МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра Автоматизированных систем управления

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**Разработка программного обеспечения поддержки процессов учета хранения товаров на складе**

Казакова Никиты Евгеньевича

Направление подготовки

Информатика и вычислительная техника

Руководитель

Терещенко П.В.

Реферат

ТОВАР, УЧЕТ, СКЛАД, ЗАПАС, ПРЕДПРИЯТИЕ, ЗАКАЗ

Проведен анализ работы логистика склада организации, занимающейся продажами отделочных и строительных материалов. Разработана информационная модель системы. Выявлен состав задач системы. Определены необходимые входные и выходные документы. Определен состав пользователей системы. Разработан технологический процесс функционирования системы в автоматизированном режиме. Определены состав и структура файлов базы данных системы. Проведена алгоритмизация, кодирование и отладка основных модулей системы. Разработан интерфейс системы.

В экономическом разделе выполнен SWOT-анализ разработки системы и приведена смета затрат на ее внедрение.

В разделе охраны труда рассмотрены условия хранения медикаментов и расчёт мощности вентилятора.

Содержание

Введение

1. Описание предметной области

1.1 Обзор систем организации управления предприятия

1.2 Характеристика объекта автоматизации

1.3 Описание и схема информационного взаимодействия элементов системы

2. Описание постановки комплекса задач системы

2.1 Общая характеристика задач системы

2.2 Выходная информация

2.3 Входная информация

2.4 Технологические процесс функционирования системы в автоматизированном режиме

2.5 Требования к программно-техническому обеспечению

2.5.1 Комплекс технических средств

2.5.2 Общесистемное программное обеспечение

2.5.3 Выбор и обоснование инструментального средства

3. Разработка информационного обеспечения системы

3.1 Состав и структура таблиц базы данных системы

3.2 Логическая модель взаимосвязи таблиц базы данных системы

3.3 Информационная модель системы

3.4 Описание алгоритмов и программ

3.4.1 Описание алгоритма программного модуля расчёта

гарантийного запаса товаров

3.4.2 Описание алгоритма программного модуля формирование оптимального размера заказа

3.4.3 Описание алгоритма программного модуля формирование

отчета «Объем продаж»

3.4.4 Описание алгоритма программного модуля формирование отчета «Ведомость остатков»

3.4.5 Описание алгоритма программного модуля формирование отчета «Списания»

3.5 Контрольный пример

4. Организационно экономическое обоснование дипломного проекта

4.1 Целесообразность разработки с экономической точки зрения

4.2 SWOT-анализ разработки

4.3 Калькуляция себестоимости научно-технической продукции

4.4. Отчисления на социальные нужды

5. Раздел «Охрана труда»

5.1 Требования безопасности к хранению медикаментов на аптечных складах

5.2 Расчёт мощности вентилятора

Заключение

Список использованных источников

Приложения

Введение

Управление товарными ресурсами в компании предполагает планирование и прогнозирование деятельности предприятия. Проведение анализа большого количества информации по истории продаж, поставок, товарных запасов, списаний и т.д. А также расчёт оптимальных размеров товарных запасов и дальнейшего планирования размеров и номенклатуры заказов поставок необходимых для поддержания эффективного функционирования склада. Оптимизация уровня товарных запасов на предприятии встает в связи проблемой содержания запасов, с одной стороны нужно избежать переполнения товаров на складах, и с другой стороны нежелательно допускать отсутствия необходимого товара длительное время. Одним из наиболее известных и эффективных методов анализа товарных запасов является формула Вильсона. А также распределение товаров на складах в зависимости от стабильности продаж, при помощи методов ABC - анализа.

Объектом автоматизации является ЗАО «Аптека-Холдинг» - крупная компания на рынке фармацевтических препаратов. Компания занимается оптовой и розничной торговлей большой номенклатуры лекарственных средств, охватывающих весь спектр необходимых фармацевтических препаратов.

Целью проектирования является разработка средства автоматизированной информационной поддержки работы логистика склада по поддержке процессов закупа и учёта хранения товаров на складе.

В соответствии с поставленной целью можно сформулировать следующие задачи разработки системы:

* системный анализ работы логистика;
* построение схемы информационного взаимодействия отдела логистики с другими подразделениями предприятия;
* определение состава автоматизируемых задач;
* постановка задач системы;
* разработка технологии функционирования системы в условиях автоматизированного управления;
* определение структуры базы данных системы, построение информационной модели системы;
* построение интерфейса взаимодействия пользователя с системой;
* разработка алгоритмов;
* отладка и тестирование программного обеспечения системы;
* оформление проекта системы, включая подготовку руководства пользователя.

1. Описание предметной области

1.1 Обзор систем организации управления предприятия

В конце 60-х годов, в связи с бурным развитием вычислительной техники, ее возможности перестали быть востребуемы только отдельными наукоёмкими отраслями, компьютерные системы прочно входили в повседневную деловую жизнь. Повсюду начались активные попытки оптимальной автоматизации и информатизации бизнеса, создавались новые концепции управления и совершенствовались уже существующие. Основными целями автоматизации производственных компаний являлись: точный расчет актуальной себестоимости продукции, ее анализ, понижение затрат в процессе производства и повышение производительности в целом, благодаря эффективному планированию производственных мощностей и ресурсов. Результатом оптимизации этих параметров являлись понижение конечной цены готовых изделий и повышение общей производительности, что соответственно немедленно отражалось на конкурентоспособности и рентабельности компании. В результате поиска решений в области автоматизации производственных систем родилась концепция планирования потребностей в материалах (MRP). По сути, MRP-методология представляет собой алгоритм оптимального управления заказами на готовую продукцию, производством и запасами сырья и материалов, реализуемый с помощью компьютерной системы. Другими словами, MRP система позволяла оптимально загружать производственные мощности, и при этом закупать именно столько материалов и сырья, сколько необходимо для выполнения текущего плана заказов и именно столько, сколько возможно обработать за соответствующий цикл производства. Тем самым планирование текущей потребности в материалах позволяло разгрузить склады как и сырья и комплектующих (сырье и комплектующие закупались ровно в том объеме, который можно обработать за один производственный цикл и поступались прямо в производственные цеха), так и склады готовой продукции (производство шло в строгом соответствии с принятым планом заказов, и продукция, относящаяся к текущему заказу, должна быть произведена ровно к сроку его исполнения (отгрузки)). Собственно методология MRP является реализацией двух известных принципов JIT (Just In Time – Вовремя заказать) и KanBan (Вовремя произвести). Разумеется, идеальная реализация концепции MRP невыполнима в реальной жизни. Например, из-за возможности срыва сроков поставок по различным причинам и последующей остановки производства в результате этого. Поэтому в жизненных реализациях MRP-систем на каждый случай предусмотрен заранее определенный страховой запас сырья и комплектующих (safety stock), объем которого определяется компетентным руководством компании.

После появления концепции MRP, казалось бы, все основные проблемы производства были решены, активно создавались и продавались компьютерные программы, реализующие принципы этой концепции. Однако в процессе дальнейшего анализа существующей ситуации в мировом бизнесе и ее развития, выяснилось, что всю большую составляющую себестоимости продукции занимают затраты напрямую не связанные с процессом и объемом производства. В связи с растущей от года к году конкуренцией, конечные потребители продукции становятся все более “избалованными”, ощутимо увеличиваются затраты на рекламу и маркетинг, уменьшается жизненный цикл изделий. Всё это требует пересмотрения взглядов на планирование коммерческой деятельности. Отныне нужно не “что-то производить и стараться потом продать”, а “стараться производить, то, что продается”. Таким образом, маркетинг и планирование продаж должны быть непосредственно связаны с планированием производства.

В 90-х годах методы MRP были модифицированы и улучшены, в связи с этим появились такие системы, как MRP II (Manufacturing Resource Planning) – это планирование производственных ресурсов и ERP (Enterprise Resource Planning) – это планирование потребностей предприятий в материалах. Преимущества, даваемые этими методами, состоят в минимизации издержек, связанных со складскими запасами сырья, комплектующих, полуфабрикатов и прочего, а также с аналогичными запасами, находящимися на различных участках непосредственно в производстве.

Для оптимизации управления логистическими цепочками была создана концепция SCM (Supply Chain Management), которую поддерживает большинство систем класса MRPII. SCM, положенная, как компонент общей бизнес стратегии компании, позволяет существенно снизить транспортные и операционные расходы, путем оптимального структурирования логистических схем поставок.

Мировой опыт показывает, что успеха достигают те компании, которые балансируют производственные, коммерческие и финансовые цели. Рассматривают предприятие как единую производственно-сбытовую систему (ПСС), связывающую воедино такие сферы как: маркетинг – создание новых изделий – снабжение – производство – сбыт – доставка продукции потребителю – сервисное обслуживание, и используют промышленные стандарты MRP/ERP в качестве базовой бизнес - модели, нацеленной на достижение экономической эффективности.

Рассмотрим концепцию MRP II поподробнее, так как на основе этой концепции будет проектироваться дипломный проект.

В системах класса MRP II содержатся следующие функции производственно - сбытовой системы:

* Планирование продаж и производства (Sales and Operation Planning);
* Управление спросом (Demand Management);
* Составление плана производства (Master Production Scheduling);
* Планирование материальных потребностей (MRP - Material Requirement Planning);
* Управление запасами (Inventory Transaction Subsystem);
* Управление плановыми поставками (Scheduled Receipts Subsystem);
* Планирование производственных мощностей (CRP – Capacity Requirement Planning);
* Материально техническое снабжение (Purchasing);
* Планирование ресурсов для распределения (DRP – Distribution Resource Planning);
* Планирование и контроль производственных операций (Tooling Planning and Control);
* Управление финансами (Financial Planning);
* Моделирование для производственной программы (Simulation);
* Оценка результатов деятельности (Performance Measurement).

Управление запасами, эта подсистема обеспечивает реализацию следующих функций:

- Inventory Control – мониторинг запасов;

- Physical Inventory – регулирование и инвентаризация складских остатков.

При решении задач управления запасами - производится обработка и корректировка всей информации о приходе, движении и расходе сырья и материалов, промежуточной продукции и готовых изделий; учет запасов по складским ячейкам, выбор индивидуальных стратегий контроля, пополнения и списания запасов по каждой позиции номенклатуры сырья и материалов, и т.д. Учитывается нормативная и текущая фактическая стоимость запасов, а также отслеживается прохождение отдельных партий запасов и серий изготавливаемой продукции.

Управления снабжением, эта подсистема реализует следующие функции:

Purchase Orders - заказы на закупку;

Supplier Schedules - график поставок;

MRP - планирование потребности в материалах, понимаемое как управление заявками на закупку.

Управление цепочками поставок - эта подсистема предназначена для обеспечения эффективного управления материальными и соответствующими им информационными потоками: от поставщика через производство к потребителю. Реализованная в подсистеме идеология «управления глобальными цепочками поставок» дает промышленным предприятиям возможность представлять свою деятельность в виде так называемых эффективных цепочек логистики: от поставщиков сырья и комплектующих до продажи готовых изделий конечному потребителю. При этом обеспечиваются широкие возможности управления транснациональными компаниями, координации распределенного между многими дочерними компаниями производства.

Результаты использования интегрированных систем стандарта MRP II:

- получение оперативной информации о текущих результатах деятельности предприятия как в целом, так и с полной детализацией по отдельным заказам, видам ресурсов, выполнению планов;

- долгосрочное, оперативное и детальное планирование деятельности предприятия с возможностью корректировки плановых данных на основе оперативной информации;

- решение задач оптимизации производственных и материальных потоков;

- реальное сокращение материальных ресурсов на складах;

- планирование и контроль за всем циклом производства с возможностью влияния на него в целях достижения оптимальной эффективности в использовании производственных мощностей, всех видов ресурсов и удовлетворения потребностей заказчиков;

- автоматизация работ отдела с полным контролем за платежами, отгрузкой продукции и сроками выполнения договорных обязательств;

- значительное сокращение непроизводственных затрат;

- возможность поэтапного внедрения системы, с учетом инвестиционной политики конкретного предприятия.

Применение MRP технологий становится стандартным на большинстве предприятий. Влияние на бизнес качества приложений MRP/ERP огромное. Использование MRP содействует прогнозированию и планированию, что может обеспечить значительное сокращение издержек и улучшение процесса производства. Производители, которые успешно внедрили системы MRP, имеют: уменьшенные складские запасы, уменьшенное время выхода на рынок новых продуктов и конечно же увеличение прибыльности.

На мировом рынке сейчас предлагается свыше 500 систем класса MRP II-ERP. Рынок бурно растет - на 35% - 40% каждый год. В настоящее время в России присутствуют около десятка западных систем, таких как: BAAN IV, SAP R3, ORACLE Applications, QAD MFG/PRO и несколько отечественные информационные системы – СПЕКТР, Практик–А, Тектон, которые можно отнести к корпоративным. Типовая стоимость проекта по внедрению такой системы составляет от 50 до 250 тысяч долларов для тиражно - заказных систем, и до 20 тысяч - для тиражируемых, или «коробочных».

На сегодняшний момент в состав MPR систем входит множество методов, включая методы расчёта оптимальных размеров товарных запасов и их распределения, а также методы расчёта оптимальных размеров заказов. В данной работе мы воспользуемся несколькими из таких методов – формулой расчёта размера оптимальных размеров заказов - формулой Вильсона, и методов АВС и XYZ- анализов для распределения товаров на складе.

1.2 Характеристика объекта автоматизации

ЗАО «Аптека-Холдинг» - крупная фармацевтическая компания, занимающаяся реализацией и производством большого ассортимента лекарственных препаратов. Компания является одним из крупнейших в Сибири дистрибьютором крупных Российских и иностранных заводов по производству фармацевтических материалов. Также ЗАО «Аптека-Холдинг» имеет ряд филиалов в городах Сибирского региона. В компании работают: региональный департамент сбыта, корпоративный департамент сбыта и городской (Новосибирский) департамент сбыта. Статус официальных представителей заводов позволяет вести гибкую ценовую политику с клиентами, формировать большие объемы поставок и варьировать сроки поставок.

Необходимо учитывать, что грамотная постановка задач менеджмента является важнейшим фактором, влияющим как и на успех деятельности предприятия в целом, так и на успех проекта автоматизации. Поэтому, чтобы проект внедрения автоматизированной системы управления оказался удачным - необходимо максимально формализовать все те контуры управления, которые собственно планируются автоматизировать.

Рассмотрим подробнее структуру работы отделов, которые затронет автоматизация предприятия.

Компанией используются несколько больших складских помещения, для обеспечения продаж широкого ассортимента лекарственных препаратов в промышленных, оптовых и розничных объемах. Для функционирования работы складов, работают несколько служб: Департамент сбыта, Департамент поставок и работники складов – логистики, кладовщики и грузчики.

Департамент сбыта – занимается приёмом заказов от департаментов сбыта регионального и корпоративного назначения, а также от частных лиц. При приёме заказа оператор проверяется наличие товаров на складах в нужном объёме и вписывает их в расходную накладную. После формирования заказа, выписываются необходимые документы на получение заказа для клиента. Затем пакет документов отправляется на склад.

Компании необходимо решать проблему оптимизации товарных запасов на складах и оптимальных размеров закупок. С целью наименьших затрат на содержание складов и более быстрого пополнения продукции на складах.

При получении заявки на складе, бухгалтер склада передаёт расходную накладную с перечнем и объёмом заказанных товаров главному кладовщику, который распределяет заказ между кладовщиками, ответственных за соответствующие территории складов и группы товаров. После этого кладовщики, вместе с грузчиками собирают заказ со складов. После того как заказ оформлен и клиент забрал его, бухгалтер склада оформляет соответствующие документы, добавляет соответствующие записи о продаже товаров в базе данных.

Департамент поставок, по базе данных и информации об остатках товаров на складах, следит за уровнем наличия товаров на складе. При достижении уровня страхового запаса, работники департамента поставок формируют заказы на поставку товаров для пополнения складов до необходимого уровня наличия товаров. При заказе товаров департамент поставок, информирует работников складов и департамент сбыта о периоде и объемах поставок товаров на склады.

При приходе товаров на склады от поставщиков, кладовщики вместе с грузчиками распределяют товары по складам, в соответствии с их типом, а бухгалтер по приходной ведомости вносит пришедшие товары в базу данных.

1.3 Описание и схема информационного взаимодействия элементов системы

Объектом автоматизации является процессы учёта, хранения и распределения товаров на складе, а также процесс закупа товаров у поставщиков, в соответствии с оптимизацией товарных запасов. Проведенное исследование данного процесса и выявление информационных потребностей склада, позволило построить информационную модель функционирования системы. Схема информационного взаимодействия подразделений системы представлена на рисунке 1.1.

На рисунке 1.1 введены следующие обозначения:

Ф1 – филиал №1;

Ф2 – филиал №2;

Фn – филиал №n;

Корпоративный ДС – корпоративный департамент сбыта;

Региональный ДС – региональный департамент сбыта;

1 – заявка на формирование заказа от покупателя;

2 – заявка на формирование заказа от филиалов, департаментов сбыта;

3 – расходная накладная;

4 – счёт, счёт-фактура клиенту;

5 – сведения о продажах, поставках и об остатках на складе;

6 – заявка поставщикам на оформление заказа от департамента поставок;

7 – приходная накладная, счет-фактура;

8 – сведения о поставках.

1

*Рисунок 1.1 – Схема взаимосвязи подразделений системы*

Ф1

Ф2

Фn

. . .

Корпоративный

ДС

Региональный ДС

Клиент

СКЛАДЫ

Департамент

поставок

Поставщики

2

5

6

7

8

1

3

3

Департамент

Сбыта

1

2

3

4

2. Описание постановки задач системы

2.1 Общая характеристика задач

На основе анализа функционирования компании можно выделить следующие задачи:

- «Учет прихода товара на склад» (Z1);

- «Учет расхода товара со склада» (Z2);

- «Учет инвентаризации товара» (Z3);

- «Формирование оптимального размера заказов» (Z4);

- «Распределение товара на складе» (Z5);

- «Формирование отчета «Объем продаж за период»» (Z6);

- «Формирование отчета «Ведомость остатков»» (Z7);

- «Ведение справочников системы» (Z8).

Задачу Z1 целесообразно разделить на подзадачи:

- «Добавление записей в справочник «Поставки»» (Z11);

- «Добавление новых видов товаров» (Z12);

- «Добавление новых групп товаров» (Z13);

- «Добавление новых поставщиков» (Z14);

- «Формирование отчета «Приходная ведомость»» (Z15).

Задачу Z2 целесообразно разделить на подзадачи:

- «Учет списанного товара со склада» (Z21);

- «Учет проданного товара со склада» (Z22)

- «Формирование отчета «Ведомость остатков»» (Z23).

Задачу Z3 целесообразно разделить на следующие подзадачи:

- «Введение данных по инвентаризации» (Z31);

- «Формирование отчета «Акт о недостаче» (Z32);

Задачу Z4 можно представить как:

- «Расчёт затрат на хранение товарных запасов» (Z41);

- «Определение гарантийного запаса товаров» (Z42);

- «Формирование оптимального размера заказов» (Z43);

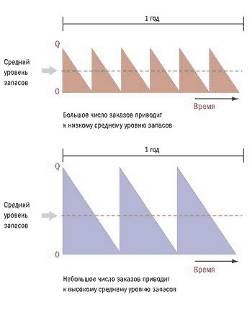
Рассмотрим подробнее каждую из поставленных задач.

Ежедневно на склады поставляется товар различной номенклатуры от разных поставщиков (Z1). Все поставки необходимо фиксировать и заносить сведения в БД (Z11), такие как наименование товара, объем поставки, дата поставки, фасовка, поставщик. Так же периодически привозят новые товары (Z12), относящиеся к новым группам товаров (Z13), от новых поставщиков (Z14). Следовательно, эти позиции необходимо добавить в справочники системы (БД). Для коррекции приходной накладной и проверке введенных данных необходимо сформировать отчет «Приходная ведомость» (Z16).

Со склада идут постоянные отгрузки товаров покупателям (Z2), а также формирование машин для региональных, городских заказчиков и филиалов. Необходимо вести учёт отгруженных товаров (Z21), вносить данные о них в БД, такие как наименование товара, объем отгрузки, дата отгрузки, фасовка, заказчик. Иногда некоторые товары со склада списывают, по причине заводского брака, порчи и невостребованности товара. При этом также заносятся сведения в БД (Z22) о списанных товарах, причине списания, объём списания, дата списания. После занесения данных в БД формируется отчет «Ведомость остатков».

Один раз в год на складе проводится инвентаризация (Z3) товаров на складах. Для корректной работы системы необходимо учитывать результаты инвентаризации и заносить их в БД (Z31). Для сверки «Инвентаризационной ведомости» с данными, хранящимися в БД, необходимо сформировать отчет «Акт о недостаче» (Z32), в зависимости от результатов сопоставления «Инвентаризационной ведомости» с «Ведомостью остатков», сформированной по данным из БД.

Расчёт затрат на хранение товарных запасов (Z41) лежит в основе оптимизации уровня запасов товаров предприятия на плановый период. Для эффективного функционирования складов заказы желательно должны быть большими и их должно быть не много (Рис. 2.1), следовательно нужно достичь наименьших затрат на содержание запасов, путём оптимизации размеров заказов. Критерием оптимизации при этом является, как правило, минимум совокупных затрат, связанных с запасом.



*Рисунок 2.1 Зависимость среднего уровня запасов от размеров заказов.*

В состав общих затрат по созданию и поддержанию запасов входят:

1) затраты на хранение запаса;

2) стоимость размещения заказа;

3) стоимость закупки партии, восполняющей запас, или стоимость заказа.

Затраты на хранение запаса в общем случае имеют прямо пропорциональную зависимость от размера заказа: чем большим количеством заказов пополняется запас, тем дороже стоит содержание запаса.

Общепринятым подходом к расчету среднего уровня запаса является то, что средний уровень запаса при восполнении его партиями по Q единиц равен половине этой величины, то есть Q/2. Следовательно, формула расчёта затрат на хранение будет иметь вид:

T = Q/2\*I + S/Q\*A + C\*S, где (2.1)

T – общие затраты на создание и поддержание запаса;

Q – размер заказа, восполняющего запас;

I – затраты на хранение единицы товара в плановом периоде времени;

S – потребность в запасе в плановом периоде;

A – стоимость размещения одного заказа;

С – цена единицы запаса.

В общем виде общие затраты можно представить как:

T = Затраты на хранение + Стоимость размещения заказа + Цена заказа

Тогда общая стоимость размещения заказа будет рассчитываться следующим образом, по формуле Вильсона:

, где (2.2)

Q\* - оптимальный размер заказа;

I – затраты на хранение единицы товара в плановом периоде времени;

S – потребность в запасе в плановом периоде;

A – стоимость размещения одного заказа;

С – цена единицы запаса.

Когда количество товаров на складе достигает гарантийного запаса (Z42), логистик склада начинает формировать заказ на пополнение заканчивающихся на складе товаров, необходимых для поддержки эффективного функционирования склада. Для подсчёта гарантийного запаса применим следующую формулу:

R = (tn +tз)\* Pдн - P0 где, (2.3)

tn – время поставки;

tз – возможная задержка поставки;

Pдн – ожидаемое дневное потребление товара;

P0 – ожидаемое потребление за время поставки.

Pдн - рассчитывается как, потребность в товаре за период - S/T

При формировании оптимального размера заказов – ОРЗ (Z43), чаще всего работают с многономенклатурными заказами, когда в одной партии поставляются разные виды товаров. Для расчета ОРЗ товара каждого наименования необходимо воспользоваться данной модификацией формулы Вильсона:

 где, (2.4)

Si – потребность в запасе i-го наименования в плановом периоде;

А – стоимость размещения одного заказа;

 – вектор потребностей в запасе различных наименований в плановом периоде времени, включает в себя множество чисел, соответствующее количеству наименований товаров в поставке; например, вектор со значениями (5; 7; 10; 12) соответствует работе с четырьмя наименованиями товаров в одном заказе; при этом в плановом периоде должен быть обеспечен запас товаров первого наименования в объеме 5 единиц, второго - 7 единиц и т. д.;

 – вектор затрат на хранение единицы запаса различных наименований в плановом периоде времени (денежные единицы измерения/единица запаса); включает в себя множество чисел, соответствующее количеству наименований товаров в поставке; например, вектор со значениями (28; 32; 30; 40) соответствует работе с четырьмя наименованиями товаров в одном заказе; при этом затраты на хранение на единицу запаса товара первого наименования составляют 28 единиц, второго - 32 единицы и т. д.;

 – произведение векторов, которое рассчитывается в данном случае как сумма произведений потребности в запасах на плановый период времени и затрат на хранение единицы запасов соответствующего наименования (в рассматриваемом примере: 5\*28+7\*32+10\*30+12\*40 = 1144 единицы).

Также некоторые поставщики дают скидку на большие оптовые закупки. Следовательно, чем больше объём заказа, тем ниже закупочные цены на приобретаемый товар. Каждой цене соответствует определённый диапазон объёма товаров. Для расчёта данного случая, можно применить следующую модификацию формулы Вильсона:

где, (2.5)

I – затраты на хранение единицы товара в плановом периоде времени;

S – потребность в запасе в плановом периоде;

A – стоимость размещения одного заказа;

i – доля цены продукции, приходящейся на затраты по хранению;

С – цена единицы запаса.

Для распределения товара на складе (Z5) воспользуемся методами анализа товарного запаса на складе. Одним из наиболее известных методов является АВС – анализ. Идея АВС – анализа основана на принципе Парето, который формулируется следующим образом: «За большинство возможных результатов отвечает относительное небольшое число причин». В настоящее время этот принцип широко известен как «правило 20 на 80».

АВС – анализ будем проводить по объему продаж за период. В результате получим 3 группы товаров:

- группа А – товар, который лучше всего продается; эта группа составляет 20% ассортимента и 49% общего объема продаж;

- группа В – товар, который хорошо продается; эта группа составляет 30% ассортимента и 30% общего объема продаж;

- группа С – товар этой группы составляет 50% ассортимента и 21% общего объема продаж; в эту группу попадает весь «ассортиментный хвост».

Нередко АВС – анализ рассматривают в совокупности с XYZ – анализом. Основная идея XYZ – анализа состоит в группировании товара по однородности анализируемых параметров, другими словами по коэффициенту вариации.

Коэффициент вариации позволяет сравнить между собой стабильность продаж нескольких товаров, имеющий разный объем продаж.

Коэффициент вариации рассчитывается по формуле (2.6).

, (2.6)

где ν – коэффициент вариации;

σ – среднее квадратическое (стандартное) отклонение;

- среднее значение.

Величина среднего квадратического отклонения позволяет оценить меру рассеивания значений вариантов относительно среднего арифметического. Чем меньше среднее квадратическое отклонение, тем ближе к среднему находится значение. Среднее квадратическое отклонение вариационного ряда рассчитывается по формуле (2.7).

, (2.7)

где  - значение i-го периода;

 - среднее значение за n периодов;

n – количество периодов.

При проведении XYZ – анализа товары группируются по величине коэффициента вариации. В группу Х попадают товары с коэффициентом вариации меньше 10%. В группу Y – товары с коэффициентом вариации от 10% до 25%. И в группу Z – товары с коэффициентом вариации более 25%.

Таким образом, применение XYZ – анализа позволяет разделить весь ассортимент на группы в зависимости от стабильности продаж. По полученным результатам целесообразно провести работу по выявлению и устранению основных причин, влияющих на стабильность и прогнозируемость продаж.

При комплексном анализе состояния системы управления товарными ресурсами наиболее продуктивно является совмещение результатов АВС – и XYZ – анализов.

При совмещении результатов определяется девять групп товаров. Группы товаров представлены в таблице 2.1.

*Таблица 2.1- Группы товаров при совмещении АВС – и XYZ – анализов*

|  |  |  |
| --- | --- | --- |
| AX | AY | AZ |
| BX | BY | BZ |
| CX | CY | CZ |

Товары групп А и В обеспечивают основной товарооборот компании, поэтому необходимо контролировать постоянное их наличие на складе. Товары групп A и B необходимо распределить по складу, таким образом, чтобы доступ к ним был наиболее быстрый и удобный, для более быстрой отгрузки товаров.

Товары группы АХ и ВХ отличает высокий товарооборот и стабильность. Необходимо обеспечить постоянное наличие данного товара, но для этого не требуется создавать избыточный страховой запас, так как расход товаров этой группы стабилен и хорошо прогнозируется.

Товары группы АY и ВY при высоком товарообороте имеют недостаточную стабильность расхода, и, как следствие, для обеспечения постоянного наличия товаров на складе - нужно увеличить страховой запас.

Товары группы AZ и BZ при высоком товарообороте отличаются низкой прогнозируемостью расхода. Попытка обеспечить гарантированное наличие по всем товарам данной группы только за счет избыточного страхового товарного запаса приведет к тому, что средний товарный запас компании значительно увеличиться. По товарам данной группы следует пересмотреть систему заказов. Часть товаров нужно перевести на систему заказов с постоянной суммой (объемом) заказа, по части товаров необходимо обеспечить более частые поставки, выбрать поставщиков, расположенных близко к вашему складу (и снизить тем самым сумму страхового товарного запаса), повысить периодичность контроля.

Товары группы C составляют до 80% ассортимента компании. Применение XYZ - анализа позволяет сильно сократить время, которое менеджер тратит на управление и контроль над товарами данной группы.

По товарам группы СХ можно использовать систему заказов с постоянной периодичностью и снизить страховой товарный запас.

По товарам группы CY можно перейти на систему с постоянной суммой (объемом) заказа, но при этом формировать страховой запас, исходя из имеющихся у компании финансовых возможностей.

В группу товаров CZ попадают все новые товары, товары спонтанного спроса, поставляемые под заказ и т. п. Часть этих товаров можно безболезненно выводить из ассортимента, а другую часть нужно регулярно контролировать, так как именно из товаров этой группы возникают неликвидные или труднореализуемые товарные запасы, от которых компания несет потери. Выводить из ассортимента необходимо остатки товаров, взятых под заказ или уже не выпускающихся, то есть товаров, обычно относящихся к категории стоков.

Итак, использование совмещенного АВС – и XYZ – анализа позволит:

- повысить эффективность системы управления товарными ресурсами;

- повысить долю высокоприбыльных товаров без нарушения принципов ассортиментной политики;

- выявить ключевые товары и причины, влияющие на количество товаров хранящихся на складе;

- выявить приоритеты для размещения товаров на складе;

- перераспределить усилия персонала в зависимости от квалификации и имеющегося опыта.

По проведенным анализам каждому товару присваивается приоритет, на основании которого данный товар «находит» свое место на складе. Склад делится на несколько зон. Товары группы A должны лежать в первой зоне, которая находится ближе к входу. Причем товар группы X имеет преимущество перед товаром группы Y и Z в размещении ближе к входу, к примеру товары группы Х лежат на первом ярусе, а товары группы Y и Z выше. Товар категории B размещается во 2 зоне склада. А товар категории C соответственно размещается в 3 зоне, в самом конце склада. Этот алгоритм позволяет минимизировать перемещения товара, связанные с пополнением ячеек комплектации и подбора товаров для формирования заказов.

Отчеты «Объем продаж за период» (Z6), «Ведомость остатков» (Z7) формируются по запросу пользователя на основе данных из БД о продажах, поступлениях и списаниях товаров со складов по конкретному товару, группе товаров, складу за выбранный период.

Процесс оптимизации хранения товаров на складах основан на использовании справочной информации. Поэтому одной из автоматизируемых задач системы выделена задача – ведение справочников системы (Z8). Справочные системы хранят в себе данные о товарах, группах товаров, фасовках, складах, стеллажах, расположении стеллажей по складам. Справочная информация характеризуется высоким уровнем постоянства, многократностью использования и длительностью хранения. Ее использование позволяет осуществить сокращение объема хранимых сведений, которое обеспечивается в основном за счет исключения дублирования данных.

С учетом вышесказанного на рисунке 2.3 представлена схема, описывающая комплекс задач разрабатываемой системы.

БД

Z1

Д1

Д2

Д3

Z2

Z4

Z7

D1

D2

D3

Z3

Д4

D4

Z6

D5

D6

*Рисунок 2.3 – Схема информационной взаимосвязи задач системы*

2.2 Выходная информация

Выходные документы формируются в процессе работы системы. Перечень этих документов приведен в таблице 2.2. Формы выходных документов содержатся в приложении А.

*Таблица 2.2 – Описание выходных документов*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование  документа | Обозначение  документа | Периодичность формирования | Получатель |
| Приходная ведомость | D1 | По приходу товара на склад | Логистик склада |
| Ведомость  остатков | D2 | По отгрузки или списании товаров и по необходимости | Логистик склада |
| Наименование  документа | Обозначение  документа | Периодичность формирования | Получатель |
| Акт о недостаче | D3 | При проведении инвентаризации | Логистик склада |
| Бланк заказа | D4 | При формировании  заказа | Логистик склада |
| Объем продаж за период | D5 | По необходимости | Логистик склада |
| Списания со склада  за период | D6 | По необходимости | Логистик склада |
| Справочные данные | D7 | По необходимости | Логистик склада |

2.3 Входная информация

В процессе обследования объекта автоматизации была выявлена входная информация, необходимая для решения поставленных задач и получения результатов. Перечень и описание входных документов представлен в таблице 2.3. Формы входных документов приведены в приложении А.

*Таблица 2.3 – Описание входных документов*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование документа | Обозначение документа | | Периодичность  поступления | Источник информации |
| Приходная накладная | Д1 | | При поставке товаров | Поставщики |
| Расходная накладная | Д2 | | При формировании заказа от клиента | Департамент  сбыта |
| Акт на списание | Д3 | | В случае списания товара со склада | Кладовщик |
| Инвентаризационная ведомость | Д4 | При проведении инвентаризации | | Комиссия по инвентаризации |

2.4 Технологические процесс функционирования системы в автоматизированном режиме

Схема, отражающая процесс функционирования системы в автоматизированном режиме, представляет собой обобщенное описание организации технологического процесса, показывает прохождение информации в системе от момента сбора оперативной информации до получения выходной документации, а также последовательность как автоматизированных, так и ручных процедур.

Рассмотрим процесс функционирования системы в автоматизированном режиме. На склад стекается следующая информация: приходная накладная от поставщиков, когда приходит товар на склад (Д1); из департамента сбыта приходит расходная накладная (Д2) при формировании заказа от клиента; от кладовщиков складов в случае списаний со склада приходит акт на списание (Д3); от комиссии по инвентаризации приходит документ «Инвентаризационная ведомость» (Д4) при проведении инвентаризации на складе. Все сведения из указанных документов заносятся в базу данных. Системой осуществляется контроль данных. Если правильность данных не подтверждается, то производится корректировка введенных данных.

Последовательность обработки информации отображена на рисунке 2.4.

Машинная обработка заключатся в формировании бланка заказа товаров, отчетов «Приходная ведомость», «Акт о недостаче», «Объем продаж», «Ведомость остатков». На основании этих расчетов производится учёт товаров на складе и задание на формирование заказа. Результаты машинной обработки заносятся в базу данных, при необходимости отображаются на дисплее и выводятся на принтер.

Дисплей

Поставщики

Департамент сбыта

Кладовщики

Д1

Д2

Д3

Ввод данных

Клавиатура

Дисплей

БД

Контроль ввода данных

Есть ошибка?

да

Машинная обработка

нет

D6

БД

Логистик склада

Комиссия по инвентаризации

Д4

D1

D3

D3

D4

D5

*Рисунок 2.4 – Схема работы системы в автоматизированном режиме*

2.5 Требования к программно-техническому обеспечению

Для нормального функционирования системы выдвигаются следующие требования к программно-техническому обеспечению и комплексу технических средств.

2.5.1 Комплекс технических средств

Для эксплуатации разрабатываемой системы предъявляются следующие

минимальные требования к техническому оснащению:

* объем оперативной памяти 128 Мб;
* объем жесткого диска 20 Гб;
* частота процессора 600 МГц.

Разрабатываемая система должна обеспечивать создание, систематизацию, накопление, хранение и обработку необходимой информации о прошлой, текущей и будущей деятельности предприятия. Поэтому аппаратные средства должны быть рассчитаны на длительную эксплуатацию в условиях увеличения объемов обрабатываемой информации, появления новых средств анализа и представления данных. При выборе подходящей конфигурации также следует учитывать ценовую политику на рынке компьютерной техники (стоимость нового оборудования с улучшенными характеристиками в некоторых случаях меньше стоимости морально устаревшего оборудования).

В связи с этим для работы с системой рекомендуются следующие характеристики технического оснащения:

* + объем оперативной памяти 256 Мб и выше;
  + объем жесткого диска 40 Гб;
  + частота процессора 800-1000 МГц.
  + Дополнительные специальные требования к конфигурации ПК:
  + дисковод 3,5";
  + клавиатура и манипулятор типа «мышь» для управления в программе;
  + принтер формата А4 для печати выходных документов;
  + цветной монитор.

2.5.2 Общесистемное программное обеспечение

Данный программный продукт может функционировать в среде WINDOWS 98/NT/2000/XP и выше. С появлением операционной системы WINDOWS появились широкие возможности для создания программных продуктов. Система WINDOWS обеспечивает многозадачный графический интерфейс пользователя (Graphical User Interface - GUI), который способствует написанию интерактивных программ. Эта система представляет собой тип операционной системы, оптимизированной для взаимодействия человека и машины.

Также для функционирования данного программного продукта необходим программный пакет BDE Administrator для связи между таблицами.

2.5.3 Выбор и обоснование инструментального средства

В качестве инструментального средства для создания программы был выбран пакет C++Builder 6.0 для операционной системы WINDOWS.

C++Builder продукт корпорации Inprise, более известной как Borland International, предназначенный для быстрой разработки приложений (RAD - Rapid Application Development) на языке С++.

C++Builder - мощная система визуального объектно-ориентированного проектирования, позволяющая решать множество задач, в частности:

- создавать законченные приложения под WINDOWS самой различной направленности, от чисто вычислительных и логических, до графических и мультимедиа;

- быстро создавать (даже начинающим программистам) профессионально выглядящий оконный интерфейс для любых приложений, написанных на любом языке;

- интерфейс удовлетворяет всем требованиям WINDOWS и автоматически настраивается на ту систему, которая установлена на компьютере пользователя, поскольку использует многие функции, процедуры, библиотеки WINDOWS;

- формировать и печатать сложные отчеты, включающие таблицы, графики и т.д.;

- создавать справочные системы (.hlp - файлы), как для своих приложений, так и для любых других, с которыми можно работать не только из приложений, но и просто через WINDOWS;

- множество других задач.

С помощью C++Builder можно создавать WINDOWS - программы на С++ быстрее и проще, чем когда-либо ранее. Возможно создавать как консольные приложения Win32, так и использовать графический интерфейс пользователя (GUI - Graphical UserInterface). Это означает, создание интерфейса пользователя (меню, диалоговые окна, кнопки и т.д.), используя технику drag-and-drop. При этом не возникает потерь в скорости выполнения программ, потому что вся мощь языка С++ по-прежнему остается в распоряжении разработчика. C++Builder поддерживает основные принципы объектно-ориентированного программирования - инкапсуляцию, полиморфизм и множественное наследование, а также последние расширения языка С++. Сам по себе язык C++ не является простым даже для профессионала, поэтому в C++Builder многое сделано для того, чтобы скрыть некоторые низкоуровневые детали, которые составляют «внутренности» Windows программ.

C++Builder обеспечивает высокое быстродействие при компиляции и сборке 32-разрядных приложений для современных операционных систем Windows 95/98/NT/XP, включая системы взаимодействия клиент-сервер. Результирующие программы оптимизированы с точки зрения скорости выполнения и затрат памяти. Удобный отладчик (с ассемблерным окном прокрутки, пошаговым исполнением, точками остановки, трассировкой и т.д.) полностью интегрирован в среду C++Builder. Дизайнер форм, редактор кода, инспектор объектов и другие средства остаются доступными во время работы программы, поэтому вносить изменения можно в процессе отладки.

С++Builder поддерживает связь с различными базами данных 3 видов: dBase и Paradox; Sybase, Oracle, InterBase и Informix; Excel, Access, Fох Pro и Btrieve.

Механизм BDE (Borland Database Engine) придает обслуживанию связей с базами данных удивительную простоту и прозрачность. Проводник Database Explorer позволяет изображать связи и объекты баз данных в графическом виде.

Справочная служба C++Builder содержит полное описание каждого управляющего компонента, включая списки свойств и методов, а также многочисленные примеры.

Access 2003 предлагает универсальный набор программных средств, которые обеспечивают широкие возможности для профессиональных разработчиков и вместе с тем могут быть легко освоены новичками. Теперь любой сотрудник может создавать и применять универсальные решения для баз данных, значительно упрощающие организацию, совместное использование данных и доступ к ним.

* + Включение данных из различных источников. Access 2003 поддерживает различные форматы данных, в том числе XML, OLE, ODBC и формат служб Microsoft Windows® SharePoint™ Services.
  + Связи между бизнес-системами. Можно связать таблицы таким образом, чтобы одновременно получать доступ к данным из различных баз, работая с формами, отчетами и страницами доступа к данным в Access 2003. Кроме того, можно связывать таблицы из других баз данных Access, электронных таблиц Microsoft Excel, источников данных ODBC, баз данных Microsoft SQL Server™ и других источников.
  + Максимально эффективное использование корпоративных данных. Можно включить данные Microsoft SQL Server в решения Access. Используйте конструктор сохраненных процедур для создания и изменения простых процедур, сохраняемых в SQL Server.

3. Разработка информационно-программного обеспечения

3.1 Состав и структура таблиц базы данных системы

«Товар» - справочная таблица (см. таблицу 3.1), содержащая перечень товаров, хранящихся на складах. Для каждого товара указаны группа товаров, к которой относится товар, производитель товара, цена товара.

«Наименования товаров» - справочная таблица (см. таблицу 3.2), содержащая перечень товаров, находящихся на складах.

«Группы товаров» - справочная таблица (см. таблицу 3.3), содержащая наименования групп товаров, на которые разбит товар.

«Сотрудники» - справочная таблица (см. таблицу 3.4), содержащая информация о сотрудниках, работающих на складах. Для каждого сотрудника указаны фамилия, имя, отчество, дата рождения, должность, на которой работает сотрудник, дата принятия на должность.

«Должности» - справочная таблица (см. таблицу 3.5), содержащая перечень должностей сотрудников склада.

«Клиенты» - таблица (см. таблицу 3.6), в которой содержится информация по всем клиентам предприятия, включая клиентов от филиалов, корпоративного ДС и городского ДС. Для каждого клиента указаны фамилия, имя, отчество, документ, удостоверяющий личность, организация, должность, контактный телефон, код банка, расчётный счёт, сведения о заказах.

«Продажи» - таблица (см. таблицу 3.7), в которой хранится информация по отгруженным товарам со склада. По каждой расходной накладной формируются строки, в которых указаны наименование товара, отгруженного со склада, дата отгрузки, количество товара, номер документа и сотрудник, отпустивший товар.

«Поставки» - таблица (см. таблицу 3.8), в которой хранится информация по товарам, полученным на склад от поставщиков. Каждый приход товаров отображается в этой таблице со следующей информацией: наименование товара, поступившего на склад, фасовка товара, дата прихода товара, номер приходной накладной, объем прихода и сотрудник, принявший товар.

«Заказы» - таблица (см. таблицу 3.9), в которой хранится информация по всем заказам. По каждому заказу указаны дата заказа, дата поставки, объём заказа, наименования товаров, цена по каждой группе товаров, общая цена заказа, сотрудник оформивший заказ.

«Поставщики» - справочная таблица (см. таблицу 3.10), содержащая информацию о поставщиках товаров работающих с компанией. Для каждого поставщика указано наименование поставщика, адрес, телефон для связи, контактное лицо, код банка, расчётный счёт, сведения о поставках.

«Банки» - таблица (см. таблицу 3.11), в которой содержится информация о банках работающих с компанией. Для каждого банка указаны его название, контактное лицо, контактный телефон, расчетный счёт, ИНН, БИК, КПП, и адрес.

«Списания» - таблица (см. таблицу 3.12) отображает все списания товаров со склада: какой товар был списан, в какой фасовке, когда, номер документа на списание, ответственное лицо и причина списания.

«Причины списания товаров» - справочная таблица (см. таблицу 3.13), содержащая перечень причин списания товаров со складов.

«Инвентаризация» - таблица (см. таблицу 3.14), содержащая информацию о проведенных инвентаризациях. По каждой инвентаризации формируются записи в эту таблицу. Записываются перечень инвентаризуемых товаров с указанием их наименования и количества.

*Таблица 3.1 – Структура таблицы «Товар»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Код товара | Id\_tov | Счетчик |  |
| Код группы товара | Id\_group | Длинное целое число |  |
| Код поставщика | Id\_post | Длинное целое число |  |
| Цена товара | Zena\_tov | Действительное число |  |
| Затраты на хранение товара | Zatr | Действительное число |  |
| Стоимость размещения единицы в заказе | Stoim | Действительное число |  |

*Таблица 3.2 – Структура таблицы «Наименования товаров»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Код товара | Id\_tov | Счетчик |  |
| Наименование товара | Tov | Строка | 100 |

*Таблица 3.3 – Структура таблицы «Группы товаров»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Код группы товаров | Id\_group | Счетчик |  |
| Наименование группы товаров | Group | Строка | 50 |

*Таблица 3.4 – Структура таблицы «Сотрудники»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Код сотрудника | Id\_sotrud | Счетчик |  |
| Фамилия сотрудника | Surname | Строка | 30 |
| Имя сотрудника | Name | Строка | 30 |
| Отчество сотрудника | Patronymic | Строка | 30 |
| Дата рождения сотрудника | Birth | Дата |  |
| Код должности | Id\_dolj | Длинное целое число |  |
| Дата приема на работу | Date | Дата |  |

*Таблица 3.5 – Структура таблицы «Должности»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Код должности | Id\_dolj | Счетчик |  |
| Наименование должности | Dolj | Строка | 50 |

*Таблица 3.6 – Структура таблицы «Клиенты»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Код клиента | Id\_klient | Счетчик |  |
| Фамилия клиента | Surname | Строка | 30 |
| Имя клиента | Name | Строка | 30 |
| Отчество клиента | Patronymic | Строка | 30 |
| Название организации | Org | Строка | 30 |
| Код должности | Id\_dolj | Длинное целое число |  |
| Расчётный счёт | Schet | Действительное число |  |
| Контактный телефон | Tel | Действительное число |  |
| Заказы клиента | N\_zak | Действительное число |  |

*Таблица 3.7 – Структура таблицы «Продажи»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Номер записи | N | Счетчик |  |
| Код товара | Id\_tov | Длинное целое число |  |
| Дата продажи товара | Date | Дата |  |
| Номер расходной накладной | N\_doc | Строка | 10 |
| Количество товара | Kolvo | Действительное число |  |
| Код сотрудника, отпустившего товар | Id\_sotrud | Длинное целое число |  |

*Таблица 3.8 – Структура таблицы «Поставки»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Номер записи | N | Счетчик |  |
| Код товара | Id\_tov | Длинное целое число |  |
| Дата поставки товара | Date | Дата |  |
| Номер приходной накладной | N\_doc | Строка | 10 |
| Количество поставленного товара | Amount | Действительное число |  |
| Код сотрудника, принявший товар | Id\_sotrud | Длинное целое число |  |
| Код поставщика | Id\_post | Длинное целое число |  |

*Таблица 3.9 – Структура таблицы «Заказы»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Номер записи | N | Счетчик |  |
| Код товара | Id\_tov | Длинное целое число |  |
| Дата заказа товара | Date | Дата |  |
| Номер бланка заказа | N\_zak | Строка | 10 |
| Код поставщика | Id\_post | Длинное целое число |  |
| Количество товара | Kolvo | Действительное число |  |
| Код сотрудника, отпустившего товар | Id\_sotr | Длинное целое число |  |
| Стоимость заказа | Zena | Действительное число |  |
| Затраты на создание заказа | Zatr\_zak | Действительное число |  |

*Таблица 3.10 – Структура таблицы «Поставщики»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Код поставщика | Id\_post | Счетчик |  |
| Наименование поставщика | Post | Строка | 50 |
| Контактное лицо | FIO | Строка | 50 |
| Телефон для связи | Tel | Строка | 25 |
| Код банка | Id\_bank | Длинное целое число |  |
| Расчётный счёт | Schet | Действительное число |  |
| Среднее время поставки | Tpost | Действительное число |  |
| Среднее время задержки поставки | Zpost | Действительное число |  |

*Таблица 3.11 – Структура таблицы «Банки»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Код банка | Id\_bank | Счетчик |  |
| Наименование банка | Bank | Строка | 50 |
| Контактное лицо | FIO | Строка | 50 |
| Телефон для связи | Tel | Строка | 25 |
| Расчётный счёт | Schet | Действительное число |  |
| ИНН | Inn | Действительное число |  |
| БИК | Bik | Действительное число |  |
| КПП | Kpp | Действительное число |  |

*Таблица 3.12 – Структура таблицы «Списания»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Номер записи | N | Счетчик |  |
| Код товара | Id\_tov | Длинное целое число |  |
| Дата списания товара | Date | Дата |  |
| Номер акта на списание | N\_doc | Строка | 10 |
| Количество списанного товара | Kolvo | Действительное число |  |
| Код сотрудника, списавшего товар | Id\_sotrud | Длинное целое число |  |
| Код причины списания товара | Id\_reason | Длинное целое число |  |

*Таблица 3.13 – Структура таблицы «Причины списания товара»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Код причины списания товара | Id\_reason | Счетчик |  |
| Причина списания товара | Reason | Строка | 150 |

*Таблица 3.14 – Структура таблицы «Инвентаризация»*

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование поля | Обозначение | Тип данных | Размер |
| Номер записи | N | Счетчик |  |
| Дата | Date | Дата |  |
| Код товара | Id\_tov | Длинное целое число |  |
| Количество товара | Kolvo | Действительное число |  |
| Поле для примечаний | Note | Строка | 200 |

3.2 Логическая модель взаимосвязи таблиц базы данных системы

Схема структуры базы данных системы взаимосвязи таблиц по ключевым признакам представлена на рисунке 4.1.

Tovari

Id\_tov (FK)

Id\_group (FK)

Id\_post (FK)

Zena\_tov

Zatr\_tov

Potr\_tov

Naimenov\_tov

Id\_tov

Tov

Group\_tov

Id\_group

Group

Sotrudniki

Id\_sotrud

Id\_dolj (FK)

Surname

Name

Patronymic

Birth

Date

Klienti

Id\_klient

Id\_dolj (FK)

Id\_bank (FK)

Surname

Name

Patronymic

Org

Tel

Schet

N\_zak

Doljnosti

Id\_dolj

Dolj

Prodagi

Id\_sotrud (FK)

Id\_tov (FK)

Id\_klient (FK)

Date

Kolvo

Postavki

Id\_tov (FK)

Id\_sotrud (FK)

Id\_post (FK)

N

Date

N\_doc

Kolvo

Zakazi

Id\_tov (FK)

Id\_post (FK)

Id\_sotrud (FK)

N

Date

Kolvo

Zena

N\_zak

Zatr\_zak

Postavshiki

Id\_post

Id\_bank (FK)

Post

FIO

Tel

Schet

Banki

Id\_bank

Bank

FIO

Tel

Schet

Inn

Bik

Kpp

Spisaniya

Id\_tov (FK)

Id\_sotrud (FK)

Id\_reason (FK)

N

Date

Kolvo

N\_doc

Prichini\_spisaniya

Id\_reason

Reason

Inventorizaciya

Id\_tov (FK)

N

Date

Kolvo

Note

*Рисунок 3.1 - Логическая модель информационной системы*

3.3 Информационная модель системы

С помощью информационной модели можно наглядно представить состав информационного обеспечения, а также взаимосвязь между входной, выходной и промежуточной информацией системы. Информационная модель системы представлена на рисунке 3.2.

На рисунке 4.2 введены следующие обозначения:

Т1 - справочная таблица «Товары»;

Т2 - справочная таблица «Наименования товаров»;

Т3 - справочная таблица «Группы товаров»;

Т4 - справочная таблица «Сотрудники»;

Т5 - справочная таблица «Должности»;

Т6 - справочная таблица «Клиенты»;

Т7 - таблица «Продажи »;

Т8 - таблица «Поставки»;

Т9 - таблица «Заказы»;

Т10 - справочная таблица «Поставщики»;

Т11 - справочная таблица «Банки»;

Т12 - таблица «Списания»;

Т13 - справочная таблица «Причины списания товаров»;

Т14 - таблица «Инвентаризация».

Входные документы Д1-Д4 описаны в таблице 2.4, выходные документы D1-D6 представлены в таблице 2.3.

Т1

Т2

Т3

Т4

Т5

Т7

Т10

Т12

Т13

Т14

Д1

Д2

Д3

Д4

D7

D6

D1

D2

D3

Т8

Т6

Т11

Т9

D4

D5

*Рисунок 3.2 – Схема информационной модели системы*

3.4 Описание алгоритмов и программ

3.4.1 Описание алгоритма программного модуля расчёт гарантийного запаса товаров

НАЧАТЬ алгоритм программного модуля расчёта гарантийного запаса товаров

ОТОБРАЗИТЬ текущую дату

ОРГАНИЗОВАТЬ меню выбора групп товаров

ЕСЛИ не выбрана группа товаров

ВЫВОД сообщения: «Выберите группу товаров для проведения

анализа»

ИНАЧЕ

GZ.gr = код выбранной группы товаров

ОТКРЫТЬ файл Prodagi

ОТКРЫТЬ файл Postavshiki

ОТКРЫТЬ файл Report\_Remainder

ОТКРЫТЬ вспомогательную таблицу GZ

УДАЛИТЬ устаревшую информацию

ОТКРЫТЬ файл Tovary

УСТАНОВИТЬ фильтр с условием: Tovary.group = GZ.gr

ЕСЛИ конец файла

ВЫВОД сообщения: «В справочнике Товары отсутствует информация по товарам группы < GZ.gr >»

АВАРИЙНЫЙ\_ВЫХОД

К\_Е

ЦИКЛ пока не конец файла Tovary

ПЕРЕЙТИ в рабочую область файла Prodagi

УСТАНОВИТЬ фильтр с условием:

*Рисунок 3.3 – Алгоритм расчёт гарантийного запаса товаров*

Prodagi.id\_tov = коду текущего товара

&& Prodagi.date <= dr && Prodagi.date >= dr-dt

S = 0

ЦИКЛ пока не конец файла Prodagi

S = S + Prodagi.kolvo

К\_Ц

PDN = S/dt

ПЕРЕЙТИ в рабочую область файла Postavshiki

УСТАНОВИТЬ фильтр с условием:

Postavhiki.id\_post = код поставщика текущего товара

FGZ = (Post.tpost+ Post.zpost)\* PDN – PDN\* Post.tpost

ПЕРЕЙТИ в рабочую область файла GZ

ДОБАВИТЬ запись в таблицу GZ

GZ.id\_tov = коду текущего товара

GZ.kolvo = S

ОТМЕНИТЬ фильтр

ОТМЕНИТЬ фильтр

К\_Ц // с переходом на следующую запись

ПЕРЕЙТИ в рабочую область файла Report\_Remainder

ЕСЛИ Report\_Remainder.kolvo <= GZ.kolvo

ВЫВОД сообщения: «Для товара < GZ.gr > необходимо пополнить запас. Запустить формирование заказа?»

ЕСЛИ выбран пункт меню да

ЗАПУСТИТЬ алгоритм формирования заказа

К\_Е

К\_Е

ЗАКРЫТЬ файл Tovari

ЗАКРЫТЬ файл Report\_Remainder

ЗАКРЫТЬ файл Prodagi

ПЕРЕЙТИ в рабочую область файла GZ

УПОРЯДОЧИТЬ записи в порядке убывания поля GZ.kolvo

ЗАКРЫТЬ файл GZ

К\_Е

К\_Е

ВЫВОД файла АВС

ЗАКРЫТЬ файл АВС

КОНЕЦ\_АЛГОРИТМА

3.4.2 Описание алгоритма программного модуля формирование оптимального размера заказа

НАЧАТЬ алгоритм программного модуля формирование оптимального размера заказа

ОТОБРАЗИТЬ текущую дату

ОРГАНИЗОВАТЬ меню выбора планового периода

ОРГАНИЗОВАТЬ меню выбора группа товаров для пополнения запасов

ЕСЛИ не выбрана группа товаров

ВЫВОД сообщения: «Выберите группу товаров для проведения анализа»

ИНАЧЕ

ORZ.gr = код выбранной группы товаров

ОТКРЫТЬ файл Prodagi

ОТКРЫТЬ файл Zakazi

ОТКРЫТЬ вспомогательную таблицу ORZ

УДАЛИТЬ устаревшую информацию

ОТКРЫТЬ файл Tovary

УСТАНОВИТЬ фильтр с условием: Tovary.group = ORZ.gr

ЕСЛИ конец файла

ВЫВОД сообщения: «В справочнике Товары отсутствует информация по товарам группы < ORZ.gr >»

АВАРИЙНЫЙ\_ВЫХОД

К\_Е

ЦИКЛ пока не конец файла Tovary

ПЕРЕЙТИ в рабочую область файла Prodagi

УСТАНОВИТЬ фильтр с условием:

Prodagi.id\_tov = коду текущего товара

&& Prodagi.date <= dr && Prodagi.date >= dr-dt

S = 0

ЦИКЛ пока не конец файла Prodagi

S = S + Prodagi.kolvo

К\_Ц

ПЕРЕЙТИ в рабочую область файла ORZ

ДОБАВИТЬ запись в таблицу ORZ

ORZ.id\_tov = коду текущего товара

ORZ.kolvo = S

ОТМЕНИТЬ фильтр

К\_Ц // с переходом на следующую запись

ПЕРЕЙТИ в рабочую область файла Tovari

УСТАНОВИТЬ фильтр с условием:

Tovari.id\_tov = коду текущего товара

Q=S\*

ДОБАВИТЬ запись в таблицу Zakazi

Zakazi.id\_tov = коду текущего товара

Zakazi.date = dr

Zakazi.id\_post = коду текущего поставщика

Zakazi.kolvo = Q

Zakazi.zena = Q\*C

ОТМЕНИТЬ фильтр

ЗАКРЫТЬ файл Tovari

ЗАКРЫТЬ файл Prodagi

К\_Е

К\_Е

ВЫВОД файла Zakazi

ЗАКРЫТЬ файл Zakazi

КОНЕЦ\_АЛГОРИТМА

3.4.3 Описание алгоритма программного модуля формирование отчета «Объем продаж»

НАЧАТЬ алгоритм программного модуля формирования отчета «Объем продаж»

ОРГАНИЗОВАТЬ ввод периода формирования отчета и установки фильтра

по группам товаров

dn = дата начала периода

dk = дата конца периода

ОТКРЫТЬ файл Prodagi

ОТКРЫТЬ файл Report\_Prodagi

УДАЛИТЬ устаревшую информацию

ОТКРЫТЬ файл Tovari

УСТАНОВИТЬ фильтр с условием: Tovari.group = выбранной группе

ЕСЛИ конец файла

ВЫВОД сообщения: «В справочнике Товары отсутствует информация по товару из группы <выбранная группа товаров>»

АВАРИЙНЫЙ\_ВЫХОД

К\_Е

ЦИКЛ пока не конец файла Tovari

ПЕРЕЙТИ в рабочую область файла Prodagi

УСТАНОВИТЬ фильтр с условием: Prodagi.date >= dn && Prodagi.date <= dk

&& Prodagi.id\_tov = коду текущего имени товара

Vprod = 0

ЦИКЛ пока не конец файла Sale

Vprod = Vprod + Prodagi.kolvo

К\_Ц // с переходом на следующую запись

ДОБАВИТЬ строку в таблицу Report\_Sale

Report\_Prodagi.id\_group = Tovari.id\_group

Report\_Prodagi.id\_tov = Tovari.id\_tov

Report\_Prodagi.amount = Vprod

ОТМЕНИТЬ фильтр

К\_Ц // с переходом на следующую запись

ОТМЕНИТЬ фильтр

ЗАКРЫТЬ файлы Report\_Prodagi, Prodagi

КОНЕЦ\_АЛГОРИТМА

3.4.4 Описание алгоритма программного модуля формирование отчета «Ведомость остатков»

НАЧАТЬ алгоритм программного модуля формирования отчета «Ведомость остатков»

ОРГАНИЗОВАТЬ ввод периода формирования отчета и установки фильтра по группам товаров

dn = дата начала периода

dk = дата конца периода

ОТКРЫТЬ файлы Tovari, Prodagi, Spisaniya

ОТКРЫТЬ файл Report\_Remainder

УДАЛИТЬ устаревшую информацию

ОТКРЫТЬ файл Tovari

ЕСЛИ установлен фильтр по группе товаров

УСТАНОВИТЬ фильтр с условием: Tovari.group = выбранной группе

К\_Е

ЕСЛИ конец файла

ВЫВОД сообщения: «В справочнике Товары отсутствует информация по товару из группы <выбранная группа товаров»

АВАРИЙНЫЙ\_ВЫХОД

К\_Е

ЦИКЛ пока не конец файла Tovari

ПЕРЕЙТИ в рабочую область файла Prodagi

УСТАНОВИТЬ фильтр с условием:

Sale.date >= dn && Prodagi.date = dk &&

Prodagi.id\_tov = коду текущего имени товара

Vprod = 0

ЦИКЛ пока не конец файла Tovari

Vprod = Vprod + Prodagi.kolvo

К\_Ц // с переходом на следующую запись

ОТМЕНИТЬ фильтр

ПЕРЕЙТИ в рабочую область файла Postavki

УСТАНОВИТЬ фильтр с условием:

Postavki.date = dn && Postavki.date = dk &&

Postavki.id\_tov = коду текущего имени товара

ЦИКЛ пока не конец файла Postavki

Vprod = Vprod - Postavki.kolvo

К\_Ц // с переходом на следующую запись

ОТМЕНИТЬ фильтр

ПЕРЕЙТИ в рабочую область файла Spisaniya

УСТАНОВИТЬ фильтр с условием:

Spisaniya.date = dn && Spisaniya.date = dk &&

Spisaniya.id\_tov = коду текущего имени товара

ЦИКЛ пока не конец файла Spisaniya

Vprod = Vprod - Spisaniya.kolvo

К\_Ц // с переходом на следующую запись

ОТМЕНИТЬ фильтр

ДОБАВИТЬ строку в таблицу Report\_Prodagi

Report\_ Remainder.id\_group = Tovari.id\_group

Report\_ Remainder.id\_tov = Tovari.id\_tov

Report\_ Remainder.kolvo = Vprod

К\_Ц // с переходом на следующую запись

ОТМЕНИТЬ фильтр

ЗАКРЫТЬ файлы Report\_Remainder, Tovari, Postavki, Spisaniya

КОНЕЦ\_АЛГОРИТМА

3.4.5 Описание алгоритма программного модуля формирование отчета «Списания товаров»

НАЧАТЬ алгоритм программного модуля формирования отчета «Списания товаров»

ОРГАНИЗОВАТЬ ввод периода формирования отчета и установки фильтра

по группам товаров

dn = дата начала периода

dk = дата конца периода

ОТКРЫТЬ файл Spisaniya

ОТКРЫТЬ файл Report\_ Spisaniya

УДАЛИТЬ устаревшую информацию

ОТКРЫТЬ файл Tovari

ЕСЛИ установлен фильтр по группе товаров

УСТАНОВИТЬ фильтр с условием: Tovari.group = выбранной группе

К\_Е

ЕСЛИ конец файла

ВЫВОД сообщения: «В справочнике Товары отсутствует информация по товару из группы <выбранная группа товаров>»

АВАРИЙНЫЙ\_ВЫХОД

К\_Е

ЦИКЛ пока не конец файла Tovari

ПЕРЕЙТИ в рабочую область файла Write\_off

УСТАНОВИТЬ фильтр с условием:

Spisaniya.date >= dn && Spisaniya.date <= dk &&

Spisaniya.id\_name = коду текущего имени товара

Vprod = 0

ЦИКЛ пока не конец файла Spisaniya

Vprod = Vprod + Spisaniya.kolvo

К\_Ц // с переходом на следующую запись

ДОБАВИТЬ строку в таблицу Report\_Spisaniya

Report\_ Spisaniya.id\_group = Tovari.id\_group

Report\_ Spisaniya.id\_tov = Tovari.id\_tov

Report\_ Spisaniya.kolvo = Vprod

ОТМЕНИТЬ фильтр

К\_Ц // с переходом на следующую запись

ОТМЕНИТЬ фильтр

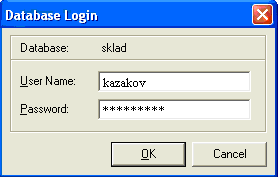
ЗАКРЫТЬ файлы Report\_ Spisaniya, Spisaniya

КОНЕЦ\_АЛГОРИТМА

*Продолжение Рисунка 3.7*

3.5 Контрольный пример

При запуске программы активируется диалоговое окно, представленное на рисунке 3.8, позволяющее идентифицировать пользователя. Если такой пользователь существует, то далее следует запуск главное окно системы, иначе выводится сообщение о том, что доступ запрещен и окно для ввода логина и пароля активируется снова.



*Рисунок 3.8 – Экранная форма идентификации пользователя*

Диалог системы и пользователя организован на основе меню. Меню системы представлено на рисунке 3.9.

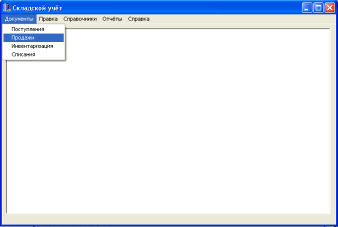
Главное меню программы содержит пять основных разделов: «Документы», «Правка», «Справочники», «Отчёты», «Справка». Рассмотрим подробнее каждый из них.

В разделе «Документы» выделены следующие подразделы:

* «Поступления»;
* «Продажи»;
* «Списания»;
* «Инвентаризация».

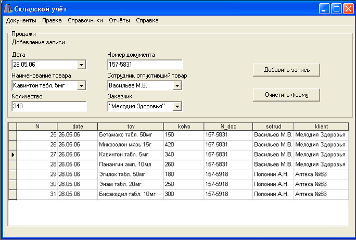
Эти подразделы предназначены для ввода документов «Приходная накладная», «Расходная накладная», «Акт на списание» и «Инвентаризационная ведомость».

Рассмотрим их работу на примере подпункта меню «Расход».



*Рисунок 3.9 – Главное меню программы*

Окно работы программы разделено на 2 части (см. рисунок 3.10). Первая часть предназначена для добавления новых записей на основе документа расходная накладная. Она имеет три поля для выбора наименования товара, заказчика, сотрудника, ответственного за отгрузку товара из справочников «Наименования товаров», «Сотрудники», «Клиенты». Так же эта часть окна предоставляет выбор даты продажи товара, ввод номера документа и количества проданного товара. На данной форме имеется две кнопки управления: «Очистить форму» и «Добавить запись». При нажатии кнопки «Очистить форму» в форме добавления записи очищаются все выбранные записи. При нажатии кнопки «Добавить строку» в базу данных добавляется новая строка и в нее записываются выбранные значения из бланка добавления записи. В другой части окна мы имеем возможность просмотра внесённых изменений в таблицу базы данных «Продажи».



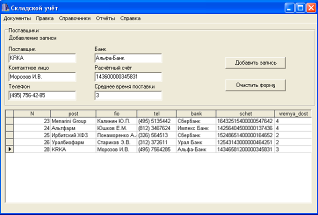
*3.10 - Экранная форма «Продажи»*

Рассмотрим раздел меню «Справочники». В этом разделе предоставляется

доступ к справочникам базы данных. Раздел «Справочники» разделен на следующие подразделы:

* «Товар»;
* «Наименования товаров»;
* «Группы товаров»;
* «Сотрудники»;
* «Должность»;
* «Клиенты»;
* «Поставщики»;
* «Банки»;
* «Причины списания».

Подпункты пункта меню «Справочники» предназначены для просмотра и редактирования справочников базы данных. Для этого предусмотрено два поля (см. рисунок 3.11): первое – добавление данных в справочник, второе – просмотр справочника системы.

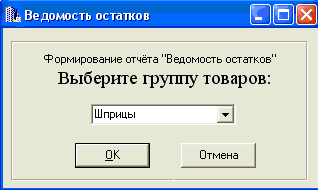


*Рисунок 3.11 – Экранная форма «Поставщики»*

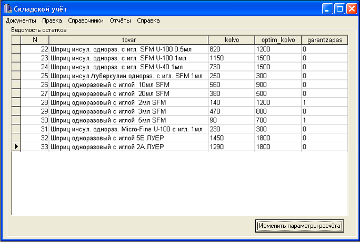
В пункте меню «Отчеты» представлены все отчеты системы. К ним относят:

* «Бланк заказа»;
* «Объем продаж»;
* «Ведомость остатков»;
* «Списания».

При выборе пункта меню «Ведомость остатков» появляется окошко (см. рисунок 3.12), предлагающее выбрать группу товаров для вывода таблицы ведомости остатков. При нажатии кнопки «Отмена» управление программой переходит в главное меню. При нажатии кнопки «ОК» программа начинает формирование ведомости по введенным параметрам.



*Рисунок 3.12 – Экранная форма задания параметров на формирование «Ведомости остатков»*

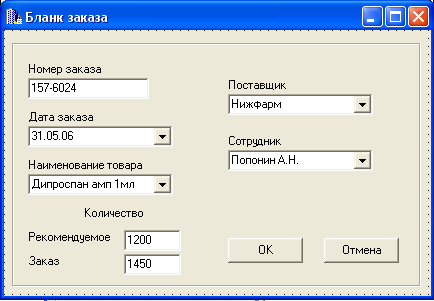


*Рисунок 3.13 – Экранная форма «Ведомость остатков»*

Результаты формирования «Ведомости остатков» представлены на рисунке 3.13. На этой экранной форме имеется кнопка управления «Изменить параметры расчета». При нажатии этой кнопки программа открывает диалог задания параметров для формирования отчёта «Ведомость остатков».

В столбце optim\_kolvo таблицы «Ведомость остатков» указывается оптимальное количество товаров, рассчитанное исходя из затрат на хранение данного товара на складе. Значение 1 в столбце garantzapas показывает, что количество товара на данный момент меньше либо равно величине гарантийного запаса. Это означает, что пользователь увидев значение 1 в столбце должен пополнить запасы данного товара.

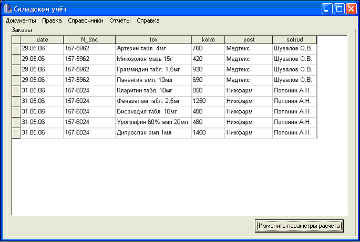
При выборе пункта меню «Бланк заказа» появляется окошко (см. рисунок 3.12), предлагающее ввести пользователю номер заказа и количество товаров в заказе, а также выбрать дату заказа, наименование товара, поставщика и сотрудника оформившего заказ для вывода таблицы заказы.



*Рисунок 3.12 – Экранная форма «Бланк заказа»*

В графе количество есть 2 параметра: рекомендуемое количество и количество заказа. Рекомендуемое количество было рассчитано программой исходя из данных об остатках товара, стоимости хранения, спроса на товар, среднего времени поставки, среднего времени задержки поставки товара. Количество заказа показывает сколько пользователь решил заказать товаров, это количество может быть равным, большим или меньшим чем рекомендуемое, в зависимости от поставленных пользователем целей.

Результаты формирования таблицы «Заказы» представлены на рисунке 3.13. На этой экранной форме имеется кнопка управления «Изменить параметры расчета». При нажатии этой кнопки программа открывает диалог задания параметров для формирования «Бланка заказа».



*Рисунок 3.13 – Экранная форма «Заказы»*

4. Организационно экономическое обоснование дипломного проекта

4.1 Целесообразность разработки с экономической точки зрения

Разработка программы, которая направлена на снижение трудоемкости работы менеджера отдела логистики аптечного склада. Заключается в автоматизации отдела логистики, например работа с договорами, документами учета и хранения товаров. Система реализует быстрый поиск и формирование отчетов и документов, что значительно снижает время и позволяет более оперативно получать необходимые сведения. Так же были разработаны специальные справочники, которые позволяют снизить вероятность ошибки при вводе.

В результате внедрения разработки повышается оперативность и качество принимаемых решений, повышается надежность, улучшаются условия труда, обеспечение сопровождения.

4.2 SWOT-анализ разработки

Название данного метода анализа представляет собой аббревиатуру английских слов Strength (сила), Weakness (слабость), Opportunities (возможности), Threats (угрозы).

Сильные стороны:

* низкая стоимость разработки;
* многофункциональность (быстрый отчёт, поиск информации, упрощенное ведение документации);
* обеспечение сопровождения.

Слабые стороны:

* взаимодействие с другим ПО, в том числе и бухгалтерских.

Возможности:

* получение некоторых сведений по сети;
* расширение круга пользователей.

Угрозы:

* изменение методов ведения работы,
* выход системы из строя.

*Таблица 4.1 – SWOT-матрица*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сильные  стороны | Возможности | | Угрозы. | | Итого |
| Получение некоторых сведений по сети | Расширение круга пользователей | Изменение методов ведения работы | Выход системы из строя |
| Низкая стоимость разработки | 0 | +1 | 0 | 0 | +1 |
| Многофункциональность | +2 | 0 | +1 | 0 | +3 |
| Обеспечение сопровождения | +2 | 0 | +2 | +2 | +5 |
| Итого | +4 | +3 | +3 | +2 | +12 |
| Слабые  стороны |  | |  | |  |
| Взаимодействие с другим ПО | - | - | 0 | - | -3 |
| Покупка пакета 1С | 0 | 0 | 0 | 0 | 0 |
| Итого | -1 | -1 | 0 | -1 | -3 |
| Общий итог | +3 | +2 | +3 | +1 | +9 |

Проанализировав полученную SWOT-матрицу, можно сделать следующие выводы:

* Наиболее важным достоинством является обеспечение сопровождения. В дальнейшем необходимо обращать особое внимание на обеспечение и расширение этой стороны разработки;
* Все выделенные слабые стороны разработки являются очень опасными. И, тем не менее, при правильном подходе они – разрешимы.
* Из рассмотренных возможностей более реальной представляется возможность получение некоторых сведений по сети, хотя данная возможность при существующих слабостях весьма проблематична. Совершенствование же разработки при недостатке финансирования и нехватке кадров вообще невозможно.
* Наиболее опасной угрозой представляется изменение методов ведения работы, но при сложившихся условиях это не столь существенная угроза. Эта угроза представляется маловероятной вследствие налаженной работы отдела.

4.3 Калькуляция себестоимости научно-технической продукции

*Таблица 4.2 – Материалы*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование материальных затрат | Ед. изм. | Кол-во | Цена без НДС с учетом комиссионных вознаграждений, таможенных пошлин и транспортных затрат | Сумма |
| USB – Flash носитель | шт. | 1 | 1000 | 1000 |
| Бумага писчая ZOOM, пачка 500 листов | шт. | 1 | 270 | 250 |
| Картридж для принтера Epson | шт. | 1 | 200 | 200 |
| Канцтовары | шт. | 5 | 15 | 70 |
| Итого: |  |  |  | 1625 |

*Таблица 4.3 – Оценка трудоемкости разработки*

|  |  |  |
| --- | --- | --- |
| № | Наименование этапа | Трудоемкость этапа, часы |
| 1 | Анализ задания и знакомство с темой | 12 |
| 2 | Изучение топологии сети и подбор литературы | 22 |
| 3 | Изучение литературы | 60 |
| 4 | Составление и согласование проекта возможной перестройки топологии сети | 40 |
| 5 | Проектирование | 120 |
| 6 | Отладка | 90 |
| 7 | Составление и согласование проекта пакета служебных инструкций | 30 |
|  | Итого | 374 |

Следовательно, всего на разработку затрачено 374 часа или 49 рабочих дней при восьмичасовом рабочем дне.

3. Затраты на оплату труда работников, непосредственно занятых созданием научно-технической продукции

Разработка выполнялась инженером-программистом в течение 46 рабочих дней при восьмичасовом рабочем дне. Месячный фонд времени работы инженера-программиста 160 часов, среднемесячная заработная плата 9000 руб.

Основная заработная плата разработчика составила

руб.

Дополнительная заработная плата составляет 20%

*Здоп* = 0,2\**Зосн* = 0,2\*16247,37 = 2929,47 руб.

Затраты на оплату труда с учетом поясного коэффициента (25%)

*ЗТР* = 1,25\*(*Зосн* +*Здоп)* = 1,25\*(16247,37+2929,47) = 21221,05 руб.

4.4 Отчисления на социальные нужды

Единый социальный налог.

а) отчисления в Фонд Социального страхования(20% от затрат на оплату труда) 0,2\*21221,05 = 4244,21 руб.;

б) отчисления в Пенсионный фонд (2,9% от затрат на оплату труда) 0,029\*21221,05 = 615,4 руб.;

в) отчисления в Федеральный Фонд обязательного медицинского страхования (1,1% от затрат на оплату труда)

0,011\*21221,05 =233,44 руб.;

Итого единый социальный налог 5517,47 руб.

5. Прочие прямые расходы.

Стоимость проезда составила 740 руб.

6. Накладные расходы составляют 80% от затрат на оплату труда

0,80\*21221,05 = 16976,84 руб.

Форма 1-пн

Кафедра АСУ

Калькуляция составлена

"20" февраля\_\_ 200 7 г.

КАЛЬКУЛЯЦИЯ

плановой себестоимости

Автоматизированного рабочего места менеджера логистического отдела

Основание для проведения работ (договор, заказ) \_\_\_заказ\_\_\_\_\_\_\_

Заказчик: ЗАО Аптека Холдинг »

Срок выполнения работы: начало 1 марта 2007 г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

окончание 31 мая 2007г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| № | Наименование статей затрат | Сумма |
| 1 | Материалы | 1625,00 |
| 2 | Спецоборудование для научных (экспериментальных) работ | 0,00 |
| 3 | Затраты на оплату труда работников, непосредственно занятых созданием научно-технической продукции | 23721,00 |
| 4 | Отчисления на социальные нужды | 6152,00 |
| 5 | Прочие прямые расходы | 840,00 |
| 6 | Накладные расходы | 16976,00 |
| 7 | Итого: | 47259,00 |
| 8 | Затраты по работам, выполняемым сторонними организациями и предприятиями | 0,00 |
| 9 | Всего себестоимость | 47259,00 |

Воспользуемся методом установления цены на основе ориентации на уровень конкуренции. Программное обеспечение, подобное тому, что было получено в результате данной разработки, реализуется по цене около 50000 руб.

Таким образом, назначив оптовую цену в 30000 руб. и реализовав 3 экземпляров программного обеспечения, мы получим следующую валовую прибыль

30000 \* 3 – 47259,00 = 43852,0 руб.

Налог на прибыль (24%) составит

0,24 \* 43852,2 = 10524,53 руб.

Ожидаемая рентабельность проекта

.

Отпускная цена одного экземпляра программы составит (с учетом НДС 18%)

30000 \* 1,18 = 35200 руб.

Очевидно, что при такой цене и с учетом дополнительного сервиса, реализованного нами, данное программное обеспечение будет конкурентоспособно и принесет прибыль.

5. Раздел «Охрана труда»

5.1 Требования безопасности к хранению медикаментов на аптечных складах

Условия хранения лекарственных продуктов должны обеспечивать сохранность свойств лекарственных продуктов на протяжении всего срока их годности, предотвращения их загрязнения, перекрестного загрязнения и пересортицы.

Хранение лекарственных продуктов должно осуществляться в соответствии с условиями, установленными держателем регистрации и утвержденными при регистрации лекарственного продукта. За исключением случаев, допустимы краткосрочные отклонения от указанных условий, например, в процессе транспортировки.

Дополнительной гарантией соблюдения правил хранения лекарственных продуктов служит соблюдение международного стандарта по надлежащей практике хранения.

В Российской Федерации требования международного стандарта закрепляются разработанным в соответствии с ним национальным стандартом по надлежащей практике хранения, который должен периодически пересматриваться с учётом научно-технического прогресса.

Лекарственные продукты, чьи свойства могут существенным образом изменяться (быть утраченными) в результате даже кратковременных температурных отклонений от установленных условий, должны храниться и транспортироваться с использованием термоиндикаторов, фиксирующих критические отклонения от необходимых условий.

Помещения для хранения лекарственных препаратов должны быть оснащены оборудованием, позволяющим обеспечить их надлежащее хранение с учетом физико-химических, фармакологических и токсикологических свойств в течение установленного срока годности.

## Требования к помещениям и оборудованию оптовой торговли лекарственными средствами:

- Складские помещения предприятий оптовой торговли лекарственными средствами должны быть изолированными, специально оборудованными помещениями, позволяющими обеспечить хранение и надлежащую сохранность лекарственных средств с учетом их физико-химических, фармакологических и токсикологических свойств, а также требований стандартов качества лекарственных средств и Государственной фармакопеи.

- В помещении предприятия оптовой торговли лекарственными средствами должны быть предусмотрены складские и административно-бытовые помещения, объединенные в одном строении или расположенные раздельно (далее - склад).

- Склад может размещаться в отдельно стоящем нежилом здании, здании производственного назначения или в нежилых помещениях жилых домов.

При этом склад должен быть изолирован от других помещений, иметь отдельный вход, подъездную площадку, рампу для разгрузки товара.

При размещении склада в здании медицинского или аптечного назначения административно-бытовые помещения могут быть общими.

В случае размещения склада в нежилых помещениях жилых домов загрузка и выгрузка медицинской продукции не должна производиться под окнами квартир.

При размещении склада должно быть обеспечено выполнение стандартов.

- Складские помещения должны иметь системы электроснабжения, отопления, систему водоснабжения, канализации, приточно-вытяжную вентиляцию.

- Отделка помещений (внутренние поверхности стен, потолков) должна быть гладкой, допускать возможность проведения влажной уборки. Полы складских помещений должны иметь не образующее пыль покрытие, устойчивое к воздействию средств механизации и влажной уборки с использованием дезинфицирующих средств, при этом не допускается использование деревянных неокрашенных поверхностей. Материалы отделки помещений должны соответствовать установленным требованиям.

- Помещения для хранения наркотических лекарственных средств, психотропных веществ, ядовитых и сильнодействующих веществ должны соответствовать установленным требованиям.

- Возможность доступа посторонних лиц в производственные помещения (часть складского помещения, предназначенная для приема, сортировки, хранения, комплектации, отпуска и отгрузки лекарственных средств) должна быть исключена.

- Помещения склада должны быть функционально взаимосвязаны по выполняемым функциям: прием, хранение, комплектация заказов и отпуск товара.

- Площадь складских помещений основного производственного назначения должна соответствовать объему хранимого товара на единицу складской площади, но не менее 150 кв. м, включая:

* зону приемки продукции;
* зону для основного хранения лекарственных средств;
* помещение для лекарственных средств, требующих особых условий хранения;
* экспедиционную.

- Общая площадь административно-бытовых помещений зависит от численности персонала и рассчитывается согласно действующим нормам и правилам, но не менее 34 кв. м, включая вспомогательные складские помещения, предназначенные для размещения аппарата управления, бытовые помещения, например, служебные помещения аппарата управления складом, пункты питания, здравпункт, санитарно-бытовые помещения, вестибюли, лестничные клетки, тамбуры.

- На складе должно быть выделено специальное изолированное место для хранения моющих и дезинфицирующих средств, инвентаря и материалов, применяемых при уборке помещений и обработке оборудования, и гардеробная.

- Предприятие оптовой торговли должно быть оснащено оборудованием и инвентарем в соответствии с выполняемыми функциями:

* стеллажами, поддонами, подтоварниками для хранения медикаментов;
* холодильными камерами для хранения термолабильных лекарственных средств;
* средствами механизации для погрузочно-разгрузочных работ;
* приборами для регистрации параметров воздуха (термометрами, гигрометрами или психрометрами);
* запирающимися металлическими шкафами и сейфами для хранения отдельных групп лекарственных препаратов, учетной документации и справочной литературы;
* шкафами для хранения верхней и специальной одежды, обуви в гардеробной;
* дезинфекционными средствами и хозяйственным инвентарем для обеспечения санитарного режима.

- Все складские помещения, в которых хранятся лекарственные средства, должны иметь термометры, гигрометры или психрометры, которые размещают на внутренней стене помещения, вдали от нагревательных приборов на высоте 1,5 - 1,7 м от пола и на расстоянии не менее 3 м от дверей. Показатели этих приборов должны ежедневно регистрироваться в специальном журнале (карте) ответственным лицом. Контролирующие приборы должны быть сертифицированы и калиброваны в установленном порядке.

- Стеллажи для хранения лекарственных средств и изделий медицинского назначения должны быть установлены следующим образом:

* расстояние до наружных стен не менее 0,6 - 0,7 м;
* расстояние до потолка не менее 0,5 м;
* расстояние от пола не менее 0,25 м;
* проходы между стеллажами не менее 0,75 м;
* на всех стеллажах, шкафах, полках прикрепляется стеллажная карта с указанием наименования лекарственного препарата, серии, срока годности, количества единиц хранения.

Организация хранения лекарственных средств:

- Все товары на складе должны размещаться на стеллажах или на подтоварниках (поддонах) высотой не ниже 14,5 см. Не допускается размещение товара на полу без поддона. Каждое наименование и каждая серия лекарственных средств должны храниться на отдельных поддонах. Поддоны могут располагаться на полу в один ряд или на стеллажах в несколько ярусов, в зависимости от высоты стеллажа. Не допускается размещение поддонов с лекарственными средствами друг на друга без стеллажей.

- Не допускается загрузка объема помещения для складирования более чем на 1/3. При ручном способе разгрузочно-погрузочных работ высота укладки товара не должна превышать 1,5 метров. При использовании механизированных средств товар хранится в несколько ярусов, высота укладки товара на полке стеллажа не должна превышать 1,5 метров. Высота размещения товаров на стеллажах не должна превышать возможности механизированных погрузочно-разгрузочных средств (подъемники, автокары, тали и др.).

- Предприятиями оптовой торговли лекарственными средствами должны быть предприняты соответствующие меры в целях избежания повреждений (разливания, рассыпания, боя), поражения микроорганизмами и перекрестной контаминации лекарственных средств.

- Хранение медикаментов может быть систематизировано:

* по фармакологическим группам;
* по способу применения.

Для учета размещения лекарственных средств могут использоваться компьютерные технологии, согласованные в установленном порядке.

- На предприятии оптовой торговли лекарственными средствами должен вестись учет лекарственных средств по срокам годности, в том числе и на бумажном носителе.

- Лекарственные средства следует хранить раздельно с учетом их физико-химических свойств, при условиях, указанных предприятием - производителем лекарственных средств, соблюдая режимы температуры, влажности и освещенности.

- Лекарственные средства, требующие защиты от света, необходимо хранить в темном помещении или шкафах с плотно пригнанными дверцами, окрашенными внутри черной краской или в плотно сбитых ящиках с плотно пригнанной крышкой.

- Требования к хранению лекарственных средств, требующих защиты от влаги:

* лекарственные средства, требующие защиты от атмосферных паров воды, хранятся в прохладном месте, в герметично укупоренной таре из материалов, непроницаемых для паров воды (стекло, металл, алюминиевая фольга, толстая пластмасса);
* гипс жженый хранится в хорошо закрытой таре (например, плотно сбитые деревянные ящики, бочонки, выложенные изнутри полиэтиленовой пленкой);
* горчица в порошке хранится в герметично закрытых жестяных банках, лакированных изнутри;
* горчичники хранятся в пачках, упакованных в пергаментную бумагу или полиэтиленовую пленку, помещенных в плотно укупоренную тару.

- Требования к хранению летучих лекарственных средств, требующих защиты от улетучивания:

* лекарственные средства, требующие защиты от улетучивания, должны храниться в прохладном месте, в герметично укупоренной таре, из непроницаемого для улетучивающихся веществ материала (стекло, металл, алюминиевая фольга и др.);
* кристаллогидраты хранятся в прохладном месте при относительной влажности воздуха 50 - 65% в герметично укупоренной стеклянной, металлической или толстостенной пластмассовой таре.

- Требования к хранению термолабильных лекарственных средств:

* при хранении лекарственных средств, требующих защиты от повышенной или пониженной температуры, необходимо соблюдать температурный режим, указанный на этикетке или в инструкции по применению;
* органопрепараты следует хранить в сухом, прохладном, защищенном от света месте при температуре 0 - +15 град. С (если нет других указаний);
* медицинские жирные масла хранят при температуре от +4 до +12 град.

- Лекарственные средства, требующие защиты от воздействия газов, хранятся в герметически закупоренной таре, изготовленной из материалов, непроницаемых для газов, по возможности заполненной доверху.

- Требования к хранению наркотических средств и психотропных веществ, а также сильнодействующих и ядовитых веществ:

* хранение наркотических средств и психотропных веществ, а также сильнодействующих и ядовитых веществ производится в помещениях, оснащенных многорубежной охранной сигнализацией с подключением каждого рубежа на отдельный пульт;
* входная дверь должна быть толщиной не менее 40 мм, обита железом с обеих сторон с загибом листа на торец двери внахлест или на внутреннюю поверхность двери, или металлическая дверь;
* обрамление дверного проема должно быть выполнено из стального профиля, внутри - решетчатая металлическая дверь;
* внутренняя решетка на оконных проемах (или решетка между рамами) должна быть выполнена из стального прута диаметром не менее 16 мм.
* необходимо наличие охранной сигнализации, включенной во второй рубеж защиты, на внутренних дверях, стенах, потолке;
* наркотические средства, психотропные, сильнодействующие и ядовитые вещества хранятся в запирающихся сейфах или железных шкафах (в технически укрепленных помещениях);
* доступ в помещения хранения наркотических средств, психотропных, ядовитых и сильнодействующих веществ разрешен только лицам, непосредственно работающим с ними, указанным в приказе руководителя предприятия.

- Требования к хранению пахучих и красящих лекарственных средств:

* пахучие лекарственные средства хранятся изолированно в герметически закрытой таре, непроницаемой для запаха, раздельно по наименованиям;
* красящие лекарственные средства хранятся в специальном шкафу в плотно укупоренной таре, раздельно по наименованиям.

- Требования к хранению лекарственного растительного сырья:

* лекарственное растительное сырье (предварительно высушенное) должно храниться в сухом, хорошо вентилируемом помещении, в закрытой таре (стеклянной, металлической, деревянной);
* резаное сырье хранят в тканевых мешках, порошок - в двойных мешках (внутренний - бумажный, наружный - тканевый);
* лекарственное растительное сырье, содержащее эфирные масла, необходимо хранить изолированно в хорошо укупоренной таре;
* ядовитое и сильнодействующее растительное сырье хранится в отдельном помещении или в отдельном закрытом на замок шкафу;

- Требования к хранению дезинфицирующих средств:

* дезинфицирующие средства хранят в герметично укупоренной таре в защищенном от света прохладном месте;
* хранение производится в отдельном помещении, изолированно от лекарственных препаратов, а также вдали от помещений хранения пластмассовых, резиновых и металлических изделий.

- Требования к хранению взрывоопасных и легковоспламеняющихся веществ:

* взрывоопасные и легковоспламеняющиеся вещества должны храниться в отдельном помещении (сухом, защищенном от прямого солнечного света, осадков и грунтовых вод, расположенном на 1 этаже, имеющем железобетонное перекрытие и металлическую дверь);
* несгораемые стеллажи должны быть установлены на расстоянии не менее 0,25 м от пола и стен, ширина стеллажей не более 1 м, и должны иметь отбортовки не менее 0,25 м, проходы между стеллажами не менее 1,35 м;
* несгораемые шкафы должны иметь дверцы шириной не менее 0,7 м и высоту не менее 1,2 м;
* электропроводка, осветительная арматура и электрооборудование должны быть выполнены во взрывобезопасном исполнении с выносом выключателя в коридор;
* крупные емкости должны храниться только в один ряд по высоте;
* расстояние от стеллажей до нагревательных приборов - не менее 1 м;
* - степень заполнения тары при хранении легковоспламеняющихся и горючих жидкостей - не более 90% (спирты - 75%);
* - газовые баллоны хранятся в вертикальном положении, кислородные подушки хранятся в подвешенном виде рядом с баллонами.

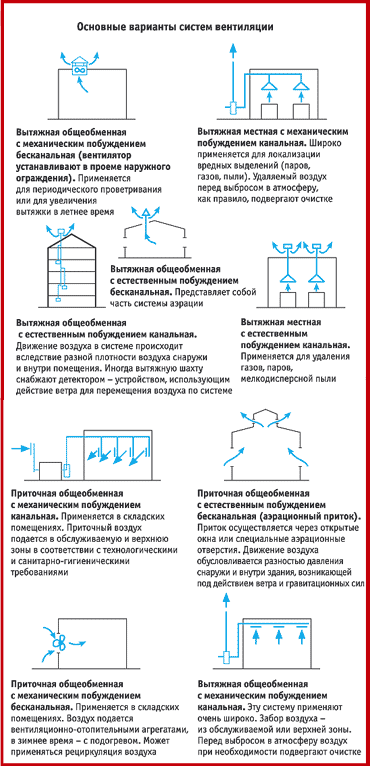
5.2 Расчёт мощности вентилятора

Сохранность материальных ценностей в складских помещениях в значительной мере обеспечивается за счет поддержания определенных параметров микроклимата – температуры, относительной влажности, подвижности воздуха.

Вытяжные системы предназначены для удаления из помещений загрязненного воздуха. Приточные системы служат для подачи в вентилируемые помещения чистого воздуха взамен удаленного. Подаваемый воздух в необходимых случаях подвергают специальной обработке – очистке, нагреву, увлажнению. В общем случае в помещении предусматривают и приточные системы, и вытяжные, причем их производительность должна быть примерно одинаковой.

По конструктивному исполнению системы вентиляции разделяют на канальные системы, которые имеют разветвленную сеть воздуховодов для перемещения воздуха, и беканальные, в которых каналы (воздуховоды) отсутствуют, например, при аэрации, установке вентиляторов в стене, перекрытиях.

Воздухообменом называется замена загрязненного воздуха помещений чистым, создающая в помещениях воздушную среду, которая соответствует санитарно-гигиеническим и технологическим требованиям. Для определения воздухообмена (объема вентиляционного воздуха в единицу времени) необходимо знать виды и количество вредных выделений в помещении – избытка тепла, влаги, газов, паров, пыли.



*Рисунок 5.1 – Основные варианты систем вентиляции*

Учитывая специфику вентиляции аптечного склада нужно проектировать приточно-вытяжную вентиляцию с механическим побуждением.

Приточно-вытяжная вентиляция - вентиляция, состоит:

- из приточной системы, подающей в помещение чистый воздух, а также возмещающей воздух, расходуемый на технологические нужды;

- из вытяжной системы, удаляющей из помещения загрязненный воздух.

Приточно-вытяжная вентиляция и ее составные части

• Наружная воздухозаборная решетка

• Обратный клапан

• Вентилятор вытяжного действия

• Автоматика

• Виброглушитель

• Воздуховодные каналы

• Регуляторы на воздуховодах

• Вытяжные диффузоры или решетки

Приточно-вытяжная вентиляция применяется в условиях, когда невозможно организовать естественный приток воздуха и существует необходимости очистки наружного воздуха. Рециркуляция воздуха в лечебных учреждениях не допускается!

Наружный воздух, подаваемый системами приточной вентиляции, надлежит очищать в фильтрах. Как правило, приточный воздух обрабатывают в центральных приточных камерах или кондиционерах. Вентиляционные приточные и вытяжные камеры размещают таким образом, чтобы была исключена передача шума в помещения. Воздуховоды систем приточной вентиляции и кондиционирования воздуха после бактериологических фильтров рекомендуется выполнять из нержавеющей стали.

Расчет системы вентиляции для помещения, как правило, начинается с подбора оборудования, подходящего по таким параметрам, как производительность по прокачиваемому объему воздуха и измеряемому в кубометрах в час.

При расчете и монтаже систем вентиляции должен быть рассчитан ряд параметров, являющихся ключевыми для нормального функционирования вентиляционной системы. Основными характеристиками системы вентиляции являются:

- Производительность по воздуху (м3/ч)

- Рабочее давление (Па) и скорость воздушного потока (м/с)

- Уровень шума (дБ)

- Мощность калорифера (кВт)

Расчет системы вентиляции начинается с подбора оборудования, подходящего по таким параметрам, как производительность по прокачиваемому объему воздуха и измеряемому в кубометрах в час. Важнейшим показателем в системе является кратность воздухообмена, которая показывает, сколько раз происходит полная замена воздуха в помещении в течение часа. Кратность воздухообмена определяется СНиП (Строительными Нормами и Правилами) и зависит от назначения помещения, количества оборудования, выделяющего тепло, а также от того, сколько людей находится в помещении. Как правило, для жилых помещений необходимая кратность воздухообмена составляет единицу, в то время как для рабочих помещений (офис и др.) это значение должно составлять 2-3. В сумме все значения по кратности воздухообмена для всех помещений составляют производительность по воздуху. Как правило, обычные значения производительности составляют: для офисов: 1000-10000 кубометров/ч, для коттеджей: 1000-2000 кубометров/ч, для квартир: 100-800 кубометров/ч.

Следующий этап в расчете вентиляции - проектирование воздухораспределительной сети, состоящей из таких компонентов, как воздуховоды, распределители воздуха, а также фасонные изделия (переходники, повороты, разветвители) Сначала разрабатывается схема воздуховодов, по которой производится расчет уровня шума, рабочего давления и скорости потока воздуха. Рабочее давление напрямую зависит от того, какова мощность используемого вентилятора и расчитывается с учетом диаметров воздуховодов, количества переходов с одного диаметра на другой, и количества поворотов.

Рассчитаем необходимую мощность вентилятора для аптечного складского помещения. Размеры аптечного склада (м): 120\*40\*5 = 2400 кубометров.

Q = V n \* K,

где V n - объем помещения, м3

K - кратность воздухообмена (равен 2, для аптечных складов)

Q = 2400 \* 2 = 4800

Мощность электродвигателя, необходимая для привода вентилятора:

W в = (1,2... 1,5) \*Q \* H в, кВт

V в \* ђ в \* ђ п

где 1,2... 1,5 - коэффициент, учитывающий

неучтенные потери напора воздушного потока.

V в - скорость вращение вентилятора, об/ч

H в - напор воздушного потока, кг / м2

ђ в - КПД вентилятора

ђ п - КПД передачи

W в = 1,2 \* \_\_\_\_4800 \* 32\_\_\_\_ = 39 кВт

3600\* 0,8 \* 0,95

Для склада необходимо приобрести 2 вентилятора центробежного типа общего назначения, с мощность по 20 кВт каждый. К примеру, вентиляторы марки Fuji RS со следующими характеристиками.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| МОДЕЛЬ | Мощность электродвигателя, кВт | Потреб-ляемая мощность, кВт | Уровень шума при работе на охл./нагр., dB | Расход воздуха, м3/час |
| RSA-18LCC (R410A) | 20 | 1,7 | 36 | 800 |

Заключение

В соответствии с заданием на дипломный проект, проведено предпроектное обследование рабочего места логистика склада. В результате анализа были выявлены основные функции и задачи, решаемые на данном рабочем месте.

Построена информационная модель существующей системы управления, описаны и квалифицированы формы входных и выходных документов.

Определены основные требования, предъявляемые пользователем к системе. Разработан технологический процесс функционирования системы в автоматизированном режиме и построена информационная модель системы. На основе проведённого анализа, спроектированы структуры файлов баз данных. Разработаны программные модули системы.

Результатом исследований существующей системы, а также дальнейшее проектирование интерфейса, стала программное обеспечение, описание которого дано в руководстве пользователя.

Отличительной частью системы является интерфейс, основанный на разработке классификации входных и выходных документов, что делает систему доступной для неквалифицированного пользователя.

Разработанную систему предлагается использовать на складах ЗАО «Аптека - Холдинг» и в его филиалах.

Список использованных источников

1. Гаджинский А.М. Практикум по логистике / А.М. Гаджинский. - М.: ИКЦ Маркетинг, 2001. – 84 с.
2. Костоглодов Д.Д., Харисова Л.М., Распределительная логистика / Д.Д. Костоглодов, Л.М. Харисова. - М.: Экспертное бюро, 1997. – 146 с.
3. Дыбская В.В. Управление складом в логистической системе / В.В. Дыбская. - М.: 2000. – 172 с.
4. Николайчук В.Е. Логистика: учебное пособие. - СПб.: «Питер», 2002. – 46с.

Аникин Б.А. Логистика / Б.А. Аникин. - М.: ИНФРА-М, 1998. – 180с.

Берков М.В. Информационные технологии в логистике / М.В. Берков. - М.: Финансы и статистика, 1999. – 76с.

1. Бойко В.В., Савинков В.М. Проектирование баз данных информационных систем / В.В. Бойко, В.М. Савинков. - М.: Финансы и статистика, 1999. - 68с.
2. Ахаян Р. Д. Эффективная работа с СУБД / Р.Д. Ахаян. - СПб.: Питер, 1997. – 284с.

9. Семёнова И.Н. Оптимальный размер заказа, формула Вильсона [Электронный ресурс] / И.Н. Семёнова; ГУ ВШЭ. – Режим доступа:

http://www.logistpro.ru/div8/upr\_zapas/wilson.html;

1. Верников Г.И. Стандарты планирования, описание стандарта MRPII [Электронный ресурс] / Г.И Верников; ГУУ. - Режим доступа: http://www.cfin.ru/vernikov/mrp/mrp2.html
2. Кондрашов С.В. Логистика в системе управления товарно-материальными потоками промышленного предприятия [Электронный ресурс] / С.В. Кондрашов; ГУУ. - Режим доступа: http://www.cfin.ru/bandurin/article/sbrn08/index.shtml

12. Залманова М.Е. Закупочно – распределительная логистика: учебное пособие. – Саратов: СПИ, 1992

13. Дыбская В.В., Зайцев Е.И., Сергеев В.И. Корпоративная логистика. 300 ответов на вопросы профессионалов / В.В. Дыбская, Е.И. Зайцев, В.И. Сергеев. - М.: Инфра - М, 2004. – 180с.

14. Гаджинский А.М. Логистика на производстве / А.М Гаджинский. - СПб.: «Питер», 1999. – 64с.

15. Кокурин Д.И. Основы логистики в промышленности / Д.И. Кокурин - М.: Финансы и статистика, 2000. – 56с.

1. Н.Г. Низовкина, Л.И. Гнездилова. Методические указания к организационно-экономической части дипломных проектов. №1879.– Новосибирск: НГТУ, 1999. – 48 с.
2. Захарова Е.Я. Экономика информатики: Учебное пособие. – Новосибирск: НГТУ, 1999. – 126с.
3. Липаев В.В., Потапов А.И. Оценка затрат на разработку программных средств / В.В. Липаев, А.И. Потапов – М.: Финансы и статистика, 1998. – 182 с.
4. Полярин Ю.В. Организация системы вентиляции на складе [Электронный ресурс] / Ю.В. Полярин; ГУ ВШЭ, - Режим доступа:

Приложение А

Формы входных и выходных документов

Приходная ведомость

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Номер по по рядку | Товар | | Единица измерения | | Вид упаковки | Количество | Цена |
|  | наименование | код | наиме- нование | код по ОКЕИ |  | штук | руб. коп. |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  | Итого |  |  |
|  |  |  |  |  | Всего по ведомости |  |  |

Дата поставки:\_\_\_\_\_\_\_\_\_\_\_\_\_ Поставщик:\_\_\_\_\_\_ \_\_\_\_\_\_\_

Ведомость об остатках

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Но мер по по- рядку | Товар | | | Единица измерения | | Вид упаковки | Количество  оставшихся товаров |
|  | наименование | код | код группы | наиме- нование | код по ОКЕИ |  | штук |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | Итого |  |
|  |  |  |  |  |  | Всего по ведомости |  |

Акт о недостаче

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N по | Товар |  | Единица измерения | | Цена. | Номер | | Фактическая недостача | Сумма |
| поряд-  ку | наиме  нование | код | наиме  нование | код ОКЕИ | руб | инвен-  тарный | паспорта | количество | руб. |
| 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Итого: |  |  |
|  |  |  |  |  |  |  | Итого по описи: |  |  |

Бланк заказа

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер по порядку | Товар | | | Поставщик | Единица измерения | | Вид упаковки | Количество | Цена |
|  | наименование | код | код группы | код | наиме- нование | код по ОКЕИ |  | штук | руб. коп. |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Итого |  |  |

Объем продаж за период

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер по порядку | Дата продажи | Товар | | | Единица измерения | | Вид  упаковки | Количество | Цена |
|  |  | наиме  нование | код | код  группы | наиме- нование | код по ОКЕИ |  | штук | руб. коп. |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Итого |  |  |
|  |  |  |  |  |  |  | Всего за период |  |  |

Списания со склада за период

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер по порядку | Дата списания | Товар | | | Единица измерения | | Вид упаковки | Количество | Цена |
|  |  | наиме  нование | код | код  группы | наиме- нование | код по ОКЕИ |  | штук | руб. коп. |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Итого |  |  |
|  |  |  |  |  |  |  | Всего за период |  |  |

Приходная накладная

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Но- мер по по- рядку | Дата | Товар | | | Единица измерения | | Вид упаковки | Количе- | Цена, |
| поставки | ство |  |
|  |  | наименование | | код | наиме- нование | код по ОКЕИ |  | штук | руб. коп |
|  |
| 1 | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 |
|  |  |  | |  |  |  |  |  |  |
|  |  |  | |  |  |  |  |  |  |
|  |  |  | |  |  |  |  |  |  |
|  |  |  | |  |  |  |  |  |  |
|  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Итого |  |  |
|  |  |  |  |  |  |  | Всего по накладной |  |  |

Расходная накладная

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Но- мер по по- рядку | Товар | | Единица измерения | | Вид упаковки | Количество | Цена |
|  | наименование | код | наиме- нование | код по ОКЕИ |  | штук | руб. коп. |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  | Итого |  |  |
|  |  |  |  |  | Всего по накладной |  |  |

Акт на списание товара

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер по порядку | Товар | |  | Единица измерения | | Причина списания | Вид упаковки | Количество | Цена,  руб. коп. |
|  | наименование | | код | наиме- нование | код по ОКЕИ |  |  | штук |  |
| 1 | 2 | | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Итого |  |  |

Инвентаризационная ведомость

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N по | Товар |  | Единица измерения | | Цена. | Номер | | Фактическое наличие | Сумма |
| поряд-  ку | наиме  нование | код | наиме  нование | код ОКЕИ | руб | инвен-  тарный | пас-  порта | количество | руб. |
| 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Итого: |  |  |
|  |  |  |  |  |  |  | Итого по описи: |  |  |

Приложение Б

Руководство пользователя

Данная система имеет два типа зарегистрированных пользователей: логистик склада и администратор системы. Поскольку данная система поддерживает основные функции логистика склада, то в данном руководстве рассматривается его работа. Логистик склада, в отличие от администратора системы, имеет ограниченные права на изменение информации в справочниках базы данных системы.

Б.1 Общие сведения

Для работы системы необходимо иметь операционную систему Windows 98/NT/2000/XP, MSOffice2000. Для удобства работы с программой расширение экрана должно быть 800х600, оперативная память не менее 64МГб, свободного места на диске не менее 50МГб, клавиатура, манипулятор типа «мышь», принтер.

Б.2 Установка системы

Скопируйте в рабочую директорию файл sklad.exe. Запустите его. Система не требует предварительной настройки.

Б.3 Работа с системой

Б.3.1 Описание предварительного диалога

Предварительный диалог пользователя с системой состоит в следующем. При запуске программы активируется диалоговое окно, представленное на рисунке 5.6, позволяющее идентифицировать пользователя. Если такой пользователь существует, то предоставляется доступ к работе с программой, иначе выводится сообщение о том, что доступ запрещен и окно для ввода логина и пароля активируется снова.

После регистрации пользователя следует запуск главное окно системы.

Б.3.2 Описание меню системы

Диалог системы и пользователя организован на основе меню. Меню системы представлено на рисунке 5.7. Главное меню программы содержит пять основных разделов:

* «Документы»;
* «Правка»
* «Справочники»;
* «Отчёты»;
* «Справка».

Б.3.3 Описание работы с пунктом меню «Файл»

Рассмотрим пункт меню «Документы». В этом разделе предоставляется доступ к журналам базы данных. Пункт меню «Файл» разделен на следующие подразделы:

* «Поступления»;
* «Продажи»;
* «Списания»;
* «Инвентаризация».

Эти подразделы предназначены для ввода документов «Приходная накладная», «Расходная накладная», «Акт на списание» и «Инвентаризационная ведомость». Рассмотрим работу их на примере подпункта меню «Продажи». Окно работы программы разделено на 2 части (см. рисунок 5.8). Первая часть предназначена для добавления новых записей на основе документа расходная накладная. Она имеет три поля для выбора наименования товара, заказчика, сотрудника, ответственного за отгрузку товара из справочников «Наименования товаров», «Сотрудники», «Клиенты». Так же эта часть окна предоставляет выбор даты продажи товара, ввод номера документа и количества проданного товара. На данной форме имеется две кнопки управления: «Очистить форму» и «Добавить запись». При нажатии кнопки «Очистить форму» в форме добавления записи очищаются все выбранные записи. При нажатии кнопки «Добавить строку» в базу данных добавляется новая строка и в нее записываются выбранные значения из бланка добавления записи. В другой части окна мы имеем возможность просмотра внесённых изменений в таблицу базы данных «Продажи».

Б.3.3 Описание работы с пунктом меню «Справочники»

Рассмотрим пункт меню «Справочники». В этом пункте меню предоставляется доступ к справочникам базы данных. Пункт меню «Справочники» разделен на следующие подпункты:

* «Товар»;
* «Наименования товаров»;
* «Группы товаров»;
* «Сотрудники»;
* «Должность»;
* «Клиенты»;
* «Поставщики»;
* «Банки»;
* «Причины списания».

Подпункты пункта меню «Справочники» предназначены для просмотра и редактирования справочников базы данных. Для этого предусмотрено два поля (см. рисунок 5.9): первое – добавление данных в справочник, второе – просмотр справочника системы.

Б.3.3 Описание работы с пунктом меню «Отчеты»

В пункте меню «Отчеты» представлены все отчеты системы. К ним относят:

* «Бланк заказа»;
* «Объем продаж»;
* «Ведомость остатков»;
* «Списания».

При выборе пункта меню «Ведомость остатков» появляется окошко (см. рисунок 5.10), предлагающее выбрать группу товаров для вывода таблицы ведомости остатков. При нажатии кнопки «Отмена» управление программой переходит в главное меню. При нажатии кнопки «ОК» программа начинает формировать отчёт «Ведомость остатков» по введенным параметрам. Результаты формирования отчёта представлены на рисунке 5.11. На этой экранной форме имеется кнопка управления «Изменить параметры расчета». При нажатии этой кнопки программа открывает диалог задания параметров для формирования отчёта «Ведомость остатков».

В столбце optim\_kolvo таблицы «Ведомость остатков» указывается оптимальное количество товаров, рассчитанное исходя из затрат на хранение данного товара на складе. Значение 1 в столбце garantzapas показывает, что количество товара на данный момент меньше либо равно величине гарантийного запаса. Это означает, что пользователь увидев значение 1 в столбце должен пополнить запасы данного товара.

Рассмотрим подпункт меню «Бланк заказа». При выборе этого подпункта меню организуется диалог с пользователем для задания параметров Бланка заказа (см. рисунок 5.12). Пользователю предлагается ввести номер заказа и количество товаров в заказе, а также выбрать дату заказа, наименование товара, поставщика и сотрудника оформившего заказ для вывода таблицы заказы.

В графе количество есть 2 параметра: рекомендуемое количество и количество заказа. Рекомендуемое количество было рассчитано программой исходя из данных об остатках товара, стоимости хранения, спроса на товар, среднего времени поставки, среднего времени задержки поставки товара. Количество заказа показывает сколько пользователь решил заказать товаров, это количество может быть равным, большим или меньшим чем рекомендуемое, в зависимости от поставленных пользователем целей.

Результаты формирования таблицы «Заказы» представлены на рисунке 5.12. На этой экранной форме имеется кнопка управления «Изменить параметры расчета». При нажатии этой кнопки программа открывает диалог задания параметров для формирования «Бланка заказа».

Приложение В

Листинг программы

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

//---------------------------------------------------------------------------

USEFORM("Unit1.cpp", Form1);

USEFORM("Unit2.cpp", Form2);

USEFORM("Unit3.cpp", Form3);

//---------------------------------------------------------------------------

WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)

{

try

{

Application->Initialize();

Application->CreateForm(\_\_classid(TForm1), &Form1);

Application->CreateForm(\_\_classid(TForm2), &Form2);

Application->CreateForm(\_\_classid(TForm3), &Form3);

Application->Run();

}

catch (Exception &exception)

{

Application->ShowException(&exception);

}

catch (...)

{

try

{

throw Exception("");

}

catch (Exception &exception)

{

Application->ShowException(&exception);

}

}

return 0;

}

//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

#include "Unit2.h"

#include "Unit3.h"

#include "date.h"

#include <systdate.h>

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N6Click(TObject \*Sender)

{

Form1->Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N8Click(TObject \*Sender)

{

this->GroupBox2->Visible=false;

this->GroupBox3->Visible=false;

this->GroupBox4->Visible=false;

this->GroupBox5->Visible=false;

this->GroupBox6->Visible=false;

this->GroupBox7->Visible=false;

this->GroupBox13->Visible=false;

this->GroupBox1->Visible=true;

UpdateComboBoxGoods();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N2Click(TObject \*Sender)

{

this->GroupBox1->Visible=false;

this->GroupBox3->Visible=false;

this->GroupBox4->Visible=false;

this->GroupBox5->Visible=false;

this->GroupBox6->Visible=false;

this->GroupBox7->Visible=false;

this->GroupBox13->Visible=false;

this->GroupBox2->Visible=true;

UpdateComboBoxSale();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N3Click(TObject \*Sender)

{

this->GroupBox1->Visible=false;

this->GroupBox2->Visible=false;

this->GroupBox4->Visible=false;

this->GroupBox6->Visible=false;

this->GroupBox5->Visible=false;

this->GroupBox7->Visible=false;

this->GroupBox13->Visible=false;

this->GroupBox3->Visible=true;

UpdateComboBoxSupplies();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N4Click(TObject \*Sender)

{

this->GroupBox2->Visible=false;

this->GroupBox3->Visible=false;

this->GroupBox1->Visible=false;

this->GroupBox5->Visible=false;

this->GroupBox6->Visible=false;

this->GroupBox7->Visible=false;

this->GroupBox13->Visible=false;

this->GroupBox4->Visible=true;

UpdateComboBoxWrite\_off();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N5Click(TObject \*Sender)

{

this->GroupBox2->Visible=false;

this->GroupBox3->Visible=false;

this->GroupBox1->Visible=false;

this->GroupBox4->Visible=false;

this->GroupBox6->Visible=false;

this->GroupBox7->Visible=false;

this->GroupBox13->Visible=false;

this->GroupBox5->Visible=true;

UpdateComboBoxInventory();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N29Click(TObject \*Sender)

{

this->GroupBox2->Visible=false;

this->GroupBox3->Visible=false;

this->GroupBox1->Visible=false;

this->GroupBox4->Visible=false;

this->GroupBox5->Visible=false;

this->GroupBox7->Visible=false;

this->GroupBox13->Visible=false;

// this->GroupBox6->Visible=true;

Form1->Enabled=false;

Form2->Show();

}

//---------------------------------------------------------------------------

void TForm1:: GZ()

{

TDataSet \*ds1 = DataSource18->DataSet;

TDataSet \*ds2 = DataSource20->DataSet;

TDataSet \*ds3 = DataSource19->DataSet;

int S= 0;

int tpost

int zpost

int length = ds1->RecordCount;

AnsiString \*str1 = new AnsiString[length];

int gr = (int)Form1->Table1->FieldValues["id\_group"];

int dn = (int)Form1->Table1->FieldValues["dn"];

int dk = (int)Form1->Table1->FieldValues["dk"];

for (ds1->First();!ds1->Eof; ds1->Next())

{

if(Form1->Table2->FieldValues["id\_group"]==gr)

{

S = 0;

for (ds2->First();!ds2->Eof; ds2->Next())

Form1->Button11->Caption = "Zapasi";

if(Form1->Table3->FieldValues["id\_name"]==Form1->Table2->FieldValues["id\_tov"])

if((int)Form1->Table3->FieldValues["date"]>=dn && (int)Form1->Table3->FieldValues["date"]<=dk && Form1->Table3->FieldValues["id\_packing"]==pac)

S=S+ Form1->Table3->FieldValues["amount"];

Form1->Table1->Append();

Form1->Table1->FieldValues["id\_name"] = Form1->Table2->FieldValues["id\_tov"];

Form1->Table1->FieldValues["Vsale"] = S;

Form1->Table1->Post();

}

}

Vprod = 0;

for (ds3->First(), ds3->Next();!ds3->Eof; ds3->Next())

S = S + Form1->Table1->FieldValues["Vsale"];

for (ds3->First(), ds3->Next();!ds3->Eof; ds3->Next())

{

Form1->Table1->Append();

Form1->Table1->FieldValues["spare"] = Form1->Table1->FieldValues FGZ = (tpost + zpost) \* P – P\* tpost

Form1->Table1->Post();

Form1->Query6->Active=false;

Form1->Query6->Active=true;

}

Form1->Query6->Active=false;

Form1->Query6->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::ORZClick(TObject \*Sender)

{

this->GroupBox2->Visible=false;

this->GroupBox3->Visible=false;

this->GroupBox1->Visible=false;

this->GroupBox4->Visible=false;

this->GroupBox5->Visible=false;

this->GroupBox6->Visible=false;

this->GroupBox13->Visible=false;

// this->GroupBox7->Visible=true;

Form1->Enabled=false;

Form3->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button11Click(TObject \*Sender)

{

Form1->GroupBox6->Visible=false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button12Click(TObject \*Sender)

{

Form1->Enabled=false;

Form2->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button13Click(TObject \*Sender)

{

Form1->GroupBox7->Visible=false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button14Click(TObject \*Sender)

{

Form1->Enabled=false;

Form3->Show();

}

//---------------------------------------------------------------------------

void TForm1::UpdateComboBoxGoods()

{

ComboBox1->Clear();

ComboBox2->Clear();

ComboBox3->Clear();

ComboBox4->Clear();

ComboBox5->Clear();

Query8->Active = false;

Query8->Active = true;

TDataSet \*ds = DataSource8->DataSet;

int length = ds->RecordCount;

AnsiString \*str = new AnsiString[length];

int i = 0, j = 0;

for (ds->First();!ds->Eof; ds->Next())

str[i++] = DataSource8->DataSet->FieldByName("Tov")->AsString;

for (j = 0; j < length; j++)

{

ComboBox1->Items->Add(str[j]);

}

Query9->Active = false;

Query9->Active = true;

TDataSet \*ds1 = DataSource9->DataSet;

length = ds1->RecordCount;

AnsiString \*str1 = new AnsiString[length];

i = 0;

for (ds1->First();!ds1->Eof; ds1->Next())

str1[i++] = DataSource9->DataSet->FieldByName("Group")->AsString;

for (j = 0; j < length; j++)

{

ComboBox2->Items->Add(str1[j]);

}

Query10->Active = false;

Query10->Active = true;

TDataSet \*ds2 = DataSource10->DataSet;

length = ds2->RecordCount;

AnsiString \*str2 = new AnsiString[length];

i = 0;

for (ds2->First();!ds2->Eof; ds2->Next())

str2[i++] = DataSource10->DataSet->FieldByName("Naimenovanie")->AsString;

for (j = 0; j < length; j++)

{

ComboBox3->Items->Add(str2[j]);

}

Query11->Active = false;

Query11->Active = true;

TDataSet \*ds3 = DataSource11->DataSet;

length = ds3->RecordCount;

AnsiString \*str3 = new AnsiString[length];

i = 0;

for (ds3->First();!ds3->Eof; ds3->Next())

str3[i++] = DataSource11->DataSet->FieldByName("Prodagi")->AsString;

for (j = 0; j < length; j++)

{

ComboBox4->Items->Add(str3[j]);

}

Query12->Active = false;

Query12->Active = true;

TDataSet \*ds4 = DataSource12->DataSet;

length = ds4->RecordCount;

AnsiString \*str4 = new AnsiString[length];

i = 0;

for (ds4->First();!ds4->Eof; ds4->Next())

str4[i++] = DataSource12->DataSet->FieldByName("Proizvoditel")->AsString;

for (j = 0; j < length; j++)

{

ComboBox5->Items->Add(str4[j]);

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

UpdateComboBoxGoods();

Edit1->Text = "";

Edit2->Text = "";

Edit3->Text = "";

Edit4->Text = "";

}

//---------------------------------------------------------------------------

void TForm1::UpdateComboBoxSale()

{

ComboBox6->Clear();

ComboBox7->Clear();

ComboBox10->Clear();

Query8->Active = false;

Query8->Active = true;

TDataSet \*ds = DataSource8->DataSet;

int length = ds->RecordCount;

AnsiString \*str = new AnsiString[length];

int i = 0, j = 0;

for (ds->First();!ds->Eof; ds->Next())

str[i++] = DataSource8->DataSet->FieldByName("Tov")->AsString;

for (j = 0; j < length; j++)

{

ComboBox6->Items->Add(str[j]);

}

Query9->Active = false;

Query9->Active = true;

TDataSet \*ds1 = DataSource9->DataSet;

length = ds1->RecordCount;

AnsiString \*str1 = new AnsiString[length];

i = 0;

for (ds1->First();!ds1->Eof; ds1->Next())

str1[i++] = DataSource9->DataSet->FieldByName("Group")->AsString;

for (j = 0; j < length; j++)

{

ComboBox7->Items->Add(str1[j]);

}

Query13->Active = false;

Query13->Active = true;

TDataSet \*ds2 = DataSource13->DataSet;

length = ds2->RecordCount;

AnsiString \*str2 = new AnsiString[length];

i = 0;

for (ds2->First();!ds2->Eof; ds2->Next())

str2[i++] = DataSource13->DataSet->FieldByName("Surname")->AsString;

for (j = 0; j < length; j++)

{

ComboBox10->Items->Add(str2[j]);

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button4Click(TObject \*Sender)

{

UpdateComboBoxSale();

Edit5->Text = "";

Edit9->Text = "";

}

//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

void TForm1::UpdateComboBoxSupplies()

{

ComboBox8->Clear();

ComboBox9->Clear();

ComboBox11->Clear();

Query8->Active = false;

Query8->Active = true;

TDataSet \*ds = DataSource8->DataSet;

int length = ds->RecordCount;

AnsiString \*str = new AnsiString[length];

int i = 0, j = 0;

for (ds->First();!ds->Eof; ds->Next())

str[i++] = DataSource8->DataSet->FieldByName("Name")->AsString;

for (j = 0; j < length; j++)

{

ComboBox8->Items->Add(str[j]);

}

// delete ds;

Query9->Active = false;

Query9->Active = true;

TDataSet \*ds1 = DataSource9->DataSet;

length = ds1->RecordCount;

AnsiString \*str1 = new AnsiString[length];

i = 0;

for (ds1->First();!ds1->Eof; ds1->Next())

str1[i++] = DataSource9->DataSet->FieldByName("Group")->AsString;

for (j = 0; j < length; j++)

{

ComboBox9->Items->Add(str1[j]);

}

Query13->Active = false;

Query13->Active = true;

TDataSet \*ds2 = DataSource13->DataSet;

length = ds2->RecordCount;

AnsiString \*str2 = new AnsiString[length];

i = 0;

for (ds2->First();!ds2->Eof; ds2->Next())

str2[i++] = DataSource13->DataSet->FieldByName("Name")->AsString;

for (j = 0; j < length; j++)

{

ComboBox11->Items->Add(str2[j]);

}

}

//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button6Click(TObject \*Sender)

{

UpdateComboBoxSupplies();

Edit6->Text = "";

Edit7->Text = "";

}

//---------------------------------------------------------------------------

void TForm1::UpdateComboBoxWrite\_off()

{

ComboBox12->Clear();

ComboBox13->Clear();

ComboBox14->Clear();

ComboBox15->Clear();

Query8->Active = false;

Query8->Active = true;

TDataSet \*ds = DataSource8->DataSet;

int length = ds->RecordCount;

AnsiString \*str = new AnsiString[length];

int i = 0, j = 0;

for (ds->First();!ds->Eof; ds->Next())

str[i++] = DataSource8->DataSet->FieldByName("Tov")->AsString;

for (j = 0; j < length; j++)

{

ComboBox12->Items->Add(str[j]);

}

Query9->Active = false;

Query9->Active = true;

TDataSet \*ds1 = DataSource9->DataSet;

length = ds1->RecordCount;

AnsiString \*str1 = new AnsiString[length];

i = 0;

for (ds1->First();!ds1->Eof; ds1->Next())

str1[i++] = DataSource9->DataSet->FieldByName("Group")->AsString;

for (j = 0; j < length; j++)

{

ComboBox13->Items->Add(str1[j]);

}

Query13->Active = false;

Query13->Active = true;

TDataSet \*ds2 = DataSource13->DataSet;

length = ds2->RecordCount;

AnsiString \*str2 = new AnsiString[length];

i = 0;

for (ds2->First();!ds2->Eof; ds2->Next())

str2[i++] = DataSource13->DataSet->FieldByName("Name")->AsString;

for (j = 0; j < length; j++)

{

ComboBox14->Items->Add(str2[j]);

}

Query14->Active = false;

Query14->Active = true;

TDataSet \*ds3 = DataSource14->DataSet;

length = ds3->RecordCount;

AnsiString \*str3 = new AnsiString[length];

i = 0;

for (ds3->First();!ds3->Eof; ds3->Next())

str3[i++] = DataSource14->DataSet->FieldByName("Spisanie")->AsString;

for (j = 0; j < length; j++)

{

ComboBox15->Items->Add(str3[j]);

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button8Click(TObject \*Sender)

{

UpdateComboBoxWrite\_off();

Edit10->Text = "";

Edit8->Text = "";

}

//---------------------------------------------------------------------------

void TForm1::UpdateComboBoxInventory()

{

ComboBox16->Clear();

ComboBox17->Clear();

ComboBox10->Clear();

Query8->Active = false;

Query8->Active = true;

TDataSet \*ds = DataSource8->DataSet;

int length = ds->RecordCount;

AnsiString \*str = new AnsiString[length];

int i = 0, j = 0;

for (ds->First();!ds->Eof; ds->Next())

str[i++] = DataSource8->DataSet->FieldByName("Name")->AsString;

for (j = 0; j < length; j++)

{

ComboBox16->Items->Add(str[j]);

}

Query9->Active = false;

Query9->Active = true;

TDataSet \*ds1 = DataSource9->DataSet;

length = ds1->RecordCount;

AnsiString \*str1 = new AnsiString[length];

i = 0;

for (ds1->First();!ds1->Eof; ds1->Next())

str1[i++] = DataSource9->DataSet->FieldByName("Group")->AsString;

for (j = 0; j < length; j++)

{

ComboBox17->Items->Add(str1[j]);

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button10Click(TObject \*Sender)

{

UpdateComboBoxInventory();

Edit11->Text = "";

Edit12->Text = "";

Edit14->Text = "";

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button15Click(TObject \*Sender)

{

Edit15->Text= "";

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button17Click(TObject \*Sender)

{

Edit13->Text = "";

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button19Click(TObject \*Sender)

{

Edit16->Text = "";

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N9Click(TObject \*Sender)

{

this->GroupBox2->Visible=false;

this->GroupBox3->Visible=false;

this->GroupBox1->Visible=false;

this->GroupBox4->Visible=false;

this->GroupBox6->Visible=false;

this->GroupBox7->Visible=false;

this->GroupBox5->Visible=false;

this->GroupBox13->Visible=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N10Click(TObject \*Sender)

{

this->GroupBox2->Visible=false;

this->GroupBox3->Visible=false;

this->GroupBox1->Visible=false;

this->GroupBox4->Visible=false;

this->GroupBox6->Visible=false;

this->GroupBox7->Visible=false;

this->GroupBox5->Visible=false;

this->GroupBox13->Visible=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N11Click(TObject \*Sender)

{

this->GroupBox2->Visible=false;

this->GroupBox3->Visible=false;

this->GroupBox1->Visible=false;

this->GroupBox4->Visible=false;

this->GroupBox6->Visible=false;

this->GroupBox7->Visible=false;

this->GroupBox5->Visible=false;

this->GroupBox13->Visible=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

/\* Table1->Append();

Table1->FieldValues["id\_name"] = ComboBox1->Text;

Table1->FieldValues["id\_group"] = ComboBox2->Text;

Table1->FieldValues["id\_packing"] = ComboBox3->Text;

Table1->FieldValues["id\_ed\_izm"] = ComboBox4->Text;

Table1->FieldValues["id\_producer"] = ComboBox5->Text;

Table1->FieldValues["k\_sh"] = Edit1->Text;\*/

/\* Table3->FieldValues["Date"] = Edit1->Text;

Table3->FieldValues["Sum"] = Edit3->Text;

Table3->FieldValues["ID\_sklad"] = ComboBox1->Text;\*/

// Table1->Post();

/\*Table1->Active=false;

Table1->Active=true; \*/

/\*Query1->Active=false;

Query1->Active=true;\*/

}

//---------------------------------------------------------------------------

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm2 \*Form2;

//---------------------------------------------------------------------------

\_\_fastcall TForm2::TForm2(TComponent\* Owner)

: TForm(Owner)

{

UpdateComboBox();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Button1Click(TObject \*Sender)

{

Form2->Close();

Form1->Enabled=true;

Form1->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Button2Click(TObject \*Sender)

{

Form1->Table1->Append();

Form1->Table1->FieldValues["dn"] = Form2->DateTimePicker1->Date;

Form1->Table1->FieldValues["dk"] = Form2->DateTimePicker2->Date;

Form1->Table1->FieldValues["id\_group"] = Form2->ComboBox1->ItemIndex+1;

Form1->Table1->FieldValues["id\_packing"] = Form2->ComboBox2->ItemIndex+1;

Form1->Table1->Post();

Form1->GZ();

Form2->Close();

Form1->Enabled=true;

Form1->Show();

Form1->GroupBox6->Visible=true;

}

//---------------------------------------------------------------------------

void TForm2::UpdateComboBox()

{

ComboBox1->Clear();

ComboBox2->Clear();

Query1->Active = false;

Query1->Active = true;

TDataSet \*dss = DataSource1->DataSet;

int length = dss->RecordCount;

AnsiString \*str = new AnsiString[length];

int i = 0, j = 0;

for (dss->First();!dss->Eof; dss->Next())

str[i++] = DataSource1->DataSet->FieldByName("Group")->AsString;

for (j = 0; j < length; j++)

{

ComboBox1->Items->Add(str[j]);

}

Query2->Active = false;

Query2->Active = true;

TDataSet \*dss1 = DataSource2->DataSet;

length = dss1->RecordCount;

AnsiString \*str1 = new AnsiString[length];

i = 0;

for (dss1->First();!dss1->Eof; dss1->Next())

str1[i++] = DataSource2->DataSet->FieldByName("Packing")->AsString;

for (j = 0; j < length; j++)

{

ComboBox2->Items->Add(str1[j]);

}

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit3.h"

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm3 \*Form3;

//---------------------------------------------------------------------------

\_\_fastcall TForm3::TForm3(TComponent\* Owner)

: TForm(Owner)

{

UpdateComboBox();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button1Click(TObject \*Sender)

{

Form3->Close();

Form1->Enabled=true;

Form1->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button2Click(TObject \*Sender)

{

Form3->Close();

Form1->Enabled=true;

Form1->Show();

Form1->GroupBox7->Visible=true;

}

//---------------------------------------------------------------------------

void TForm3::UpdateComboBox()

{

ComboBox1->Clear();

ComboBox2->Clear();

Query1->Active = false;

Query1->Active = true;

TDataSet \*dss = DataSource1->DataSet;

int length = dss->RecordCount;

AnsiString \*str = new AnsiString[length];

int i = 0, j = 0;

for (dss->First();!dss->Eof; dss->Next())

str[i++] = DataSource1->DataSet->FieldByName("Group")->AsString;

for (j = 0; j < length; j++)

{

ComboBox1->Items->Add(str[j]);

}

Query2->Active = false;

Query2->Active = true;

TDataSet \*dss1 = DataSource2->DataSet;

length = dss1->RecordCount;

AnsiString \*str1 = new AnsiString[length];

i = 0;

for (dss1->First();!dss1->Eof; dss1->Next())

str1[i++] = DataSource2->DataSet->FieldByName("Zakazi")->AsString;

for (j = 0; j < length; j++)

{

ComboBox2->Items->Add(str1[j]);

}

}

//---------------------------------------------------------------------------

#ifndef Unit1H

#define Unit1H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <Menus.hpp>

#include <DB.hpp>

#include <DBCtrls.hpp>

#include <DBGrids.hpp>

#include <DBTables.hpp>

#include <ExtCtrls.hpp>

#include <Grids.hpp>

#include <Buttons.hpp>

#include <ComCtrls.hpp>

//---------------------------------------------------------------------------

class TForm1: public TForm

{

\_\_published: // IDE-managed Components

TMainMenu \*MainMenu1;

TMenuItem \*N1;

TMenuItem \*N2;

TMenuItem \*N3;

TMenuItem \*N4;

TMenuItem \*N5;

TMenuItem \*N6;

TMenuItem \*N7;

TMenuItem \*N8;

TMenuItem \*N9;

TMenuItem \*N10;

TMenuItem \*N11;

TMenuItem \*N12;

TMenuItem \*ORZ1;

TMenuItem \*ORZ2;

TGroupBox \*GroupBox1;

TDataSource \*DataSource1;

TDBGrid \*DBGrid1;

TDBNavigator \*DBNavigator1;

TQuery \*Query1;

TDatabase \*Database1;

TSpeedButton \*SpeedButton1;

TGroupBox \*GroupBox2;

TDBGrid \*DBGrid2;

TDBNavigator \*DBNavigator2;

TDataSource \*DataSource2;

TQuery \*Query2;

TGroupBox \*GroupBox3;

TDBGrid \*DBGrid3;

TDBNavigator \*DBNavigator3;

TDataSource \*DataSource3;

TQuery \*Query3;

TGroupBox \*GroupBox4;

TDBGrid \*DBGrid4;

TDBNavigator \*DBNavigator4;

TQuery \*Query4;

TDataSource \*DataSource4;

TGroupBox \*GroupBox5;

TDBNavigator \*DBNavigator5;

TDBGrid \*DBGrid5;

TQuery \*Query5;

TDataSource \*DataSource5;

TGroupBox \*GroupBox6;

TDataSource \*DataSource6;

TDBGrid \*DBGrid6;

TDBNavigator \*DBNavigator6;

TQuery \*Query6;

TGroupBox \*GroupBox7;

TDataSource \*DataSource7;

TDBGrid \*DBGrid7;

TDBNavigator \*DBNavigator7;

TQuery \*Query7;

TGroupBox \*GroupBox8;

TComboBox \*ComboBox1;

TLabel \*Label1;

TComboBox \*ComboBox2;

TComboBox \*ComboBox3;

TComboBox \*ComboBox4;

TComboBox \*ComboBox5;

TEdit \*Edit1;

TLabel \*Label2;

TLabel \*Label3;

TEdit \*Edit2;

TEdit \*Edit3;

TEdit \*Edit4;

TButton \*Button1;

TButton \*Button2;

TGroupBox \*GroupBox9;

TLabel \*Label4;

TComboBox \*ComboBox6;

TButton \*Button3;

TButton \*Button4;

TComboBox \*ComboBox7;

TDateTimePicker \*DateTimePicker1;

TComboBox \*ComboBox10;

TEdit \*Edit5;

TEdit \*Edit9;

TGroupBox \*GroupBox10;

TLabel \*Label5;

TComboBox \*ComboBox8;

TButton \*Button5;

TButton \*Button6;

TComboBox \*ComboBox9;

TDateTimePicker \*DateTimePicker2;

TComboBox \*ComboBox11;

TEdit \*Edit6;

TEdit \*Edit7;

TGroupBox \*GroupBox11;

TLabel \*Label6;

TComboBox \*ComboBox12;

TButton \*Button7;

TButton \*Button8;

TComboBox \*ComboBox13;

TDateTimePicker \*DateTimePicker3;

TComboBox \*ComboBox14;

TEdit \*Edit8;

TEdit \*Edit10;

TLabel \*Label7;

TComboBox \*ComboBox15;

TGroupBox \*GroupBox12;

TLabel \*Label8;

TComboBox \*ComboBox16;

TButton \*Button9;

TButton \*Button10;

TComboBox \*ComboBox17;

TDateTimePicker \*DateTimePicker4;

TEdit \*Edit11;

TEdit \*Edit12;

TEdit \*Edit14;

TButton \*Button11;

TButton \*Button12;

TButton \*Button13;

TButton \*Button14;

TSpeedButton \*SpeedButton2;

TSpeedButton \*SpeedButton3;

TDataSource \*DataSource8;

TQuery \*Query8;

TQuery \*Query9;

TDataSource \*DataSource9;

TDataSource \*DataSource10;

TQuery \*Query10;

TQuery \*Query11;

TDataSource \*DataSource11;

TDataSource \*DataSource12;

TQuery \*Query12;

TQuery \*Query13;

TDataSource \*DataSource13;

TDataSource \*DataSource14;

TQuery \*Query14;

TGroupBox \*GroupBox13;

TDBGrid \*DBGrid8;

TDataSource \*DataSource15;

TQuery \*Query15;

TDBNavigator \*DBNavigator8;

TGroupBox \*GroupBox14;

TLabel \*Label9;

TEdit \*Edit15;

TDBGrid \*DBGrid9;

TDBGrid \*DBGrid10;

TDBNavigator \*DBNavigator9;

TDBNavigator \*DBNavigator10;

TDataSource \*DataSource16;

TDataSource \*DataSource17;

TQuery \*Query16;

TQuery \*Query17;

TGroupBox \*GroupBox15;

TLabel \*Label10;

TEdit \*Edit13;

TGroupBox \*GroupBox16;

TLabel \*Label11;

TEdit \*Edit16;

TTable \*Table1;

TTable \*Table2;

TTable \*Table3;

void \_\_fastcall N6Click(TObject \*Sender);

void \_\_fastcall N8Click(TObject \*Sender);

void \_\_fastcall N2Click(TObject \*Sender);

void \_\_fastcall N3Click(TObject \*Sender);

void \_\_fastcall N4Click(TObject \*Sender);

void \_\_fastcall N5Click(TObject \*Sender);

void \_\_fastcall ORZ1Click(TObject \*Sender);

void \_\_fastcall Button11Click(TObject \*Sender);

void \_\_fastcall Button12Click(TObject \*Sender);

void \_\_fastcall Button13Click(TObject \*Sender);

void \_\_fastcall Button14Click(TObject \*Sender);

void \_\_fastcall Button2Click(TObject \*Sender);

void \_\_fastcall Button4Click(TObject \*Sender);

void \_\_fastcall Button6Click(TObject \*Sender);

void \_\_fastcall Button8Click(TObject \*Sender);

void \_\_fastcall Button10Click(TObject \*Sender);

void \_\_fastcall Button15Click(TObject \*Sender);

void \_\_fastcall Button17Click(TObject \*Sender);

void \_\_fastcall N9Click(TObject \*Sender);

void \_\_fastcall N10Click(TObject \*Sender);

void \_\_fastcall N11Click(TObject \*Sender);

void \_\_fastcall Button1Click(TObject \*Sender);

// void UpdateComboBox();

private: // User declarations

public: // User declarations

\_\_fastcall TForm1(TComponent\* Owner);

void UpdateComboBoxGoods();

void UpdateComboBoxSale();

void UpdateComboBoxSupplies();

void UpdateComboBoxWrite\_off();

void UpdateComboBoxInventory();

void GZ();

// void UpdateComboBox();

};

//---------------------------------------------------------------------------

extern PACKAGE TForm1 \*Form1;

//---------------------------------------------------------------------------

#endif

//---------------------------------------------------------------------------

#ifndef Unit2H

#define Unit2H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <ComCtrls.hpp>

#include <DB.hpp>

#include <DBCtrls.hpp>

#include <DBTables.hpp>

//---------------------------------------------------------------------------

class TForm2: public TForm

{

\_\_published: // IDE-managed Components

TDateTimePicker \*DateTimePicker1;

TDateTimePicker \*DateTimePicker2;

TLabel \*Label1;

TLabel \*Label2;

TLabel \*Label3;

TLabel \*Label4;

TDataSource \*DataSource1;

TQuery \*Query1;

TButton \*Button1;

TButton \*Button2;

TComboBox \*ComboBox1;

TLabel \*Label5;

TComboBox \*ComboBox2;

TDataSource \*DataSource2;

TQuery \*Query2;

void \_\_fastcall Button1Click(TObject \*Sender);

void \_\_fastcall Button2Click(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm2(TComponent\* Owner);

void UpdateComboBox();

};

//---------------------------------------------------------------------------

extern PACKAGE TForm2 \*Form2;

//---------------------------------------------------------------------------

#endif

//---------------------------------------------------------------------------

#ifndef Unit3H

#define Unit3H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <ComCtrls.hpp>

#include <DB.hpp>

#include <DBTables.hpp>

//---------------------------------------------------------------------------

class TForm3: public TForm

{

\_\_published: // IDE-managed Components

TLabel \*Label1;

TLabel \*Label5;

TLabel \*Label4;

TLabel \*Label3;

TLabel \*Label2;

TDateTimePicker \*DateTimePicker2;

TDateTimePicker \*DateTimePicker1;

TComboBox \*ComboBox2;

TComboBox \*ComboBox1;

TButton \*Button2;

TButton \*Button1;

TLabel \*Label6;

TComboBox \*ComboBox3;

TQuery \*Query1;

TQuery \*Query2;

TDataSource \*DataSource1;

TDataSource \*DataSource2;

void \_\_fastcall Button1Click(TObject \*Sender);

void \_\_fastcall Button2Click(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm3(TComponent\* Owner);

void UpdateComboBox();

};

//---------------------------------------------------------------------------

extern PACKAGE TForm3 \*Form3;

//---------------------------------------------------------------------------

#endif