	4500	2883,8	1616,2	4500
3	1	4500	2883,8	1616,2	6116,2
4	2	9000	2883,8	6116,2	12232,4
5	2	9000	2883,8	6116,2	18348,6
6	2	9000	2883,8	6116,2	24464,8
7	3	13500	2883,8	10616,2	35081
8	3	13500	2883,8	10616,2	45697,2
9	3	13500	2883,8	10616,2	56313,4
10	3	13500	2883,8	10616,2	66929,6
11	3	13500	2883,8	10616,2	77545,8
12	3	13500	2883,8	10616,2	88162

Проведем оценку инвестиционного проекта информационной системы по формуле (3.2):

$$NPV = \sum_{t=1}^{n} \frac{cF_1}{k+q} - I_0, \text{ где} \qquad (3.2)$$

NPV – чистая текущая стоимость инвестиций;

CF – поступление денежных средств в конце t- ого периода;

q- банковская ставка;

I – стоимость реализации инвестиционного проекта (инвестиции).

NPV = 88162 / (1 + 0.12) – 54337.21= 24378.8

$$PI = \left[\sum_{t=1}^{n} \frac{cF_t}{(1+k)^t}\right] \Big/ \left[\sum_{t=1}^{n} \frac{I_t}{(1+k)^t}\right], \text{ где} \qquad (3.3)$$

PI – рентабельность инвестиций.

PI = (88162 /(1+ 0.12))/ 54337.21= 1.44

$$PP = \frac{I_0}{cF_t^{(\Sigma)}}, \text{ где} \qquad (3.4)$$

PP – период окупаемости (лет);

I_0 – первоначальные инвестиции;

$cF_t^{(\Sigma)}$ – годичная сумма денежных поступлений от реализации инвестированного проекта.

PP = 88162/ (54337.21/ (1 + 0.12)) = 1.8

Затраты на разработку и внедрение системы окупятся спустя 4 месяца со дня введения системы в действие, чистая текущая стоимость инвестиций имеет положительное значение, рентабельность составляет 144 %. Внедрение системы позволяет увеличить количество продаж, уменьшается время обслуживания посетителей.

3.7 SWOT анализ разработки

SWOT – аббревиатура английских слов Strength (сила), Weakness (слабость), Opportunities (возможности), Threats (угрозы) [10].

Этапы SWOT-анализа:

1) Выявление сильных и слабых сторон разработки, её возможностей и угроз;

2) Формирование и анализ SWOT-матрицы;

3) Выработка заключения о перспективности разработки [10].

Проанализировать возможности развития информационной системы учета товаров поможет SWOT-матрица, в которой приводится влияние каждого преимущества или недостатка разработки на существующие возможности развития и возникающие при этом угрозы.

В таблице 12 приведена SWOT-матрица разработки:

Таблица 12 - SWOT-матрица

Сильные стороны	Возможности		Угрозы			Итого
	Расширение круга потребителей	Совершенствование разработки	Появление новых конкурентов	Быстрое моральное устаревание	Временное увеличение нагрузки на сотрудников при внедрении системы	
Низкая стоимость разработки	++	0	++	0	0	+4
Многофункциональность	++	++	+	+	++	+8
Обеспечение сопровождения	+	+	+	+	+	+5
Итого	+5	+3	+4	+2	+3	+17
Слабые стороны	Возможности		Угрозы			Итого
	Расширение круга потребителей	Совершенствование разработки	Появление новых конкурентов	Быстрое моральное устаревание	Временное увеличение нагрузки на сотрудников при внедрении системы	
Недостаточное финансирование	– –	– –	–	–	0	–6
Нехватка квалифицированных кадров	– –	– –	0	–	0	–7
Итого	–4	–4	–1	–2	0	–13
Общий итог	+1	–1	+3	0	+3	+4

Проанализировав данную SWOT-матрицу, можно сделать следующие выводы.

Самой сильной стороной разработки является ее многофункциональность. В дальнейшем необходимо обращать особое внимание на обеспечение и расширение этой стороны разработки.

Все выделенные слабые стороны разработки являются достаточно важными и необходимо принять меры по их устранению.

Из рассмотренных возможностей более реальной считается возможность расширения круга возможных потребителей, хотя данная возможность при существующих слабостях весьма проблематична. Совершенствование же разработки при недостатке финансирования и нехватке кадров вообще невозможно. Наиболее оптимальным решением этой проблемы является увеличение числа квалифицированных кадров.

Наиболее опасной угрозой представляется быстрое устаревание разработки, но при сложившихся условиях это не столь существенная угроза. Появление конкурентов представляется маловероятным вследствие наличия сильных сторон разработки.

Заключение о перспективности разработки можно сделать следующее, что в настоящее время имеются некоторые трудности в дальнейшем развитии разработки, но, тем не менее, разработка является достаточно перспективной, об этом говорит оценка «+4». Первоочередной задачей для совершенствования программного продукта является повышение квалификации кадров.

Выводы по третьей главе

В рассмотренной главе была построена новая информационная модель «КАК ДОЛЖНО БЫТЬ» и по сравнению с моделью «КАК ЕСТЬ» показывает некоторые изменения, которые будут совершены в результате разработки автоматизированной системы. Были описаны основные классификаторы и системы кодирования. С помощью CASE-средств была

спроектирована логическая и физическая модели базы данных. Были описаны инфологическая и даталогическая модель базы данных. Приведена характеристика входной и результатной информации. А также представлено программное и технологическое обеспечение задачи, был описан контрольный пример реализации проекта.

В конце главы была обоснована целесообразность разработки с экономической точки зрения и проведён SWOT-анализ, в результате которого были выявлены сильные и слабые стороны разработки, а также возможности ее развития и угрозы

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы были решены все поставленные задачи. Результатом проделанной работы является разработанная информационная система учета товаров, которая выполняет функции сбора, хранения и обработки информации.

При выполнении выпускной квалификационной работы был выполнен весь необходимый перечень и объем работ. Приобретены навыки практического решения информационных задач в качестве разработчика информационной системы.

В результате выполнения выпускной квалификационной работы были решены следующие задачи: собран материал по предметной области и произведен его анализ, обоснована необходимость разработки системы автоматизации и определены требования к информационной системе, проанализирован и обоснован выбор инструментальных средств, спроектирована и реализована, а также протестирована информационная система, рассчитана себестоимость системы автоматизации учета.

В первой главе была описана технико-экономическая характеристика предприятия, экономическая сущность задачи, обоснована необходимость и цель вычислительной техники для решения задачи, проанализированы существующие разработки для решения поставленной задачи.

Во второй главе было выполнено обоснование проектных решений по техническому, информационному, программному, технологическому обеспечению задачи, а также обоснование выбора программных средств.

В третьей главе было представлено информационное обеспечение, включающее новую информационную модель учета товаров «КАК ДОЛЖНО БЫТЬ», которая была спроектирована при помощи CASE-средства AllFusion Process Modeler 7. Были описаны используемые классификаторы, характеристика первичных и результатных документов, были спроектированы логическая и физическая модель базы данных с

использованием CASE-средства AllFusion ERwin Data Modeler 7. Описано программное и технологическое обеспечение, рассмотрен контрольный пример реализации задачи.

В качестве инструментальных средств, используемых при создании и проектировании информационной системы, применялись клиент-серверная СУБД Firebird и утилита IBExpert для разработки базы данных, программный продукт Borland C++ Builder 6.0, предназначенный для разработки пользовательского интерфейса информационной системы.

Разработанная информационная система учета товаров позволяет повысить оперативность и производительность труда сотрудников магазина. Разработанная система позволяет синхронизировать используемые данные и сократить бумажные архивы, а также предоставляет полную картину о состоянии товаров в магазине.

Разработанное программное средство имеет удобный и интуитивно понятный интерфейс взаимодействия с пользователем, позволяет повысить качество обработки информации, ее достоверность и надежность. В системе предусмотрена возможность формирования отчетов и использования справочников, что позволяет своевременно и оперативно выявлять необходимость в тех или иных видах товаров. Разработанная автоматизированная система соответствует требованиям, предъявляемым к современным программным продуктам.

В итоге разработанный программный продукт позволяет выполнять все задачи, необходимые для эффективного осуществления деятельности по учету товаров в магазине розничной торговли.

Предполагается дальнейшее развитие и совершенствование разработанной информационной автоматизированной системы учета.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Архангельский А. Я. C++Builder 6 Справочное пособие. Книга 2. Классы и компоненты. – М.: Бином-Пресс, 2004. – 528 с.
2. Балдин К.В., Уткин В.Б. Информационные системы в экономике. Учебное пособие. – Дашков и К, 2008. – 395 с.
3. Бобровский С. Технологии C++Builder. Разработка приложений для бизнеса. - Издательство Питер, 2007. – 1104 с.
4. Бондарь А. Практическое руководство для умных пользователей и начинающих разработчиков. - СПб.: «БХВ-Перербург», 2007. – 592 с.
5. Борри Х. Firebird: руководство разработчика баз данных: Пер. с англ. – СПб.: БХВ-Петербург, 2006. - 1104 с.
6. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. - М.: Финансы и статистика, 2000. – 352 с.
7. Вендров, А.М. Современные методы и средства проектирования информационных систем [Текстъ/ А.М. Вендров - М.: Финансы и статистика, 2008. – 65 с.
8. Голицына О.Л. Программное обеспечение / О. Л. Голицына, И. И. Попов, Т. Л. Партыка. – М.: Форум, 2013. – 448 с.
9. Гультяев А.К. Проектирование и дизайн пользовательского интерфейса. - СПб.: КОРОНАпринт, 2000. - 349 с.
10. Евдокимова В.В. Экономическая информатика. СПб.:Питер паблишинг, 2008. – 468 с.
11. Иванова Г.С. Объектно-ориентированное программирование: учебник для вузов / Г.С. Иванова, Т. Н. Ничушкина, Е. К. Пугачев. - М.: МГТУ им. Н.Э. Баумана, 2003. - 368 с.
12. Илюшечкин В.М. Основы использования и проектирования баз данных. - М.: «Издательство Юрайт» 2010. -213с.

13. Ипатова Э.Р. Методологии и технологии системного проектирования информационных систем / Э.Р. Ипатова, Ю.В. Ипатов. – М.: Флинта, 2008. – 256 с.

14. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2002. – 304 с.

15. Конноли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3–е издание. - М.: Издательский дом "Вильямс", 2003. - 1440 с.

16. Культин Н. Самоучитель C++Builder -СПб.: БХВ-Перербург, 2004.-203 с.

17. Лавров С.С. Программирование. Математические основы, средства, теория: учебное пособие. / С.С. Лавров. - СПб.: БХВ-Петербург, 2001. - 320 с.

18. Левчук Е.А. Технологии организации, хранения и обработки данных / Е.А. Левчук. – Минск: Вышэйшая школа, 2007. – 240 с.

19. Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite. – М.: Диалог-МИФИ, 2007. -432 с.

20. Маклаков С.В. BPwin, ERwin. CASE-средства разработки информационных систем. – М.: ДИАЛОГ-МИФИ, 2007. – 304 с.

21. Мандел Т. Разработка пользовательского интерфейса. – М.:ДМК Пресс, 2008. - 412 с.

22. Михелёв В.М. Базы данных и СУБД: Учебное пособие. Белгород: Издательство БелГУ, 2007. – 200 с.

23. Муромцев В.В. Проектирование информационных систем: Учебное пособие для студентов вузов заочной формы обучения по спец. 010502 "Прикладная информатика в экономике" / Муромцев В.В.; Рец.: В.А. Ломазов, С.И. Маторин; Федеральное агентство по образованию; Фак. КНИТ каф. прикладной информатики БелГУ; БелГУ. - Белгород: БелГУ, 2007. - 160 с.

24. Пахомов Б.И. C/C++ и Borland C++ Builder для начинающих. – СПб.: БХВ-Петербург, 2005. – 640 с.

25. Пахомов Б.И. C\C++ и Borland C++ Builder для студента. СПб.: БХВ-Петербург, 2006. – 448 с.

26. Петров В.Н. Информационные системы. – СПб.: Питер, 2002. – 688 с.

27. Послед Б.С. Borland C++ Builder 6. Разработка приложений баз данных – СПб.: ООО «ДиаСофтЮП», 2003 –320 с.

28. Смирнова Г.Н., Сорокин А.А. Проектирование экономических информационных систем. Учебное пособие. – М.: Высшая школа, 2002. – 428 с.

29. Титоренко Г.А. Автоматизированные информационные технологии в экономике: учебное пособие. / Г.А. Титоренко, 2003 г. – 245 с.

30. Устав ООО «АгроМир Белгород».

31. Федоров Н.В. Проектирование информационных систем на основе современных CASE-технологий: учебное пособие. / Н. В. Федоров.- МГИУ, 2008.-128 с.

32. Федорова Е.Н. Теоретические основы программирования: учебное пособие. / Е. Н. Федорова.- МГИУ, 2012.-214 с.

33. Фельдман Я.А. Создаем информационную систему / Я.А. Фельдман. – М.: Солон-Пресс, 2007. – 120 с.

34. Хомоненко А.Д. Базы данных: учебник для высших учебных заведений, 4-е издание дополненное и переработанное. – СПб.: Корона, 2004. – 736 с.

35. Хомоненко А.Д., Ададуров С.Е. Работа с базами данных в C++ Builder.-СПб.:БХВ-Петербург, 2006.-496 с.

36. Черемных С.В., Семенов И.О., Ручкин В.С. Структурный анализ систем: IDEF-технологии. – М: Финансы и статистика, 2001. – 208 с.

37. 1С:Предприятие 8. 1С-Логистика:Управление складом 3.0. [Электронный ресурс] Режим доступа: http://solutions.1c.ru/catalog/wms, свободный.

38. Основы проектирования реляционных баз данных. [Электронный ресурс] Режим доступа: http://www.intuit.ru/goods_store/ebooks/8322, свободный.

39. Ресурсы информационных систем. [Электронный ресурс] Режим доступа: http://www.economica-upravlenie.ru/content/view/204/, свободный.

40. Сертифицированные информационные системы [Электронный ресурс] Режим доступа: http://certsys.ru/products/index.php, свободный.

41. Фрегат-Склад. [Электронный ресурс] Режим доступа: http://www.audit-it.ru/allsoft/soft.php?id=111290, свободный.

42. Solvo.WMS. [Электронный ресурс] Режим доступа: http://www.solvo.ru/products/systems/wms/index.php, свободный.

ПРИЛОЖЕНИЯ

Приложение А

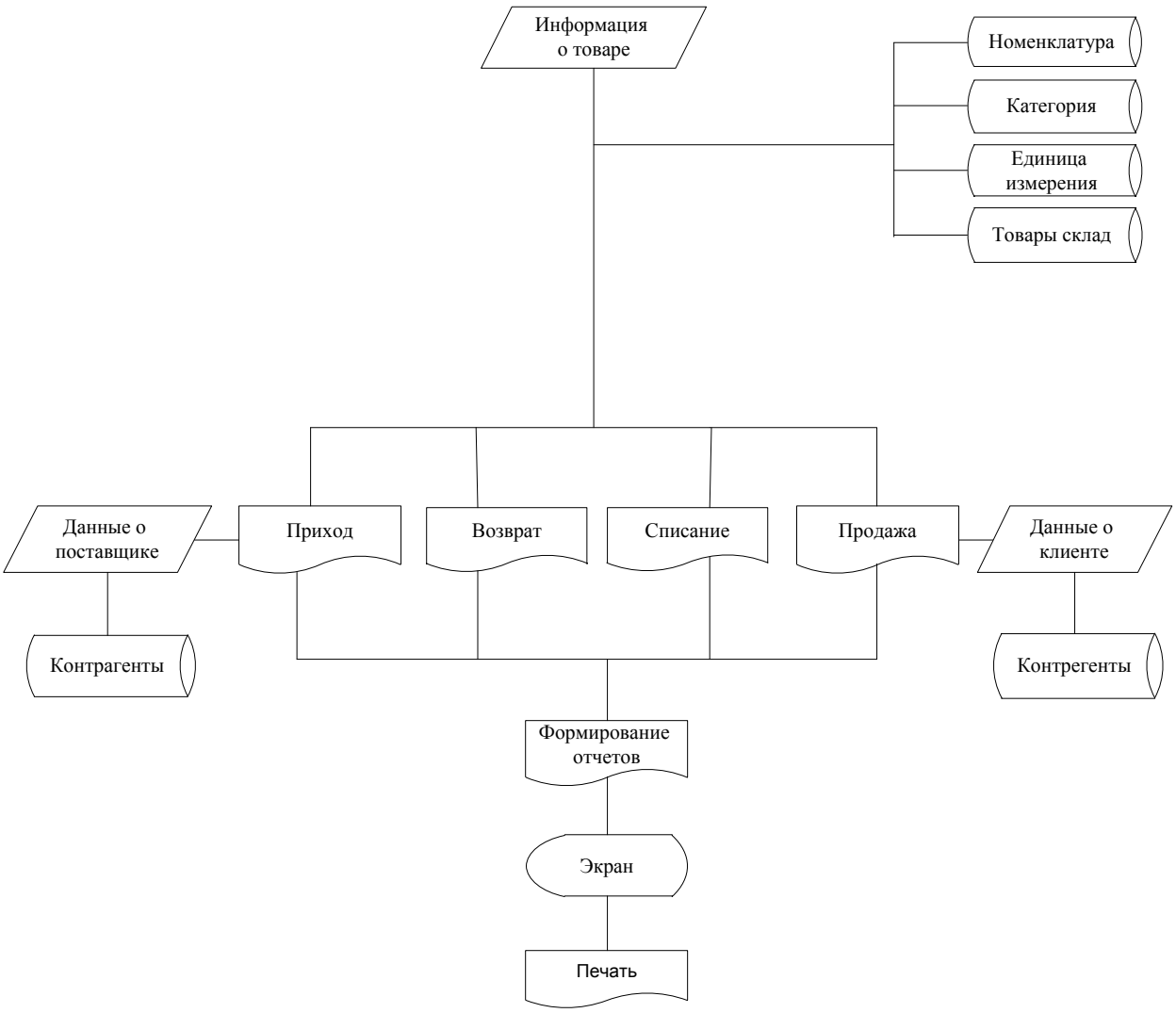


Рисунок А.1 – Схема данных

Приложение Б

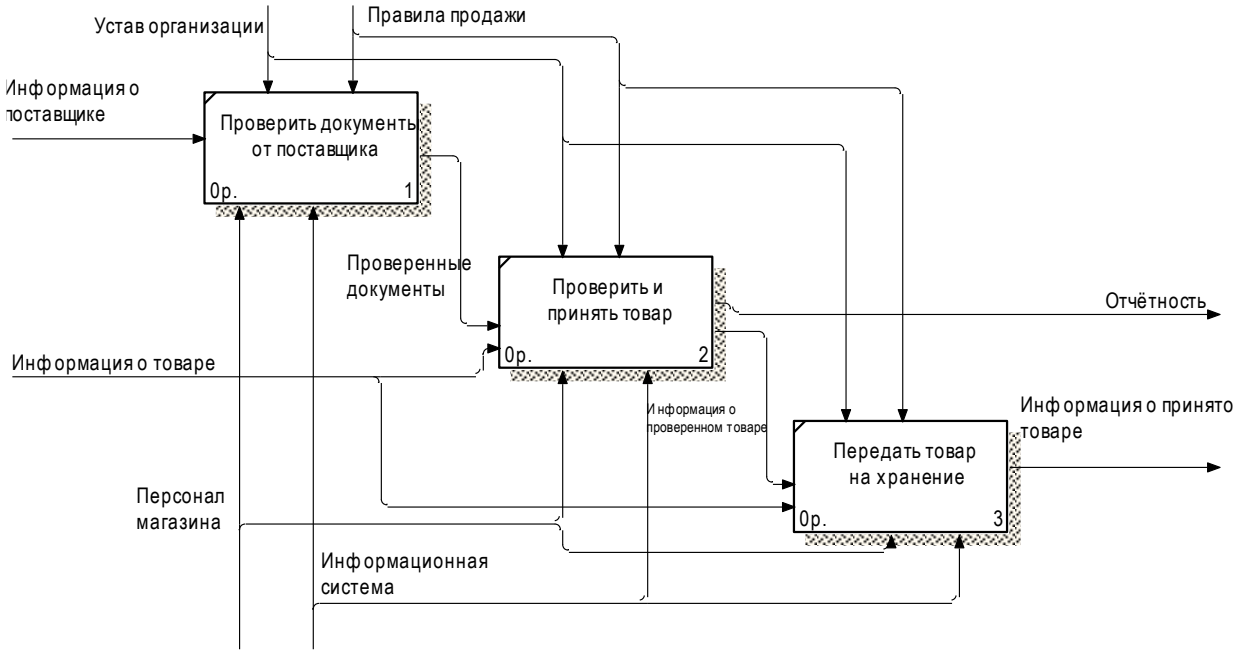


Рисунок Б.1 – Диаграмма декомпозиции блока «Приём товара» (IDEF0)

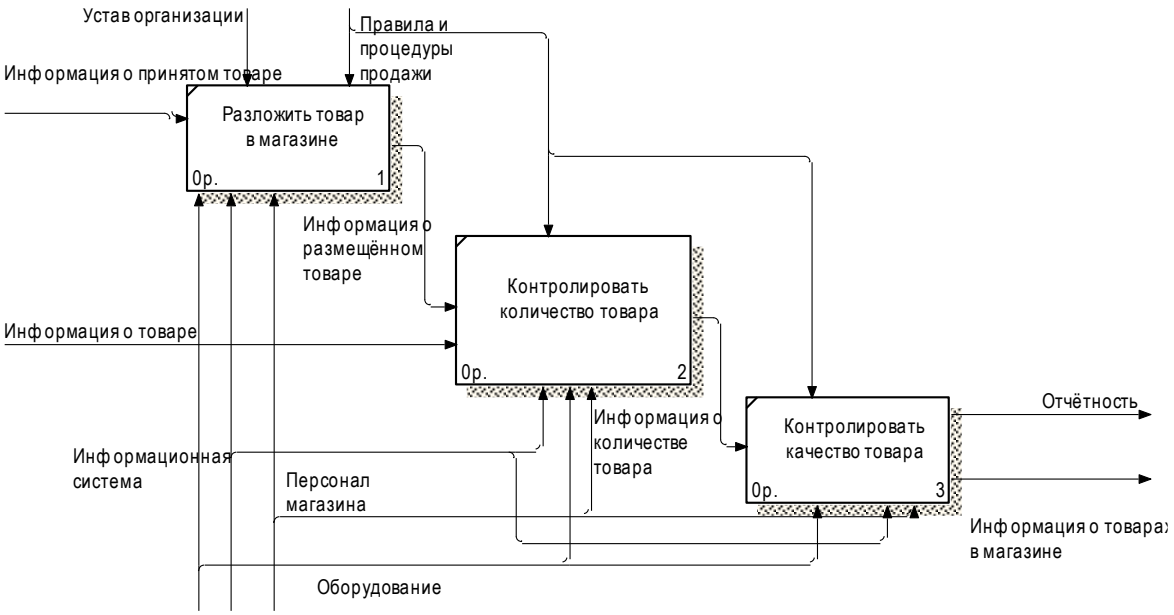


Рисунок Б.2 – Диаграмма декомпозиции блока «Учёт товара» (IDEF0)

Приложение В

Программный код базы данных

Создание таблиц

```
CREATE TABLE TDOCUMTITLES (
   FID      BIGINT NOT NULL,
   FTYPE    INTEGER NOT NULL,
   FSTATE   INTEGER DEFAULT 1 NOT NULL,
   FDATE    DATE DEFAULT CURRENT_DATE NOT NULL,
   FTIME    TIME DEFAULT CURRENT_TIME,
   FPERSON  INTEGER,
   FUSER    INTEGER NOT NULL,
   FSUMM    DECIMAL(15,2));
ALTER TABLE TDOCUMTITLES ADD CONSTRAINT PK_TDOCUMTITLES PRIMARY
KEY (FID);
ALTER TABLE TDOCUMTITLES ADD CONSTRAINT FK_TDOCUMTITLES_1 FOREIGN
KEY (FTYPE) REFERENCES TDOCUMTYPE (FID);
ALTER TABLE TDOCUMTITLES ADD CONSTRAINT FK_TDOCUMTITLES_2 FOREIGN
KEY (FPERSON) REFERENCES TPERSON (FID);
ALTER TABLE TDOCUMTITLES ADD CONSTRAINT FK_TDOCUMTITLES_3 FOREIGN
KEY (FUSER) REFERENCES TUSERS (FID);
ALTER TABLE TDOCUMTITLES ADD CONSTRAINT FK_TDOCUMTITLES_4 FOREIGN
KEY (FSTATE) REFERENCES TDOCUMSTATE (FID);

CREATE TABLE TDOCUMCONTENT (
   FID       BIGINT NOT NULL,
   FDOCID    BIGINT NOT NULL,
   FDOCTYPE  SMALLINT NOT NULL,
   FGOODSID  INTEGER NOT NULL,
   FAMOUNT   NUMERIC(8,3) NOT NULL,
   FPRICE    DECIMAL(15,2),
   FSTOREID  BIGINT);
ALTER TABLE TDOCUMCONTENT ADD CONSTRAINT PK_TDOCUMCONTENT
PRIMARY KEY (FID);
ALTER TABLE TDOCUMCONTENT ADD CONSTRAINT FK_TDOCUMCONTENT_1
FOREIGN KEY (FDOCID) REFERENCES TDOCUMTITLES (FID);
ALTER TABLE TDOCUMCONTENT ADD CONSTRAINT FK_TDOCUMCONTENT_2
FOREIGN KEY (FGOODSID) REFERENCES TGOODS (FID);
ALTER TABLE TDOCUMCONTENT ADD CONSTRAINT FK_TDOCUMCONTENT_3
FOREIGN KEY (FSTOREID) REFERENCES TSTORE (FID);

CREATE TABLE TGOODS (
   FID        INTEGER NOT NULL,
   FGROOP     INTEGER DEFAULT          1,
   FGOODSNAME  VARCHAR(150) DEFAULT 'Новый_товар' NOT NULL,
   FUNIT      INTEGER,
   FARTICLE   VARCHAR(30),
   FCOMMENT   VARCHAR(30),
```

```
   FSTATE      SMALLINT DEFAULT 1,
   FAMOUNT     NUMERIC(8,3),
   FPRICE      DECIMAL(15,2));
ALTER TABLE TGOODS ADD CONSTRAINT PK_TGOODS PRIMARY KEY (FID);
ALTER TABLE TGOODS ADD CONSTRAINT FK_TGOODS_1 FOREIGN KEY (FGROOP)
REFERENCES TGROOP (FID);
ALTER TABLE TGOODS ADD CONSTRAINT FK_TGOODS_2 FOREIGN KEY (FUNIT)
REFERENCES TUNITS (FID);

CREATE TABLE TPERSON (
   FID         INTEGER NOT NULL,
   FTYPE       SMALLINT DEFAULT 1,
   FGROUP      SMALLINT DEFAULT 0 NOT NULL,
   FSHORTNAME  VARCHAR(30) DEFAULT 'NewPerson' NOT NULL,
   FFULLNAME   VARCHAR(250),
   FADDR       VARCHAR(250),
   FADDR2      VARCHAR(250),
   FPHONE      VARCHAR(30),
   FEMAIL      VARCHAR(50),
   FSCHET      VARCHAR(50),
   FBIK        VARCHAR(15),
   FKPP        VARCHAR(15),
   FINN        VARCHAR(12),
   FOKPO       VARCHAR(250),
   FOGRN       VARCHAR(20));
ALTER TABLE TPERSON ADD CONSTRAINT PK_TPERSON PRIMARY KEY (FID);
ALTER TABLE TPERSON ADD CONSTRAINT FK_TPERSON_1 FOREIGN KEY
(FGROUP) REFERENCES TPERSONGROOP (FID);

CREATE TABLE TSTORE (
   FID       BIGINT NOT NULL,
   FGOODSID  INTEGER NOT NULL,
   FAMOUNT   NUMERIC(8,3) NOT NULL,
   FPRICE    DECIMAL(15,2) NOT NULL,
   FPERSON   INTEGER NOT NULL,
   FUSER     INTEGER DEFAULT 1 NOT NULL);
ALTER TABLE TSTORE ADD CONSTRAINT PK_TSTORE PRIMARY KEY (FID);
ALTER TABLE TSTORE ADD CONSTRAINT FK_TSTORE_1 FOREIGN KEY
(FGOODSID) REFERENCES TGOODS (FID);
ALTER TABLE TSTORE ADD CONSTRAINT FK_TSTORE_2 FOREIGN KEY (FPERSON)
REFERENCES TPERSON (FID);
ALTER TABLE TSTORE ADD CONSTRAINT FK_TSTORE_3 FOREIGN KEY (FUSER)
REFERENCES TUSERS (FID);
```

Создание хранимых процедур

```
create or alter procedure DOC_EX (
   VIN_DOCID bigint not null = 0,
   VIN_PERSON integer not null = 0,
   VIN_USER integer not null = 0)
as
declare variable VMY_DOCTYPE smallint; /* тип документа */
declare variable VMY_DOCSTATE smallint; /* состояние документа */
```

```
declare variable VMY_GOODSID integer; /* номер товара */
declare variable VMY_AMOUNT double precision; /* количество товара */
declare variable VMY_PRICE decimal(15,2); /* цена товара */
declare variable VMY_PRICE_DISPLAY decimal(15,2); /* отображаемый цена */
declare variable VMY_AMOUNT_DISPLAY double precision; /* отображаемый остаток */
declare variable VMY_LAPCOUNT integer;
declare variable VMY_STOREID integer; /* код товара на складе */
declare variable VMY_SUMM decimal(15,2); /* сумма */
BEGIN
  VMY_SUMM = 0;
  VMY_LAPCOUNT = 0;
  /* ПОЛУЧИТЬ ТИП ДОКУМЕНТА */
  SELECT FTYPE, FSTATE
  FROM TDOCUMTITLES
  WHERE FID = :VIN_DOCID
  INTO :VMY_DOCTYPE, :VMY_DOCSTATE;
  /* ПРОВЕРИТЬ СОСТОЯНИЕ ДОКУМЕНТА */
  IF (:VMY_DOCSTATE <> 1 ) THEN
     EXCEPTION EDOCUMTITLES_ERRSTATEDOC 'Документ уже Проведён или
Удалён!';
  /*---------- П О С Т У П Л Е Н И Е   Т М Ц ----------*/
  IF (:VMY_DOCTYPE = 1) THEN
  BEGIN
    FOR SELECT FGOODSID, FAMOUNT, FPRICE
       FROM TDOCUMCONTENT
       WHERE (FDOCID = :VIN_DOCID)
       INTO :VMY_GOODSID, :VMY_AMOUNT, :VMY_PRICE
    DO
    BEGIN
      VMY_LAPCOUNT = (:VMY_LAPCOUNT + 1);
      /* РАСЧЕТ СУММЫ */
      VMY_SUMM = (:VMY_SUMM + (:VMY_AMOUNT * :VMY_PRICE));
      /*----- ПОЛУЧИТЬ ТЕКУЩИЕ ЗНАЧЕНИЯ ----- */
      SELECT FAMOUNT, FPRICE
      FROM TGOODS
      WHERE (FID = :VMY_GOODSID)
      INTO :VMY_AMOUNT_DISPLAY, :VMY_PRICE_DISPLAY;
      /*----- ВСТАВКА НА СКЛАД----- */
      INSERT INTO TSTORE (FGOODSID, FAMOUNT, FPRICE, FPERSON, FUSER)
      VALUES (:VMY_GOODSID, :VMY_AMOUNT, :VMY_PRICE, :VIN_PERSON,
:VIN_USER);
      /*----- ВИЗУАЛЬНЫЕ ИЗМЕНЕНИЯ ----- */
      /* РАСЧИТАТЬ НОВОЕ КОЛИЧЕСТВО */
      IF (:VMY_AMOUNT_DISPLAY IS NULL) THEN
      VMY_AMOUNT_DISPLAY = 0;
      VMY_AMOUNT = (:VMY_AMOUNT_DISPLAY + :VMY_AMOUNT);
      /* РАСЧИТАТЬ НОВУЮ ЦЕНУ */
      IF (:VMY_PRICE_DISPLAY IS NOT NULL) THEN
        IF ((:VMY_PRICE_DISPLAY = 0) OR ((:VMY_PRICE_DISPLAY <> :VMY_PRICE)))
THEN
          VMY_PRICE = 0;
      /* ОБНОВИТЬ ТАБЛИЦУ */
```

```
    UPDATE TGOODS
    SET FAMOUNT = :VMY_AMOUNT,
       FPRICE = :VMY_PRICE
    WHERE FID = :VMY_GOODSID;
   END
  END
 ELSE
  /*---------- Р Е А Л И З А Ц И Я   Т О В А Р А ----------*/
 IF (:VMY_DOCTYPE = 2) THEN
 BEGIN
   FOR SELECT C.FGOODSID, C.FAMOUNT, C.fprice, C.FSTOREID
      FROM (TDOCUMCONTENT C
      LEFT JOIN TGOODS G ON C.FGOODSID = G.FID)
      WHERE (C.FDOCID = :VIN_DOCID)
      INTO :VMY_GOODSID, :VMY_AMOUNT, :VMY_PRiCe, :VMY_STOREID
   DO
   BEGIN
    VMY_LAPCOUNT = (:VMY_LAPCOUNT + 1);
    /* РАСЧЕТ СУММЫ */
    VMY_SUMM = (:VMY_SUMM + (:VMY_AMOUNT * :VMY_price));
    BEGIN
       /*----- СПИСАНИЕ СО СКЛАДА ----- */
       SELECT FAMOUNT
       FROM TSTORE
       WHERE (FID = :VMY_STOREID)
       INTO VMY_AMOUNT_DISPLAY;/* ПОЛУЧИТЬ ТЕК. ЗНАЧЕНИЕ НА СКЛАДЕ */
       IF ((:VMY_AMOUNT_DISPLAY - :VMY_AMOUNT) < 0 ) THEN
       EXCEPTION ESTORE_LACKAMOUNT 'Не хватает количества для позиции №
'''||:VMY_LAPCOUNT||''';
       UPDATE TSTORE
       SET FAMOUNT = (:VMY_AMOUNT_DISPLAY - :VMY_AMOUNT)
       WHERE (FID = :VMY_STOREID);/* ОБНОВИТЬ ЗНАЧЕНИЕ НА СКЛАДЕ */
       /*----- ВИЗУАЛЬНЫЕ  ИЗМЕНЕНИЯ----- */
       SELECT FAMOUNT, FPRICE
       FROM TGOODS
       WHERE (FID = :VMY_GOODSID)
       INTO VMY_AMOUNT_DISPLAY, :VMY_PRICE_DISPLAY;
       /* РАСЧИТАТЬ НОВОЕ КОЛИЧЕСТВО */
       VMY_AMOUNT = (:VMY_AMOUNT_DISPLAY - :VMY_AMOUNT);
       IF (:VMY_AMOUNT = 0) THEN
       BEGIN
        VMY_AMOUNT = NULL;
        VMY_PRICE = NULL;
       END
       ELSE
       BEGIN
       /* ПРОВЕРИТЬ НА НАЛИЧИЕ НЕСКОЛЬКИХ ЦЕН */
        IF (:VMY_PRICE_DISPLAY = 0) THEN
          IF ((SELECT FIRST 1 COUNT(DISTINCT FPRICE)
             FROM TSTORE
             WHERE ((FGOODSID = :VMY_GOODSID) AND (FAMOUNT > 0))) > 1) THEN
          VMY_PRICE = 0;
```

```
          ELSE
           VMY_PRICE = (SELECT FIRST 1 FPRICE
             FROM TSTORE
             WHERE ((FGOODSID = :VMY_GOODSID) AND (FAMOUNT > 0)));
        END
        /* ОБНОВИТЬ ТАБЛИЦУ */
        UPDATE TGOODS
        SET FAMOUNT = :VMY_AMOUNT,
          FPRICE = :VMY_PRICE
        WHERE FID = :VMY_GOODSID;
        end end
    END
    ELSE
     /*---------- С П И С А Н И Е   Т О В А Р А ------------*/
   IF (:VMY_DOCTYPE = 3) THEN
   BEGIN
    VMY_LAPCOUNT = (:VMY_LAPCOUNT + 1);
    FOR SELECT FGOODSID, FAMOUNT, FPRICE, FSTOREID
       FROM TDOCUMCONTENT
       WHERE (FDOCID = :VIN_DOCID)
       INTO :VMY_GOODSID, :VMY_AMOUNT, :VMY_PRICE, :VMY_STOREID
    DO
    BEGIN
      /*----- СПИСАНИЕ СО СКЛАДА ----- */
      SELECT FAMOUNT
      FROM TSTORE
      WHERE (FID = :VMY_STOREID)
      INTO VMY_AMOUNT_DISPLAY;/* ПОЛУЧИТЬ ТЕК. ЗНАЧЕНИЕ НА СКЛАДЕ */
      IF ((:VMY_AMOUNT_DISPLAY - :VMY_AMOUNT) < 0 ) THEN
      EXCEPTION ESTORE_LACKAMOUNT 'Не хватает количества для позиции №
'''||:VMY_LAPCOUNT||''';
      UPDATE TSTORE
      SET FAMOUNT = (:VMY_AMOUNT_DISPLAY - :VMY_AMOUNT)
      WHERE (FID = :VMY_STOREID);/* ОБНОВИТЬ ЗНАЧЕНИЕ НА СКЛАДЕ */
      /* РАСЧЕТ СУММЫ */
      VMY_SUMM = (:VMY_SUMM + (:VMY_AMOUNT * :VMY_PRICE));
      /*----- ВИЗУАЛЬНЫЕ  ИЗМЕНЕНИЯ----- */
      SELECT FAMOUNT, FPRICE
      FROM TGOODS
      WHERE (FID = :VMY_GOODSID)
      INTO VMY_AMOUNT_DISPLAY, :VMY_PRICE_DISPLAY;
      /* РАСЧИТАТЬ НОВОЕ КОЛИЧЕСТВО */
      VMY_AMOUNT = (:VMY_AMOUNT_DISPLAY - :VMY_AMOUNT);
      /* РАСЧИТАТЬ НОВУЮ ЦЕНУ */
      IF (:VMY_AMOUNT = 0) THEN
      BEGIN
        VMY_AMOUNT = NULL;
        VMY_PRICE = NULL;
      END
      ELSE
      BEGIN
        /* ПРОВЕРИТЬ НА НАЛИЧИЕ НЕСКОЛЬКИХ ЦЕН */
```

```
          IF (:VMY_PRICE_DISPLAY = 0) THEN
           IF ((SELECT FIRST 1 COUNT(DISTINCT FPRICE)
              FROM TSTORE
              WHERE ((FGOODSID = :VMY_GOODSID) AND (FAMOUNT > 0))) > 1) THEN
            VMY_PRICE = 0;
            ELSE
             VMY_PRICE = (SELECT FIRST 1 FPRICE
              FROM TSTORE
              WHERE ((FGOODSID = :VMY_GOODSID) AND (FAMOUNT > 0)));
     END
     /* ОБНОВИТЬ ТАБЛИЦУ */
     UPDATE TGOODS
     SET FAMOUNT = :VMY_AMOUNT,
        FPRICE = :VMY_PRICE
     WHERE FID = :VMY_GOODSID;
    END
  END
  ELSE
    /*---------- В О З В Р А Т   Т О В А Р А ----------*/
  IF (:VMY_DOCTYPE = 4) THEN
  BEGIN
    FOR SELECT C.FGOODSID,  C.FAMOUNT, C.FPRICE
       FROM (TDOCUMCONTENT C
       LEFT JOIN TGOODS G ON C.FGOODSID = G.FID)
       WHERE (C.FDOCID = :VIN_DOCID)
       INTO :VMY_GOODSID, :VMY_AMOUNT, :VMY_PRICE
    DO
    BEGIN
     VMY_LAPCOUNT = (:VMY_LAPCOUNT + 1);
     /* РАСЧЕТ СУММЫ */
     VMY_SUMM = (:VMY_SUMM + (:VMY_AMOUNT * :VMY_PRICE));
        /*----- ВСТАВКА НА СКЛАД----- */
        INSERT INTO TSTORE (FGOODSID, FAMOUNT, FPRICE, FPERSON, FUSER)
        VALUES (:VMY_GOODSID, :VMY_AMOUNT, :VMY_PRICE, :VIN_PERSON,
:VIN_USER);
        /*----- ВИЗУАЛЬНЫЕ  ИЗМЕНЕНИЯ----- */
        SELECT FAMOUNT, FPRICE
        FROM TGOODS
        WHERE (FID = :VMY_GOODSID)
        INTO VMY_AMOUNT_DISPLAY, :VMY_PRICE_DISPLAY;
        /* РАСЧИТАТЬ НОВОЕ КОЛИЧЕСТВО */
        IF (:VMY_AMOUNT_DISPLAY IS NULL) THEN
        VMY_AMOUNT_DISPLAY = 0;
        VMY_AMOUNT = (:VMY_AMOUNT_DISPLAY + :VMY_AMOUNT);
        /* РАСЧИТАТЬ НОВУЮ ЦЕНУ */
        IF (:VMY_PRICE_DISPLAY IS NOT NULL) THEN
         IF ((:VMY_PRICE_DISPLAY = 0) OR ((:VMY_PRICE_DISPLAY <>
:VMY_PRICE))) THEN
           VMY_PRICE = 0;
        /* ОБНОВИТЬ ТАБЛИЦУ */
        UPDATE TGOODS
        SET FAMOUNT = :VMY_AMOUNT,
```

```
            FPRICE = :VMY_PRICE
          WHERE FID = :VMY_GOODSID;
    END /* FOR */
    End

create or alter procedure PDOCUMTITLES_GETTITLE (
    VIN_SHOWDEL smallint,
    VIN_STARTDATE date,
    VIN_ENDDATE date,
    VIN_METHOD smallint,
    VIN_VALUE integer,
    VIN_USERID integer)
returns (
    VID bigint,
    VTYPENAME varchar(20),
    VTYPEID smallint,
    VSTATEID smallint,
    VSTATENAME varchar(15),
    VDATE date,
    VTIME time,
    VPERSONID bigint,
    VPERSON varchar(30),
    VUSER varchar(20),
    VSUMM decimal(15,2))
as
BEGIN
  FOR SELECT D.FID, D.FTYPE,b.ftype, D.FSTATE,a.FSTATE, D.FDATE, D.FTIME,
D.FPERSON, P.FSHORTNAME, U.FLOGIN, D.FSUMM
      FROM ((((TDOCUMTITLES D
      LEFT JOIN TPERSON P ON D.FPERSON = P.FID)
      LEFT JOIN TUSERS U ON D.FUSER = U.FID)
      LEFT JOIN tdocumstate a ON D.fstate = a.fid)
      LEFT JOIN tdocumtype b ON D.ftype = b.fid)
      WHERE ((:VIN_SHOWDEL = 0 AND
          D.FSTATE IN (1, 2) /*ОТОБРАЖАТЬ ТОЛЬКО НЕ УДАЛЕННЫЕ*/
          OR (:VIN_SHOWDEL = 1)) /*ОТОБРАЖАТЬ ВСЕ*/
          AND
          ((:VIN_METHOD = 0 AND D.FUSER = :VIN_USERID)
          OR (:VIN_METHOD IN (1, 2) AND D.FPERSON = :VIN_VALUE AND D.FUSER =
:VIN_USERID)
          OR (:VIN_METHOD = 3 AND D.FTYPE = :VIN_VALUE AND D.FUSER =
:VIN_USERID)
          OR (:VIN_METHOD = 5 AND (:VIN_VALUE = 0 OR (:VIN_VALUE > 0 AND
D.FUSER = :VIN_VALUE))))
          AND (D.FDATE BETWEEN :VIN_STARTDATE AND :VIN_ENDDATE)
          OR (:VIN_METHOD = 0 AND D.FSTATE = 0 AND D.FUSER = :VIN_USERID))
      INTO :VID, :VTYPEid, :VTYPEname, :VSTATEID, :VSTATENAME, :VDATE, :VTIME,
:VPERSONID, :VPERSON, :VUSER, :VSUMM
  DO
  BEGIN
    SUSPEND;
  END
```

```
create or alter procedure PGOODS_GETGOODS (
    VIN_CURRID integer not null,
    VIN_AVAILABLE smallint,
    VIN_FINDTEXT varchar(40),
    VIN_SHOWDEL integer)
returns (
    VID integer,
    VGOODSNAME varchar(150),
    VARTICLE varchar(30),
    VUNIT varchar(10),
    VCOMMENT varchar(30),
    VPRICE varchar(15),
    VAMOUNT varchar(10),
    VSTATE smallint)
as
BEGIN
   /*VIN_CURRID =
   >=0 Return category content
   -1  Find AUTO
   -2  Find by BARCODE
   -3  Find by ID
   -4  Find by Name:
   -5  Find by Article
    */
  FOR SELECT G.FID, G.FSTATE, G.FGOODSNAME,  G.FARTICLE, G.FCOMMENT,
U.FUNITNAME, G.FAMOUNT, G.FPRICE
      FROM (TGOODS G
      LEFT JOIN TUNITS U ON G.FUNIT = U.FID)
      WHERE ((:VIN_CURRID > 0 AND
          G.fgroop = :VIN_CURRID)
          OR (:VIN_CURRID = -1 AND (
             G.FGOODSNAME CONTAINING (:VIN_FINDTEXT) OR
             G.FARTICLE LIKE :VIN_FINDTEXT || '%'
              ))
          OR (:VIN_CURRID = -3 AND G.FID = :VIN_FINDTEXT)
          OR (:VIN_CURRID = -4 AND G.FGOODSNAME CONTAINING
(:VIN_FINDTEXT))
          OR (:VIN_CURRID = -5 AND G.FARTICLE LIKE :VIN_FINDTEXT || '%'))
          AND
          ((:VIN_SHOWDEL = 0 AND
          (G.FSTATE = 1)) /*ОТОБРАЖАТЬ ТОЛЬКО НЕ УДАЛЕННЫЕ*/
          OR (:VIN_SHOWDEL = 1)) /*ОТОБРАЖАТЬ ВСЕ*/
          AND
          ((:VIN_AVAILABLE = 0) /*ОТОБРАЖАТЬ БЕЗ УЧЕТА НАЛИЧИЯ*/
          OR (:VIN_AVAILABLE = 1 AND
          G.FAMOUNT IS NOT NULL)) /*ОТОБРАЖАТЬ ТОЛЬКО ТОВАРЫ В
НАЛИЧИИ*/
      INTO :VID, :VSTATE,  :VGOODSNAME,  :VARTICLE, :VCOMMENT, :VUNIT,
:VAMOUNT, :VPRICE
  DO
  BEGIN
```

```
  IF (:VPRICE = 0) THEN VPRICE = '<несколько>';
    SUSPEND;
  END
```

Создание триггеров

```
CREATE OR ALTER TRIGGER TSTORE_BI FOR TSTORE
ACTIVE BEFORE INSERT POSITION 0
AS
BEGIN
  IF (NEW.FID IS NULL) THEN
    NEW.FID = GEN_ID(GEN_TSTORE_ID,1);
END
```

Создание генераторов

```
CREATE SEQUENCE GEN_TDOCUMCONTENT_ID;
CREATE SEQUENCE GEN_TDOCUMTITLES_ID;
CREATE SEQUENCE GEN_TGOODS_ID;
CREATE SEQUENCE GEN_TGROOP_ID;
CREATE SEQUENCE GEN_TPERSON_ID;
CREATE SEQUENCE GEN_TSTORE_ID;
CREATE SEQUENCE GEN_TUNITS_ID;
CREATE SEQUENCE GEN_TUSERS_ID;
```

Приложение Г

Программный код win-приложения

Unit1.cpp

```
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#include "Unit5.h"
#include "Unit4.h"
#include "Unit6.h"
#include "Unit7.h"
#include "Unit8.h"
#include "Unit10.h"
#include "Unit11.h"
#include "Unit12.h"
#include "Unit9.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
__fastcall TForm1::TForm1(TComponent* Owner)
      : TForm(Owner)
{
}
//ПРИХОД ТОВАРА
void __fastcall TForm1::Button1Click(TObject *Sender)
{Application->CreateForm(__classid(TForm3), &Form3);
Form3->type_doc->Caption="Приход товара (Приходная накладная)";
Form3->type->Caption="1";
Form3->state_doc->Caption="Сохранён, не проведён";
DataModule2->new_doc->ParamByName("VIN_TYPE")->AsInteger=1;
DataModule2->new_doc->ParamByName("VIN_USER")->AsInteger=1;
DataModule2->new_doc->Prepare();
DataModule2->new_doc->ExecProc();
Form3->number_doc->Caption=DataModule2->new_doc-
>ParamByName("VNEWID")->AsString;
DataModule2->IBTransaction1->Commit();
DataModule2->person_select->Close();
DataModule2->person_select->ParamByName("VIN_GROUP")->AsInteger=1;
DataModule2->person_select->Open();
DataModule2->groop_select->Open();
DataModule2->goods_select->Close();
```

```
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->ParamByName("VIN_AVAILABLE")-
>AsInteger=0;
DataModule2->goods_select->Open();
Form3->ShowModal();
}//---------------------------------------------------------------------------

//ПРОДАЖА ТОВАРА
void __fastcall TForm1::Button2Click(TObject *Sender)
{Application->CreateForm(__classid(TForm3), &Form3);
Form3->type_doc->Caption="Продажа товара (Расходная накладная)";
Form3->type->Caption="2";
Form3->state_doc->Caption="Сохранён, не проведён";
DataModule2->new_doc->ParamByName("VIN_TYPE")->AsInteger=2;
DataModule2->new_doc->ParamByName("VIN_USER")->AsInteger=1;
DataModule2->new_doc->Prepare();
DataModule2->new_doc->ExecProc();
Form3->number_doc->Caption=DataModule2->new_doc-
>ParamByName("VNEWID")->AsString;
DataModule2->IBTransaction1->Commit();
DataModule2->person_select->Close();
DataModule2->person_select->ParamByName("VIN_GROUP")->AsInteger=2;
DataModule2->person_select->Open();
DataModule2->groop_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->ParamByName("VIN_AVAILABLE")-
>AsInteger=1;
DataModule2->goods_select->Open();
Form3->ShowModal();
}//---------------------------------------------------------------------------

//СПИСАНИЕ ТОВАРА
void __fastcall TForm1::Button3Click(TObject *Sender)
{Application->CreateForm(__classid(TForm3), &Form3);
Form3->type_doc->Caption="Списание товара (Расходная накладная)";
Form3->type->Caption="3";
Form3->state_doc->Caption="Сохранён, не проведён";DataModule2->new_doc-
>ParamByName("VIN_TYPE")->AsInteger=3;
DataModule2->new_doc->ParamByName("VIN_USER")->AsInteger=1;
DataModule2->new_doc->Prepare();
DataModule2->new_doc->ExecProc();
```

```
Form3->number_doc->Caption=DataModule2->new_doc-
>ParamByName("VNEWID")->AsString;
DataModule2->IBTransaction1->Commit();
DataModule2->person_select->Close();
DataModule2->person_select->ParamByName("VIN_GROUP")->AsInteger=3;
DataModule2->person_select->Open();
DataModule2->groop_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->ParamByName("VIN_AVAILABLE")-
>AsInteger=1;
DataModule2->goods_select->Open();
Form3->ShowModal();
}//---------------------------------------------------------------------------

//ВОЗВРАТ ТОВАРА
void __fastcall TForm1::Button4Click(TObject *Sender)
{Application->CreateForm(__classid(TForm3), &Form3);
Form3->type_doc->Caption="Возврат товара (Приходная накладная)";
Form3->type->Caption="4";
Form3->state_doc->Caption="Сохранён, не проведён";
DataModule2->new_doc->ParamByName("VIN_TYPE")->AsInteger=4;
DataModule2->new_doc->ParamByName("VIN_USER")->AsInteger=1;
DataModule2->new_doc->Prepare();
DataModule2->new_doc->ExecProc();
Form3->number_doc->Caption=DataModule2->new_doc-
>ParamByName("VNEWID")->AsString;
DataModule2->IBTransaction1->Commit();
DataModule2->person_select->Close();
DataModule2->person_select->ParamByName("VIN_GROUP")->AsInteger=2;
DataModule2->person_select->Open();
DataModule2->groop_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->ParamByName("VIN_AVAILABLE")-
>AsInteger=0;
DataModule2->goods_select->Open();
Form3->ShowModal();
}
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{DataModule2->IBTransaction1->Commit();}
```

```
void __fastcall TForm1::Button5Click(TObject *Sender)
{Application->CreateForm(__classid(TForm4), &Form4);
Form4->PageControl1->ActivePage=0;
Form4->ShowModal();
}

void __fastcall TForm1::Button9Click(TObject *Sender)
{Application->CreateForm(__classid(TForm8), &Form8);
DataModule2->title_select->Close();
DataModule2->title_select->ParamByName("VIN_SHOWDEL")->AsInteger=0;
DataModule2->title_select->ParamByName("VIN_STARTDATE")-
>AsDate=Form8->DateTimePicker1->Date;
DataModule2->title_select->ParamByName("VIN_ENDDATE")-
>AsDate=Form8->DateTimePicker2->Date;
DataModule2->title_select->ParamByName("VIN_METHOD")->AsInteger=0;
DataModule2->title_select->ParamByName("VIN_VALUE")->AsInteger=1;
DataModule2->title_select->ParamByName("VIN_USERID")->AsInteger=1;
DataModule2->title_select->Open();
Form8->ShowModal();}

void __fastcall TForm1::Button6Click(TObject *Sender)
{Application->CreateForm(__classid(TForm10), &Form10);
Form10->ShowModal();}

void __fastcall TForm1::Button7Click(TObject *Sender)
{ Application->CreateForm(__classid(TForm11), &Form11);
Form11->ShowModal();}

void __fastcall TForm1::Button8Click(TObject *Sender)
{Application->CreateForm(__classid(TForm12), &Form12);
Form12->ShowModal();}
```

Unit3.cpp

```
#include <vcl.h>
#pragma hdrstop
#include "Unit3.h"
#include "Unit1.h"
#include "Unit2.h"
#include "Unit5.h"
#include "Unit4.h"
#include "Unit6.h"
#include "Unit7.h"
#include "Unit8.h"
#include "Unit9.h"
```

```
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm3 *Form3;
__fastcall TForm3::TForm3(TComponent* Owner)
	: TForm(Owner)
{
}
void __fastcall TForm3::DBGrid1DblClick(TObject *Sender)
{
DataModule2->edit_title->ParamByName("VIN_ACTION")->AsInteger=0;
DataModule2->edit_title->ParamByName("VIN_ID")-
>AsInteger=StrToInt(number_doc->Caption);
DataModule2->edit_title->ParamByName("VIN_PERSON")-
>AsInteger=DataModule2->person_select->FieldByName("VID")->AsInteger;
DataModule2->edit_title->ParamByName("VIN_STATE")->AsInteger=1;
DataModule2->edit_title->ParamByName("VIN_SUMM")-
>AsFloat=DataModule2->get_content->FieldByName("summ")->AsFloat;
DataModule2->edit_title->Prepare();
DataModule2->edit_title->ExecProc();
ShowMessage("Контрагент успешно изменён!");
Button1->Enabled=True;
}
void __fastcall TForm3::Button1Click(TObject *Sender)
{
DataModule2->doc_ex->ParamByName("VIN_DOCID")-
>AsInteger=StrToInt(Form3->number_doc->Caption);
DataModule2->doc_ex->ParamByName("VIN_PERSON")-
>AsInteger=DataModule2->person_select->FieldByName("VID")->AsInteger;
DataModule2->doc_ex->ParamByName("VIN_USER")->AsInteger=1;
DataModule2->doc_ex->Prepare();
DataModule2->doc_ex->ExecProc();
DataModule2->edit_title->ParamByName("VIN_ACTION")->AsInteger=1;
DataModule2->edit_title->ParamByName("VIN_ID")-
>AsInteger=StrToInt(number_doc->Caption);
DataModule2->edit_title->ParamByName("VIN_PERSON")-
>AsInteger=DataModule2->person_select->FieldByName("VID")->AsInteger;
DataModule2->edit_title->ParamByName("VIN_STATE")->AsInteger=2;
DataModule2->edit_title->ParamByName("VIN_SUMM")-
>AsFloat=StrToFloat(Form3->itog->Caption);
DataModule2->edit_title->Prepare();
DataModule2->edit_title->ExecProc();
DataModule2->IBTransaction1->Commit();
Form3->Close();
}
```

```
void __fastcall TForm3::DBGrid2CellClick(TColumn *Column)
{
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->Open();
}
//---------------------------------------------------------------------------
void __fastcall TForm3::GRID_GOODSDblClick(TObject *Sender)
{
int doctype;
doctype=StrToInt(Form3->type->Caption);
DataModule2->edit_content->ParamByName("VIN_ID")->AsInteger=-1;
DataModule2->edit_content->ParamByName("VIN_DOCID")-
>AsInteger=StrToInt(Form3->number_doc->Caption);
DataModule2->edit_content->ParamByName("VIN_DOCTYPE")-
>AsInteger=doctype;
DataModule2->edit_content->ParamByName("VIN_GOODSID")-
>AsInteger=DataModule2->goods_select->FieldByName("VID")->AsInteger;
switch(doctype)
{
case 1: Application->CreateForm(__classid(TForm5), &Form5);
Form5->name_goods->Caption=DataModule2->goods_select-
>FieldByName("VGOODSNAME")->AsString;
Form5->ShowModal();break;
case 2: Application->CreateForm(__classid(TForm9), &Form9);
Form9->name_goods->Caption=DataModule2->goods_select-
>FieldByName("VGOODSNAME")->AsString;
DataModule2->price_select->Close();
DataModule2->price_select->ParamByName("VIN_GOODSID")-
>AsInteger=DataModule2->goods_select->FieldByName("VID")->AsInteger;
DataModule2->price_select->ParamByName("VIN_ACT")->AsInteger=1;
DataModule2->price_select->Open();
Form9->ShowModal();break;
case 3: Application->CreateForm(__classid(TForm9), &Form9);
Form9->name_goods->Caption=DataModule2->goods_select-
>FieldByName("VGOODSNAME")->AsString;
DataModule2->price_select->Close();
DataModule2->price_select->ParamByName("VIN_GOODSID")-
>AsInteger=DataModule2->goods_select->FieldByName("VID")->AsInteger;
DataModule2->price_select->ParamByName("VIN_ACT")->AsInteger=1;
DataModule2->price_select->Open();
Form9->ShowModal();break;
case 4: Application->CreateForm(__classid(TForm5), &Form5);
```

```
Form5->name_goods->Caption=DataModule2->goods_select-
>FieldByName("VGOODSNAME")->AsString;
Form5->ShowModal();break;
}}
```
Unit4.cpp
```
#include <vcl.h>
#pragma hdrstop
#include "Unit4.h"
#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#include "Unit5.h"
#include "Unit6.h"
#include "Unit7.h"
#include "Unit8.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm4 *Form4;
__fastcall TForm4::TForm4(TComponent* Owner)
      : TForm(Owner)
{
}
void __fastcall TForm4::DBGrid5CellClick(TColumn *Column)
{DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->Open();}
//---------------------------------------------------------------------------
void __fastcall TForm4::TabSheet1Show(TObject *Sender)
{DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->Open();}
void __fastcall TForm4::TabSheet2Show(TObject *Sender)
{DataModule2->groop_select->Open();}
void __fastcall TForm4::TabSheet3Show(TObject *Sender)
{DataModule2->units_select->Open();}
//---------------------------------------------------------------------------
//КАТЕГОРИИ НОМЕНКЛАТУРЫ
void __fastcall TForm4::Button1Click(TObject *Sender)
{String name;
name=InputBox("Создать новый элемент", "Наименование", "");
DataModule2->edit_groop->ParamByName("VIN_ACTION")->AsInteger=1;
```

```
DataModule2->edit_groop->ParamByName("VIN_ID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->edit_groop->ParamByName("VIN_NAME")->AsString=name;
DataModule2->edit_groop->Prepare();
DataModule2->edit_groop->ExecProc();
DataModule2->IBTransaction1->Commit();
DataModule2->groop_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->Open();
DataModule2->units_select->Open();}
//---------------------------------------------------------------------------
void __fastcall TForm4::Button2Click(TObject *Sender)
{String name;
name=InputBox("Изменить текущий элемент", "Наименование", DataModule2-
>groop_select->FieldByName("VNAME")->AsString);
DataModule2->edit_groop->ParamByName("VIN_ACTION")->AsInteger=2;
DataModule2->edit_groop->ParamByName("VIN_ID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->edit_groop->ParamByName("VIN_NAME")->AsString=name;
DataModule2->edit_groop->Prepare();
DataModule2->edit_groop->ExecProc();
DataModule2->IBTransaction1->Commit();
DataModule2->groop_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->Open();
DataModule2->units_select->Open();}
//---------------------------------------------------------------------------
void __fastcall TForm4::Button3Click(TObject *Sender)
{DataModule2->edit_groop->ParamByName("VIN_ACTION")->AsInteger=3;
DataModule2->edit_groop->ParamByName("VIN_ID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->edit_groop->ParamByName("VIN_NAME")->AsString="";
DataModule2->edit_groop->Prepare();
DataModule2->edit_groop->ExecProc();
DataModule2->IBTransaction1->Commit();
DataModule2->groop_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->Open();
```

```
DataModule2->units_select->Open();}
//---------------------------------------------------------------------------
//ЕДИНИЦЫ ИЗМЕРЕНИЯ
void __fastcall TForm4::Button4Click(TObject *Sender)
{String name;
name=InputBox("Создать новый элемент", "Наименование", "");
DataModule2->edit_units->ParamByName("VIN_ACTION")->AsInteger=1;
DataModule2->edit_units->ParamByName("VIN_ID")->AsInteger=DataModule2-
>units_select->FieldByName("FID")->AsInteger;
DataModule2->edit_units->ParamByName("VIN_NAME")->AsString=name;
DataModule2->edit_units->Prepare();
DataModule2->edit_units->ExecProc();
DataModule2->IBTransaction1->Commit();
DataModule2->groop_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->Open();
DataModule2->units_select->Open();}
//---------------------------------------------------------------------------
void __fastcall TForm4::Button5Click(TObject *Sender)
{String name;
name=InputBox("Изменить текущий элемент", "Наименование", DataModule2-
>units_select->FieldByName("FUNITNAME")->AsString);
DataModule2->edit_units->ParamByName("VIN_ACTION")->AsInteger=2;
DataModule2->edit_units->ParamByName("VIN_ID")->AsInteger=DataModule2-
>units_select->FieldByName("FID")->AsInteger;
DataModule2->edit_units->ParamByName("VIN_NAME")->AsString=name;
DataModule2->edit_units->Prepare();
DataModule2->edit_units->ExecProc();
DataModule2->IBTransaction1->Commit();
DataModule2->groop_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->Open();
DataModule2->units_select->Open();}
//---------------------------------------------------------------------------
void __fastcall TForm4::Button6Click(TObject *Sender)
{DataModule2->edit_units->ParamByName("VIN_ACTION")->AsInteger=3;
DataModule2->edit_units->ParamByName("VIN_ID")->AsInteger=DataModule2-
>units_select->FieldByName("FID")->AsInteger;
DataModule2->edit_units->ParamByName("VIN_NAME")->AsString="";
DataModule2->edit_units->Prepare();
```

```
DataModule2->edit_units->ExecProc();
DataModule2->IBTransaction1->Commit();
DataModule2->groop_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->Open();
DataModule2->units_select->Open();}
//НОМЕНКЛАТУРА
void __fastcall TForm4::Button7Click(TObject *Sender)
{Application->CreateForm(__classid(TForm6), &Form6);
Form6->Caption="Создать новую позицию";
DataModule2->edit_goods->ParamByName("VIN_ACTION")->AsInteger=1;
Form6->ShowModal();}
void __fastcall TForm4::Button8Click(TObject *Sender)
{Application->CreateForm(__classid(TForm6), &Form6);
Form6->Caption="Редактировать текущую позицию";
Form6->Edit1->Text=DataModule2->goods_select-
>FieldByName("VGOODSNAME")->AsString;
Form6->Edit2->Text=DataModule2->goods_select-
>FieldByName("VARTICLE")->AsString;
Form6->Edit3->Text=DataModule2->goods_select-
>FieldByName("VCOMMENT")->AsString;
DataModule2->edit_goods->ParamByName("VIN_ACTION")->AsInteger=2;
Form6->ShowModal();}
//---------------------------------------------------------------------------
void __fastcall TForm4::Button9Click(TObject *Sender)
{DataModule2->edit_goods->ParamByName("VIN_ACTION")->AsInteger=3;
DataModule2->edit_goods->ParamByName("VIN_ID")-
>AsInteger=DataModule2->goods_select->FieldByName("VID")->AsInteger;
DataModule2->edit_goods->ParamByName("VIN_GOODSNAME")-
>AsString="";
DataModule2->edit_goods->ParamByName("VIN_ARTICLE")->AsString="";
DataModule2->edit_goods->ParamByName("VIN_GROOP")->AsString="";
DataModule2->edit_goods->ParamByName("VIN_UNIT")->AsString="";
DataModule2->edit_goods->ParamByName("VIN_COMMENT")->AsString="";
DataModule2->edit_goods->Prepare();
DataModule2->edit_goods->ExecProc();
DataModule2->IBTransaction1->Commit();
DataModule2->groop_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->Open();
```

```
DataModule2->units_select->Open();}
//КОНТРАГЕНТЫ
void __fastcall TForm4::Button10Click(TObject *Sender)
{Application->CreateForm(__classid(TForm7), &Form7);
Form7->Caption="Создать новую позицию";
DataModule2->edit_person->ParamByName("VIN_ID")->AsInteger=-1;
DataModule2->edit_person->ParamByName("VIN_NEWTYPE")->AsInteger=1;
Form7->ShowModal();}
//---------------------------------------------------------------------------
void __fastcall TForm4::Button11Click(TObject *Sender)
{Application->CreateForm(__classid(TForm7), &Form7);
Form7->Caption="Редактировать текущую позицию";
DataModule2->edit_person->ParamByName("VIN_ID")-
>AsInteger=DataModule2->person_select->FieldByName("VID")->AsInteger;
DataModule2->edit_person->ParamByName("VIN_NEWTYPE")->AsInteger=-1;
Form7->Edit1->Text=DataModule2->person_select-
>FieldByName("vshortname")->AsString;
Form7->Edit2->Text=DataModule2->person_select->FieldByName("vfullname")-
>AsString;
Form7->Edit3->Text=DataModule2->person_select->FieldByName("vaddr")-
>AsString;
Form7->Edit4->Text=DataModule2->person_select->FieldByName("vaddr2")-
>AsString;
Form7->Edit5->Text=DataModule2->person_select->FieldByName("vphone")-
>AsString;
Form7->Edit6->Text=DataModule2->person_select->FieldByName("vemail")-
>AsString;
Form7->Edit7->Text=DataModule2->person_select->FieldByName("vschet")-
>AsString;
Form7->Edit8->Text=DataModule2->person_select->FieldByName("vbik")-
>AsString;
Form7->Edit9->Text=DataModule2->person_select->FieldByName("vkpp")-
>AsString;
Form7->Edit10->Text=DataModule2->person_select->FieldByName("vinn")-
>AsString;
Form7->Edit11->Text=DataModule2->person_select->FieldByName("vokpo")-
>AsString;
Form7->Edit12->Text=DataModule2->person_select->FieldByName("vogrn")-
>AsString;
Form7->ShowModal();}
//---------------------------------------------------------------------------
void __fastcall TForm4::Button12Click(TObject *Sender)
{DataModule2->edit_person->ParamByName("VIN_ID")-
>AsInteger=DataModule2->person_select->FieldByName("VID")->AsInteger;
```

```
DataModule2->edit_person->ParamByName("VIN_NEWTYPE")->AsInteger=0;
DataModule2->edit_person->ParamByName("VIN_SHORTNAME")-
>AsString="";
DataModule2->edit_person->ParamByName("VIN_FULLNAME")->AsString="";
DataModule2->edit_person->ParamByName("VIN_ADDR")->AsString="";
DataModule2->edit_person->ParamByName("VIN_ADDR2")->AsString="";
DataModule2->edit_person->ParamByName("VIN_PHONE")->AsString="";
DataModule2->edit_person->ParamByName("VIN_EMAIL")->AsString="";
DataModule2->edit_person->ParamByName("VIN_SCHET")->AsString="";
DataModule2->edit_person->ParamByName("VIN_BIK")->AsString="";
DataModule2->edit_person->ParamByName("VIN_KPP")->AsString="";
DataModule2->edit_person->ParamByName("VIN_INN")->AsString="";
DataModule2->edit_person->ParamByName("VIN_OKPO")->AsString="";
DataModule2->edit_person->ParamByName("VIN_OGRN")->AsString="";
DataModule2->edit_person->Prepare();
DataModule2->edit_person->ExecProc();
DataModule2->IBTransaction1->Commit();
DataModule2->person_select->Open();}
//---------------------------------------------------------------------------
void __fastcall TForm4::RadioButton1Click(TObject *Sender)
{DataModule2->person_select->Close();
DataModule2->person_select->ParamByName("VIN_GROUP")->AsInteger=1;
DataModule2->person_select->Open();
DataModule2->edit_person->ParamByName("VIN_GROUP")->AsInteger=1;}
void __fastcall TForm4::RadioButton2Click(TObject *Sender)
{DataModule2->person_select->Close();
DataModule2->person_select->ParamByName("VIN_GROUP")->AsInteger=2;
DataModule2->person_select->Open();
DataModule2->edit_person->ParamByName("VIN_GROUP")->AsInteger=2;}
void __fastcall TForm4::RadioButton3Click(TObject *Sender)
{DataModule2->person_select->Close();
DataModule2->person_select->ParamByName("VIN_GROUP")->AsInteger=3;
DataModule2->person_select->Open();
DataModule2->edit_person->ParamByName("VIN_GROUP")->AsInteger=3;}
//---------------------------------------------------------------------------
```

Unit8.cpp

```
#include <vcl.h>
#pragma hdrstop
#include "Unit8.h"
#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#include "Unit4.h"
```

```
#include "Unit5.h"
#include "Unit6.h"
#include "Unit7.h"
#include "Unit9.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm8 *Form8;
__fastcall TForm8::TForm8(TComponent* Owner)
      : TForm(Owner)
{
}
void __fastcall TForm8::Button1Click(TObject *Sender)
{DataModule2->title_select->Close();
DataModule2->title_select->ParamByName("VIN_SHOWDEL")->AsInteger=0;
DataModule2->title_select->ParamByName("VIN_STARTDATE")-
>AsDate=Form8->DateTimePicker1->Date;
DataModule2->title_select->ParamByName("VIN_ENDDATE")-
>AsDate=Form8->DateTimePicker2->Date;
DataModule2->title_select->ParamByName("VIN_METHOD")->AsInteger=5;
DataModule2->title_select->ParamByName("VIN_VALUE")->AsInteger=0;
DataModule2->title_select->ParamByName("VIN_USERID")->AsInteger=1;
DataModule2->title_select->Open();}
//---------------------------------------------------------------------------
void __fastcall TForm8::DBGrid1DblClick(TObject *Sender)
{Application->CreateForm(__classid(TForm3), &Form3);
DataModule2->person->Close();
DataModule2->person->ParamByName("VID")->AsInteger=DataModule2-
>title_select->FieldByName("VPERSONID")->AsInteger;
DataModule2->person->Open();
Form3->number_doc->Caption=DataModule2->title_select-
>FieldByName("VID")->AsString;
int typedoc;
int statedoc;
typedoc=DataModule2->title_select->FieldByName("VTYPEID")->AsInteger;
statedoc=DataModule2->title_select->FieldByName("VSTATEID")->AsInteger;
switch(typedoc)
{
case 1: Form3->type_doc->Caption="Приход товара (Приходная накладная)";
Form3->type->Caption="1";
DataModule2->person_select->Close();
DataModule2->person_select->ParamByName("VIN_GROUP")->AsInteger=1;
DataModule2->person_select->Open();
DataModule2->goods_select->Close();
```

```
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->ParamByName("VIN_AVAILABLE")-
>AsInteger=0;
DataModule2->goods_select->Open();
break;
case 2: Form3->type_doc->Caption="Продажа товара (Расходная накладная)";
Form3->type->Caption="2";
DataModule2->person_select->Close();
DataModule2->person_select->ParamByName("VIN_GROUP")->AsInteger=2;
DataModule2->person_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->ParamByName("VIN_AVAILABLE")-
>AsInteger=1;
DataModule2->goods_select->Open();
break;
case 3: Form3->type_doc->Caption="Списание товара (Расходная накладная)";
Form3->type->Caption="3";
DataModule2->person_select->Close();
DataModule2->person_select->ParamByName("VIN_GROUP")->AsInteger=3;
DataModule2->person_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->ParamByName("VIN_AVAILABLE")-
>AsInteger=1;
DataModule2->goods_select->Open();
break;
case 4: Form3->type_doc->Caption="Возврат товара (Приходная накладная)";
Form3->type->Caption="4";
DataModule2->person_select->Close();
DataModule2->person_select->ParamByName("VIN_GROUP")->AsInteger=2;
DataModule2->person_select->Open();
DataModule2->goods_select->Close();
DataModule2->goods_select->ParamByName("VIN_CURRID")-
>AsInteger=DataModule2->groop_select->FieldByName("VID")->AsInteger;
DataModule2->goods_select->ParamByName("VIN_AVAILABLE")-
>AsInteger=0;
DataModule2->goods_select->Open();
break;
}
switch(statedoc)
```

```
{case 1: Form3->state_doc->Caption="Сохранён, не проведён";break;
case 2: Form3->state_doc->Caption="Проведён";break;}
if(statedoc==2)
{Form3->DBEdit1->Visible=True;
Form3->Button1->Visible=False;
Form3->DBGrid1->Visible=False;
Form3->DBGrid4->Enabled=False;
Form3->GroupBox2->Visible=False;}
DataModule2->get_content->Close();
DataModule2->get_content->ParamByName("VIN_DOCID")-
>AsInteger=DataModule2->title_select->FieldByName("VID")->AsInteger;
DataModule2->get_content->Open();
float s;
s=0;
DataModule2->get_content->First();
while(!DataModule2->get_content->Eof)
{s=s+DataModule2->get_content->FieldByName("summ")->AsFloat;
 DataModule2->get_content->Next();}
 Form3->itog->Caption=FloatToStr(s);
Form3->ShowModal();
}
//---------------------------------------------------------------------------
void __fastcall TForm8::N1Click(TObject *Sender)
{
int state;
state=DataModule2->title_select->FieldByName("VSTATEID")->AsInteger;
switch(state){
case 1:
DataModule2->edit_title->ParamByName("VIN_ACTION")->AsInteger=1;
DataModule2->edit_title->ParamByName("VIN_ID")->AsInteger=DataModule2-
>title_select->FieldByName("VID")->AsInteger;
DataModule2->edit_title->ParamByName("VIN_PERSON")->AsInteger=0;
DataModule2->edit_title->ParamByName("VIN_STATE")->AsInteger=3;
DataModule2->edit_title->ParamByName("VIN_SUMM")->AsFloat=0;
DataModule2->edit_title->Prepare();
DataModule2->edit_title->ExecProc();
DataModule2->IBTransaction1->Commit();
DataModule2->title_select->Open();  break;
case 2: ShowMessage("Нельзя удалять документ, который ПРОВЕДЁН");
break;
case 3: ShowMessage("Документ уже удалён"); break;
}
}
```