

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

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

Кафедра «Прикладная математика и информатика»  
(наименование)

09.03.03 Прикладная информатика  
(код и наименование направления подготовки, специальности)

«Бизнес информатика»  
(направленность (профиль) / специализация)

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка автоматизированной информационной системы учета компьютерной техники и комплектующих организации (на примере ООО «Уральский Пружинный Завод»)»

Студент

Е.О. Пантелеев

(И.О. Фамилия)

(личная подпись)

Руководитель

Н.Н. Казаченок

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

Тольятти 2020

## **Аннотация**

В процессе написания работы разработана информационная система учета компьютерной техники на предприятии «Уральский пружинный завод».

Дано описание предприятия, выполнен структурный анализ бизнес-процессов с использованием методологий IDEF0 и DFD, описана организационная структура, описана сущность задачи автоматизации.

Рассмотрены основные аналогичные системы, принято решение о разработке собственно системы подобного плана. Выработаны требования к системе, выбраны средства проектирования и разработки.

Далее проводится концептуальное, логическое и физическое моделирование базы данных.

После этого база данных реализована с использованием СУБД MS SQL Server, а также разработано полнофункциональное приложение для взаимодействия с базой данных.

Проведено экономическое обоснование проекта.

Планируется внедрение информационной системы на предприятии.

## Оглавление

Введение.....	4
Глава 1 Анализ задачи учета компьютерной техники на предприятии .....	6
1.1 Техничко-экономическая характеристика учета компьютерной техники.....	6
1.2 Концептуальное моделирование задачи учета компьютерной техники.....	12
1.3 Постановка задачи .....	19
1.4 Анализ существующих разработок и обоснование выбора технологии проектирования .....	26
1.5 Требования к разрабатываемой системе .....	35
Глава 2 Разработка и реализация проектных решений .....	36
2.1 Логическое моделирование предметной области.....	36
2.2 Физическое моделирование АИС .....	43
2.3 Технологическое обеспечение задачи .....	51
2.4 Контрольный пример реализации проекта и его описание .....	53
Глава 3 Оценка и обоснование экономической эффективности проекта .....	59
3.1 Выбор и обоснование методики расчёта экономической эффективности .....	59
3.2 Расчет показателей экономической эффективности .....	63
Заключение .....	69
Список используемой литературы и источников .....	71
Приложение А Сценарий создания базы данных .....	73
Приложение Б Исходный код приложения .....	77

## Введение

На сегодняшний день трудно себе представить какую-либо фирму, которая бы не пользовалась программными продуктами для различного вида учета, а хранила бы всю информацию на бумаге. В любой фирме существует набор программных средств, с помощью которых бухгалтер либо работник фирмы ведет учет и фиксирует акты выполненных работ. Это во многом облегчает процедуру учета и составления актов выполненных работ.

Современные информационные системы для хранения своих данных в качестве основы используют реляционные базы данных. Технологии и основные концепции данных имеют тенденцию к тому, чтобы складываться постепенно. Они достаточно тесно связаны развитием основных систем, призванных осуществлять автоматизированную обработку информации.

Достаточно широкое использование основных баз данных относительно различных категорий пользователей приводит к созданию интерфейсов, которые требуют минимум времени для освоения основных средств управления системой, а с другой стороны оказывают воздействие на построение достаточно мощных и гибких систем управления базами данных (СУБД), которые имеют развитые средства защиты данных от преднамеренного или случайного уничтожения. В последние годы появились новые средства для автоматизации разработок, которые позволяют создавать пользователям базы данных, в том числе и тем, которые не владеют основами теории БД.

Предметная область исследований выпускной квалификационной работы (ВКР) – учет компьютерной техники и комплектующих на предприятии «Уральский Пружинный Завод». На предприятии происходит периодический ремонт, перемещение и списание оборудования, и необходимо организовать учетную деятельность в компании.

Целью работы является разработка программы учета движения основных средств и комплектующих вычислительной техники и операций над ней для автоматизации учета на предприятии.

В связи с этим для систематического и непрерывного учета выполненных работ необходимо разработать информационную систему.

Практическая ценность такой системы заключается в том, что она позволяет автоматизировать ручной труд как инженера, так и офис-менеджера по составлению необходимой документации, по контролю выполняемых работ и оформлению соответствующей документации.

Актуальность работы заключается в том, что единая база данных по ведению учета компьютерной техники на предприятии позволяет решить множество проблем. В частности – внедрение подобной базы позволяет упорядочить и упростить учёт техники, значительно снижает воздействие человеческого фактора на операции учета, позволяет существенно снизить количество ошибок при оформлении техники.

Задачи, которые должны быть решены для достижения цели:

- анализ процессов документооборота внутри фирмы и выявление проблем;
- постановка задачи;
- анализ существующих программных средств для ведения учета выполненных работ и перемещения оборудования;
- моделирование системы;
- выбор средств реализации;
- инфологическое и физическое проектирование базы данных;
- проектирование и разработка приложения.

# **Глава 1 Анализ задачи учета компьютерной техники на предприятии**

## **1.1 Техничко-экономическая характеристика учета компьютерной техники**

### **1.1.1 Характеристика предприятия**

ООО «Уральский Пружинный Завод» — это специализированное производственное предприятие по изготовлению ассортимента пружин горячей и холодной навивки, основанное в 2010 г. ООО «Уральский Пружинный Завод» один из лидеров на российском рынке по продаже пружин сжатия, растяжения, кручения холодной и горячей навивки, пружин цилиндрических, винтовых, рессорного подвешивания тележек, грузовых железнодорожных вагонов, железнодорожных комплектующих. Предприятием освоена широкая номенклатура пружин для машиностроительной, автомобильной, энергетической отраслей, а также вагоностроения.

Направления деятельности завода - проектирование, производство, реализация пружин растяжения, сжатия, кручения для различных отраслей промышленности:

- железнодорожного вагоностроения,
- автомобильной отрасли,
- сельскохозяйственного машиностроения,
- пищевой отрасли,
- станкостроения,
- энергомашиностроения,
- нефтегазового и дробильно-сортировочного оборудования и др.

Структура компании. Производственная площадка находится в г. Белорецк, республика Башкортостан. Офисы продаж представлены в

следующих городах России: г. Магнитогорск, г. Челябинск, г. Санкт-Петербург, г. Москва, а также в Германии.

На сегодняшний день предприятие сотрудничает с местными предпринимателями, с крупными потребителями на внешнем рынке, как в Европе, так и в азиатском регионе.

Предприятием взят курс на приобретение современного оборудования и освоение видов продукции, наиболее востребованных на рынке, также осуществляются тестовые партии пружин для достижения удовлетворенности требований покупателей.

Новый этап развития начался в 2011 г. с установкой линии по производству пружин цилиндрических винтовых для подвижного состава железных дорог. В 2012 г. было получено первое разрешение на применение Знака соответствия Системы сертификации на Федеральном железнодорожном транспорте на основании полученного сертификата соответствия продукции. Объемы производства и ассортимент продукции за прошедший с этого момента период значительно выросли. Сегодня на предприятии работает более 300 человек.

Деятельность предприятия направлена на повышение производительности труда, снижение себестоимости продукции, уменьшение сроков поставки, снижение прочих издержек и потерь производства. Для этого предприятие осваивает современные технологии менеджмента: производственную систему бережливого производства – «Lean Production», которая включает такие понятия, как ценность и поток, вытягивающее производство, «Точно в срок», Канбан, Кайдзен, TQM, 5S.

Уральский пружинный завод имеет сертификаты французской фирмы UTAS о соответствии системы менеджмента качества предприятия требованиям международных стандартов ISO 9001:2015 «Системы менеджмента качества. Требования» и ISO/TS 16949:2016 «Системы менеджмента качества. Особые требования по применению ИСО 9001:2015

для организаций - производителей серийных и запасных частей для автомобильной промышленности».

Качество продукции завода подтверждается периодическими аудитами со стороны крупных потребителей, а также соответствующими сертификатами.

### **1.1.2 Краткая характеристика подразделения и его видов деятельности**

Автоматизированная информационная система учета компьютерной техники разрабатывается для отдела ИТ ООО «Уральский Пружинный Завод».

Предметом исследования в бакалаврской работе является область функционирования ИТ-отдела компании, возникающая в процессе взаимоотношения между ИТ-отделом и другими подразделениями компании, а также внутри самого отдела.

При анализе деятельности ИТ-отдела следует учитывать то, что работа специалистов отдела является весьма сложной и напряженной. Прежде всего, потому что современный уровень развития информационных технологий очень высок и это требует повышенного внимания с их стороны. Это касается как аппаратного обеспечения, которое является предметом работы инженеров-электронщиков, так и программного обеспечения, за которое отвечают инженеры-программисты и системные администраторы. Современные компьютеры, сервера, локальные и глобальные сети, информационные системы собственной разработки и приобретенные – это очень обширная и сложная область, требующая глубоких знаний сотрудников отдела. Все сотрудники являются высококлассными специалистами в своей области, что является залогом успеха решения тех сложных задач, которые стоят перед отделом. Но поддержание этой классности в очень быстро меняющемся мире информационных технологий, поиск оптимальных путей решения возникающих проблем, грамотная

реализация найденных решений требуют от сотрудников ИТ-отдела значительных усилий.

Второй спецификой и трудностью работы ИТ-отдела является то, что сотрудники отдела нередко оказываются в центре многих проблем компании, решение которых лежит не в области деятельности какого-то одного подразделения компании, а на пересечении ответственности двух и более подразделений. Нередко бывает так, что решение проблем взаимодействия подразделений связано с технической возможностью решения таких проблем, определить которую, а также установить правила и протоколы взаимодействия может только ИТ-отдел.

Не меньшую трудность в работе ИТ-отдела создает то, что он нужен всем. Практически на каждом рабочем месте офисных работников компании, а также на многих местах на производстве установлены компьютеры и оргтехника. Все они объединены в локальную сеть со сложной архитектурой. Все сотрудники пользуются сетью Интернет, электронной почтой, общесистемным и специальным программным обеспечением, среди которого как общеупотребительные пакеты (например, компании Microsoft), так и специальное программное обеспечение. Работоспособность всего этого большого организма поддерживается усилиями немногочисленного коллектива ИТ-отдела, вынужденного решать множество проблем, возникающих в ходе работы всех подразделений компании.

Таким образом, работа ИТ-отдела весьма сложна и трудоемка. Повысить эффективность работы отдела, увеличить отдачу каждого сотрудника, уменьшить производственные потери можно только благодаря автоматизации деятельности предприятия в целом и каждого его подразделения в отдельности.

Укрупнено функции ИТ-отдела можно определить так:

- поддержание работы сети,
- поддержание работы технических средств,
- разработка и поддержание работы программного обеспечения.

Как видно из данного перечня функций, учет компьютерной техники не является прямой обязанностью отдела ИТ. Действительно, учетом и инвентаризацией объектов основных средств, которыми являются компьютеры и оргтехника, на предприятии занимаются сотрудники бухгалтерии и ведут этот учет в соответствии с правилами бухгалтерского учета. Однако на практике оказывается, что для отдела ИТ иметь данные об имеющихся на предприятии компьютерах и оргтехнике, их распределении по подразделениям и сотрудникам важнее, чем для бухгалтерии.

Это связано с тем, что отделу ИТ необходимо не просто вести учет имеющейся компьютерной техники, но и организовать эффективную работу этой компьютерной техники в составе единой информационной системы предприятия. Эффективность работы такой системы возможна лишь при согласованности всех входящих в нее технических элементов и программного обеспечения. Для достижения этой цели нередко требуются данные, гораздо более широкие, чем может предоставить сотрудникам отдела ИТ бухгалтерский учет инвентаризационных объектов основных средств.

Задача учета компьютерной техники тем более важна, что отдел ИТ непосредственно участвует в процессе бухгалтерского и складского учета компьютерной техники. Именно начальник отдела ИТ от имени руководителя предприятия заключает договора на поставку компьютерной техники, запасных частей, расходных материалов, проведение ремонтных работ. Все первичные документы на поставку компьютерной техники, акты выполненных работ, складские документы по движению материалов на складе отдела ИТ подписываются начальником отдела ИТ и после этого сдаются в бухгалтерию. Распределение компьютерной техники по подразделениям предприятия и сотрудникам подразделений – это тоже прерогатива отдела ИТ. Документы списания материалов на ремонт и модернизацию компьютерной техники, актов ремонта, выхода из строя так же находятся в компетенции начальника отдела ИТ. Хотя данные

бухгалтерского учета компьютерной техники с юридической точки зрения более важны, первичными данными для бухгалтерского учета являются документы и данные учета, ведущегося в отделе ИТ.

Таким образом, учет компьютерной техники несмотря на то, что не является прямой обязанностью отдела ИТ, входит в число первостепенных задач, которые нужно решить руководству отдела ИТ для организации его эффективного функционирования.

### **1.1.3 Сущность задачи автоматизации учета компьютерной техники на предприятии**

Актуальность задачи автоматизации обусловлена тем, что в настоящее время на предприятии представлено просто огромное количество оргтехники и компьютерной техники. И вести адекватный учет всего этого огромного парка техники без использования современных компьютерных технологий просто невозможно. Устаревшие способы ведения баз данных, такие как использование MS Access и MS Excel – не позволяют адекватно хранить подобные большие объемы информации и вести учет всех операций над оборудованием, по той причине что это динамический процесс, и ежедневное внесение данных в подобном виде просто неудобно. Кроме того – подобные средства не удовлетворяют современным требованиям, они не многопользовательские, и их использование через локальную сеть либо сильно затруднено, либо просто невозможно. Поэтому разработка подобной системы является очень актуальной задачей для предприятия.

В результате автоматизация позволяет получить следующие результаты:

- Обеспечение эффективной работы отдела ИТ;
- обеспечения непрерывного функционирования бизнес-процессов;
- создание единого инструмента для учета компьютерной техники на предприятии;
- снижения трудоемкости операций по подготовке отчетов и данных.

- Повышение качества обслуживания смежных подразделений за счет:
- увеличение «прозрачности» учета компьютерной техники, находящейся в подразделениях;
- сокращение времени проведения ремонтов;
- обеспечение руководства достоверной оперативной информацией;
- создание, поддержание в актуальном состоянии, надежное хранение информации, обеспечение защиты от несанкционированного доступа к ней.

Таким образом, использование информационной системы, выполняющей автоматизацию работы отдела ИТ в части учета компьютерной техники поднимает деятельность отдела ИТ на качественно иной уровень, что дает неоспоримый экономический эффект.

## **1.2 Концептуальное моделирование задачи учета компьютерной техники**

Инженер ИТ-отдела занимается выполнением большого количества задач, связанных с учетом компьютерной техники, расходных материалов и запчастей. В частности, среди подобных задач можно выделить следующие:

- учет закупки оргтехники,
- постановка оргтехники на учет и списание,
- учет перемещения техники по подразделениям предприятия,
- учет производимых ремонтов, замены частей, профилактики,
- учет заправки принтеров, ксероксов и другого печатающего оборудования,
- учет картриджей и их заправка,
- получение всевозможных отчетов и аналитических форм,

– планирование деятельности службы обслуживания оргтехники.

Таким образом, функциональную модель деятельности инженера ИТ-отдела можно представить с помощью следующей модели, изображенной на рисунке 1.

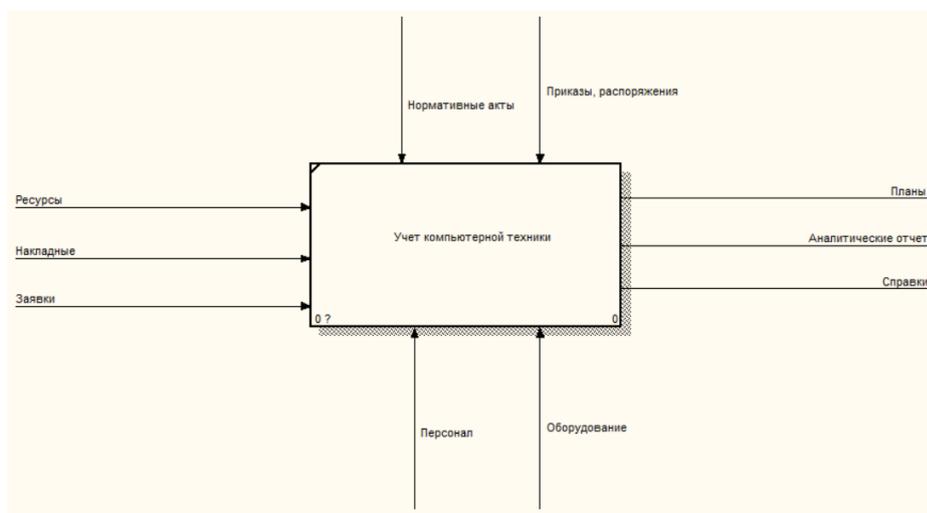


Рисунок 1 – Функциональная модель деятельности отдела ИТ-отдела

Основные участники процесса учета, анализа и планирования на предприятии:

- 1) инженер по обслуживанию оргтехники,
- 2) менеджер по закупке материальных ресурсов,
- 3) бухгалтер,
- 4) заведующий ИТ-отделом;

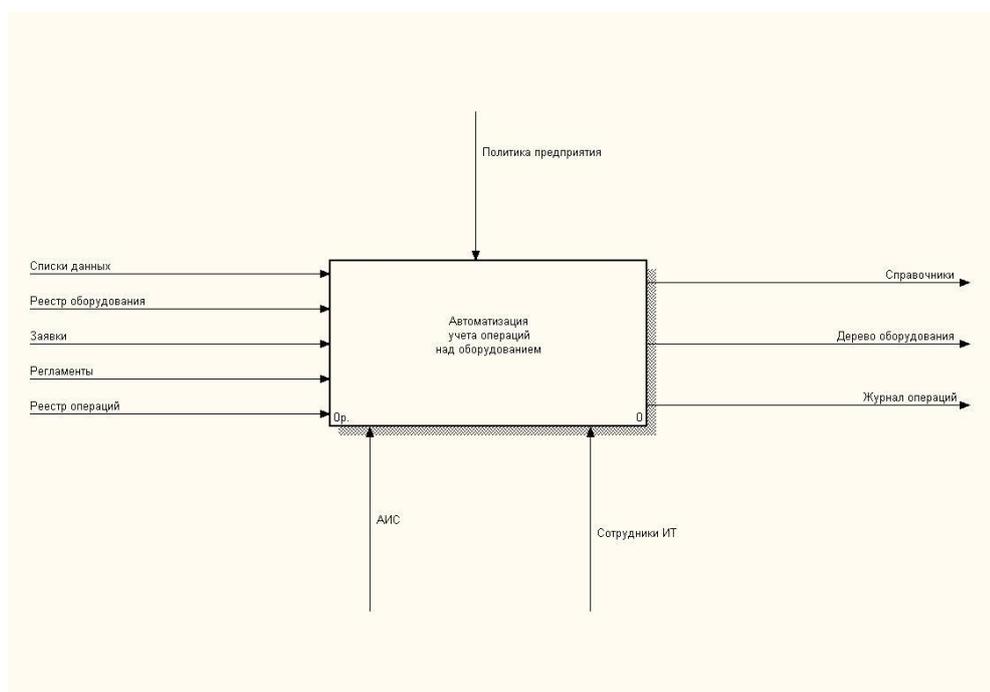
В настоящее время на предприятии не предусмотрено четкого распределения ролей сотрудников по тем или иным сферам деятельности. То есть, например инженер по обслуживанию может как заниматься обслуживанием компьютерной техники, так и производить закупки комплектующих и оборудования.

В настоящее время учет компьютерной техники, оргтехники, и расходных материалов – на предприятии ведется с использованием такого

средства как Excel. Данное средство уже достаточно устарело, и не позволяет вести полный учет, в частности в процессе учета затруднены операции перемещения, ремонтов, учета ресурсов и т.д. Кроме того, следует отметить что учет операций не всегда своевременный, так как данные не всегда оперативно вносятся в учетные документы. Отсутствуют возможности формирования аналитических отчетов, многопользовательской работы, что также существенно затрудняет учет техники на предприятии.

Отсутствие средств автоматизации учета парка компьютерной техники не позволяет в полном объеме проводить своевременный учет и контроль состояния ресурсов компьютерного парка. Отсутствие аналитических возможностей в существующей системе учета ресурсов компьютерного парка лишает руководство возможности аргументировано планировать обновление и расширение компьютерного парка – оно выполняется от случая к случаю и необоснованно.

Для более наглядного представления проектируемой задачи были разработаны «IDEF-диаграммы» функций разрабатываемой системы, приведенные на рисунках ниже.



## Рисунок 2 – Диаграмма автоматизации учета выполненных работ нулевого уровня

Как видно из рисунка 2 данными для разрабатываемой системы являются данные:

- Списки данных по моделям и маркам оборудования, структуре предприятия;
- Реестр имеющегося оборудования;
- Реестр операций, которые могут быть выполнены над оборудованием;
- Заявки пользователей на проведение операций;
- Регламенты на проведение технического обслуживания.

Оформление документов ведется посредством АИС. Оформление выполненных работ происходит согласно политике предприятия.

Результатом процесса являются:

- Заполненные справочники системы;
- Дерево оборудования;
- Журнал выполненных операций.

Более подробная детализация задачи учета компьютерной техники представлена на диаграмме декомпозиции (рисунок 3).

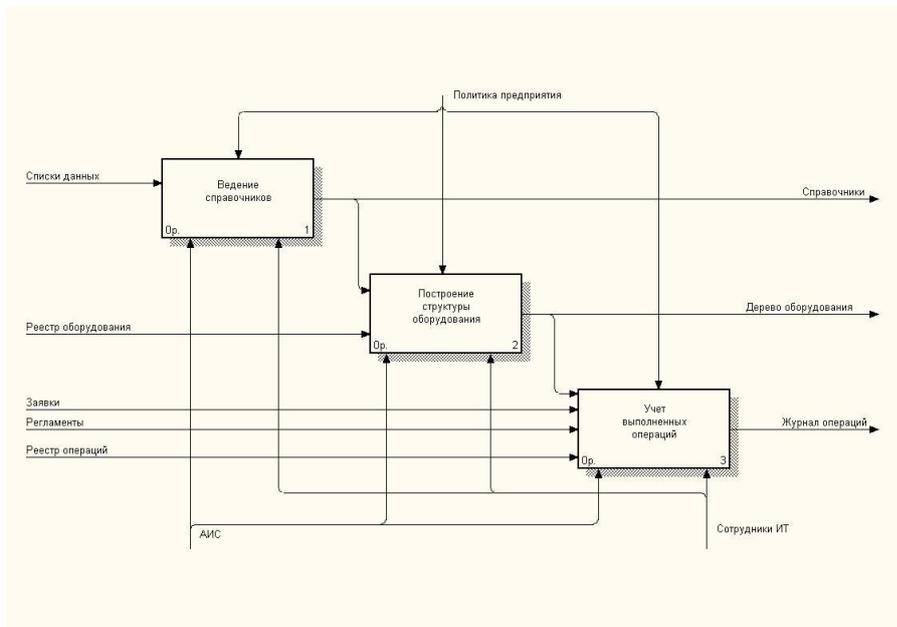


Рисунок 3 – Декомпозиция процесса учета компьютерной техники  
 Декомпозиция блока «Ведение справочников» представлена на рисунке

4.

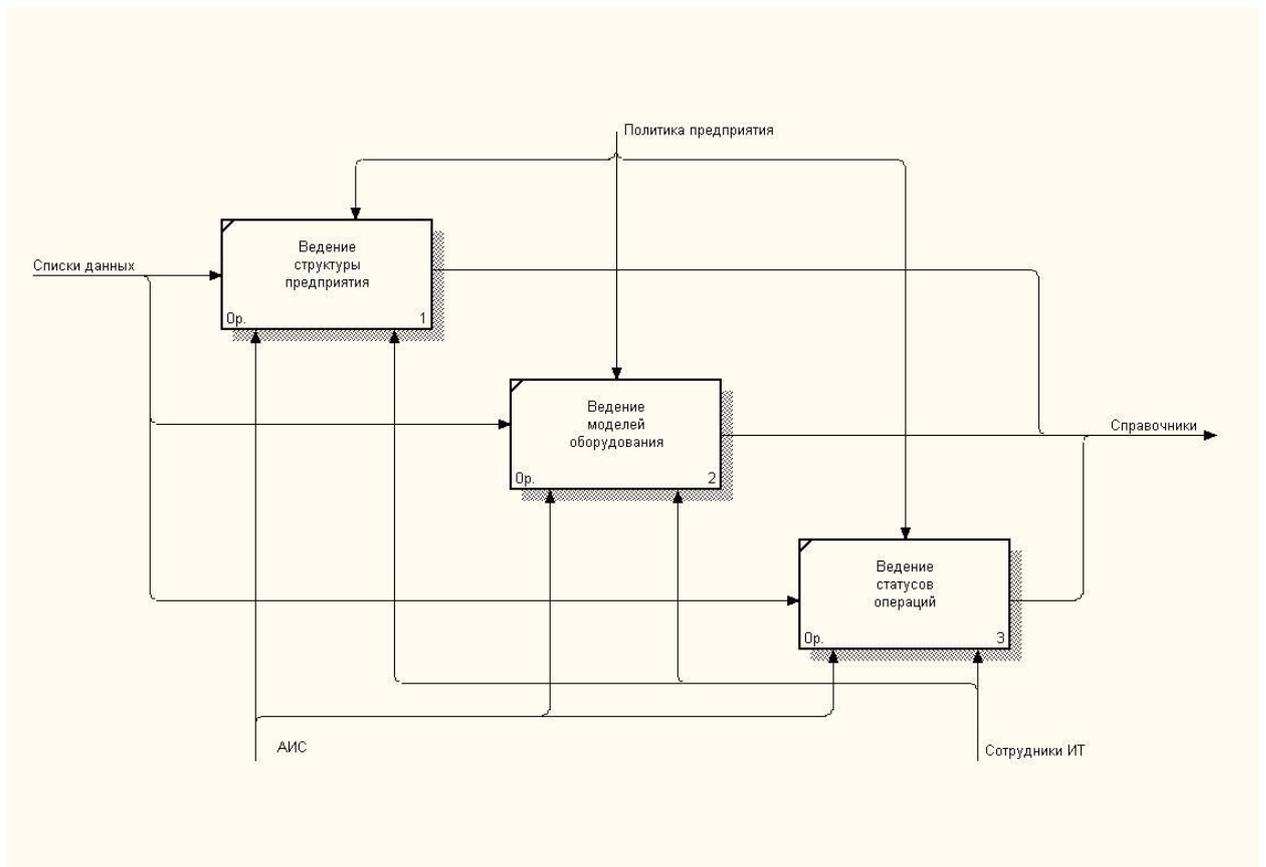


Рисунок 4 – Детализация блока «Ведение справочников»

Подзадачами блока «Ведение справочников» являются:

- Ведение структуры предприятия;
- Ведение моделей оборудования;
- Ведение статусов операций.

Все данные вносятся сотрудниками отдела ИТ в АИС согласно политике предприятия. Все внесенные данные записываются в БД АИС.

Декомпозиция блока «Построение структуры оборудования» представлена на рисунке 5.

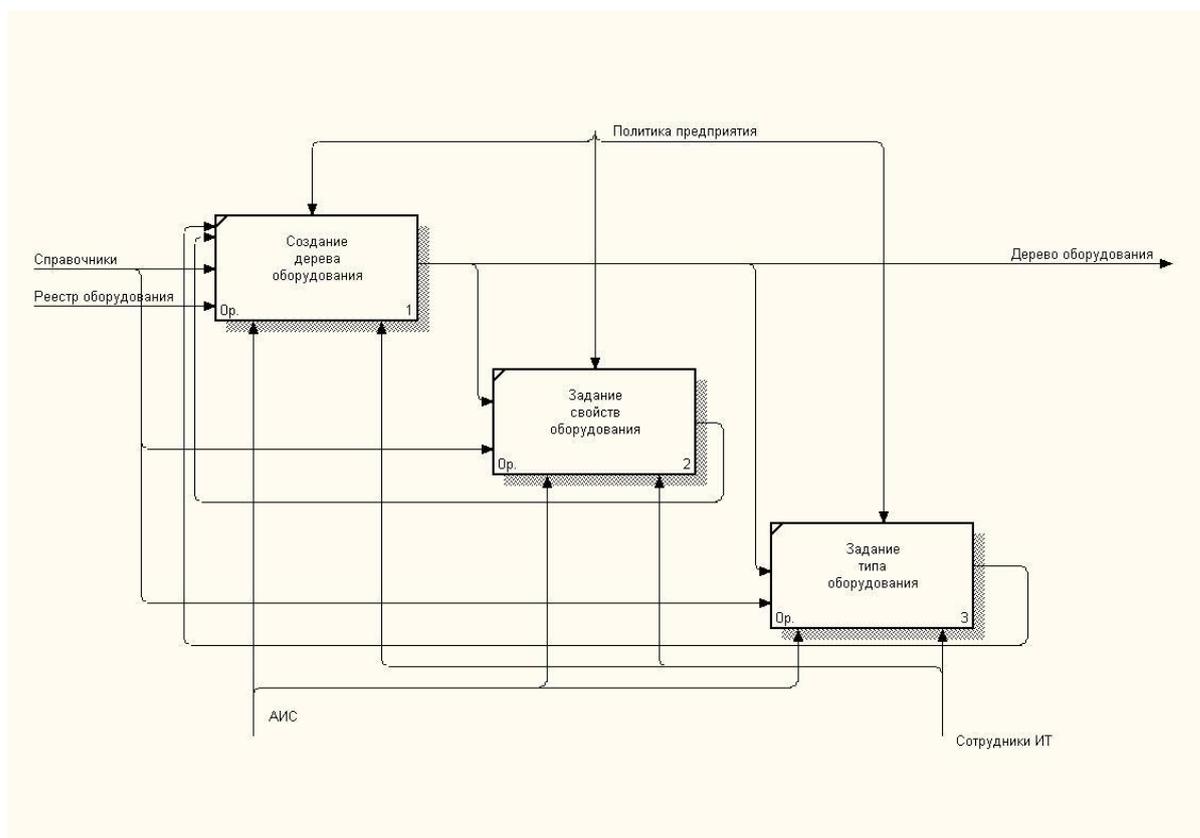


Рисунок 5 – Детализация блока «Построение структуры оборудования»

Подзадачами блока «Построение структуры оборудования» являются:

- Создание дерева оборудования;

- Задание свойств оборудования;
- Задание типа оборудования.

Все данные вносятся сотрудниками отдела ИТ в АИС согласно политике предприятия. Все внесенные данные записываются в БД АИС. Результатом работы является записанное в базу данных дерево оборудования.

Декомпозиция блока «Учет выполнения операций» представлена на рисунке 6.

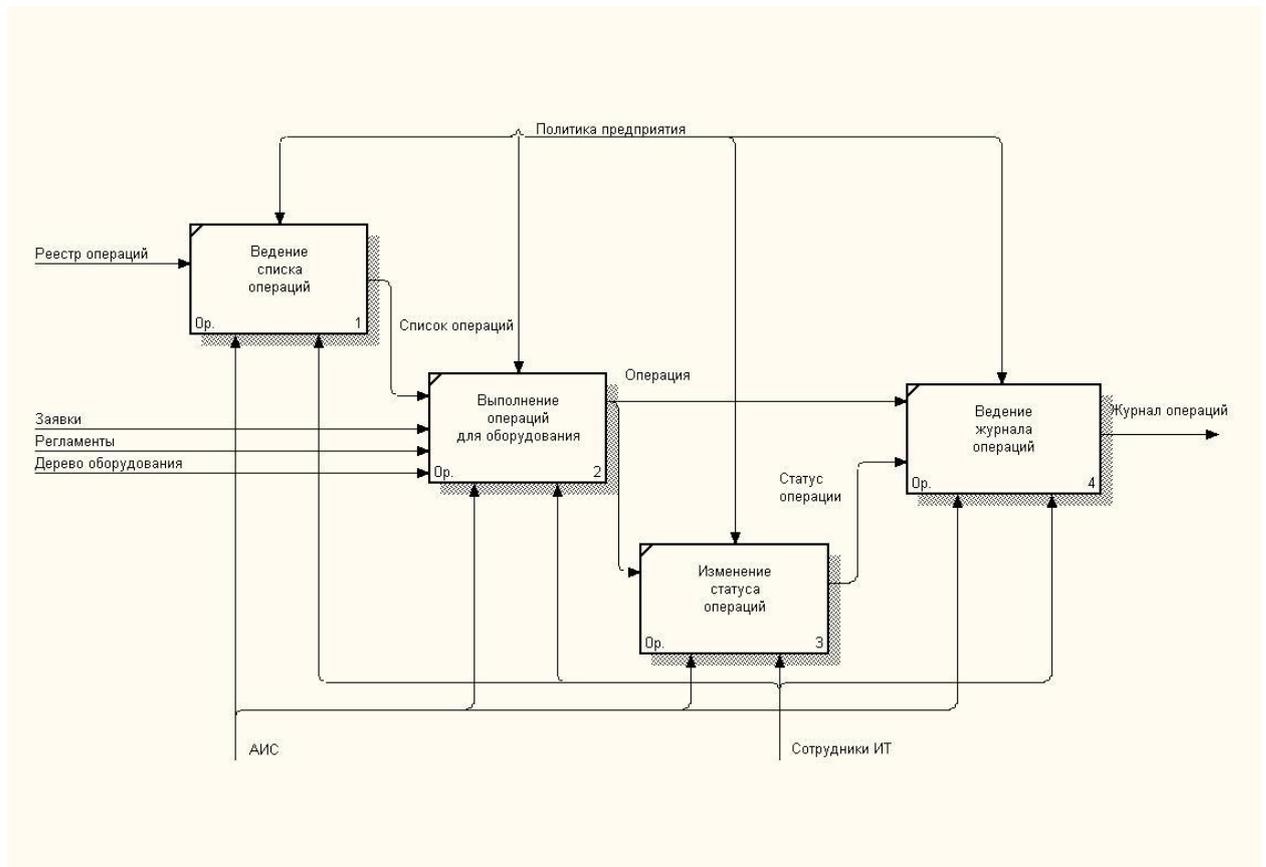


Рисунок 6 – Детализация блока «Учет выполнения операций»

Блок подразделяется на следующие подзадачи:

- Ведение списка операций;
- Фиксация выполнения операции над оборудованием;
- Изменение статуса операции по мере ее выполнения;
- Ведение журнала операций.

Входными данными, как и в первоначальном варианте являются заявки на проведение операций, поступающие от сотрудников подразделений и от начальника отдела в соответствии с регламентами на проведение технического обслуживания. Бумажный реестр операций теперь преобразуется в справочник операций, хранящийся в базе данных, Дерево оборудования, созданное в базе данных на предыдущем этапе, также является входом данного блока. Выходом блока является заполненный и хранящийся в базе данных журнал операций. Все работы выполняют сотрудники отдела ИТ в соответствии с политикой предприятия.

Разрабатываемая система должна отвечать следующим принципам:

- Организационная масштабируемость. Использовать систему на предприятии можно сначала на одном компьютере. После того как в базе данных будет накапливаться информация о компьютерной технике на предприятии – можно вводить в действие систему на новых рабочих местах;
- Корпоративность. Доступ к системе должен быть разграничен для различных видов пользователей. Должно быть предусмотрено несколько различных ролей доступа;
- Управляемость. Если возникает необходимость в перенастройке системы – спроектированная структура должна обеспечивать возможность быстрой перенастройки и наладки;
- Защищенность. Доступ к системе должен быть защищен как дискреционным методом, так и с помощью шифрования канала данных.

В данном параграфе мы показали учет операций над оборудованием в системе «КАК ЕСТЬ» и «КАК БУДЕТ».

### **1.3 Постановка задачи**

### **1.3.1 Цель и назначение автоматизированного варианта решения задачи учета компьютерной техники на предприятии**

При анализе процесса выполнения задачи учета компьютерной техники были выявлены такие основные недостатки:

- Большие затраты времени на выполнение процесса учета оборудования;
- Высокая стоимость выполнения процесса;
- Человеческий фактор, то есть в текущей системе учета существует возможность допуска большого количества ошибок, что не обеспечивает достаточный уровень надежности и защищенности системы.

Решением всех этих проблем может стать внедрение информационной системы учета компьютерной техники.

Цель разрабатываемой информационной системы:

- автоматизация деятельности по учету компьютерной техники;
- обеспечение хранения данных в электронном виде;
- подготовка печатных форм выходных документов.

При внедрении указанной информационной системы будут решены три группы задач:

- Улучшение экономических показателей предприятия. После внедрения системы количество трудовых и стоимостных ресурсов, затрачиваемых на учет компьютерной техники на предприятии – должно значительно сократиться за счет оптимизации процесса учета и внедрения типовых справочников;
- Улучшение показателей качества обработки информации в системе, за счет уменьшения количества ошибок как при вводе данных, так и при выполнении аналитических отчетов;
- Повышение удовлетворенности обслуживанием сотрудников предприятия и его клиентов.

Кроме того, снизится загруженность и утомляемость работников, повысится их мотивация к труду, что позитивно скажется на общей эффективности работы отдела.

### **1.3.2 Общая характеристика организации решения задачи на ЭВМ**

Исходными документами для разрабатываемой системы служат:

- Журнал перемещения техники;
- Акты перемещения;
- Акты списания.

Эти же типы документов присутствуют в информационной системе.

Информация выше вносится в информационную систему, на основании чего формируются следующие отчеты, являющиеся выходными данными:

- Работы по сотруднику;
- Перемещение техники за период;
- Отчет по списанию.

Функциональные возможности системы можно разделить на три основные части:

- Ведение списков и справочников;
- Построение структуры оборудования;
- Учет выполненных операций.

Ввод в систему справочников – позволяет значительно сократить объем вводимых вручную в систему данных. Ведение справочников обеспечивает заблаговременный ввод данных в специализированные таблицы, после чего пользователь при вводе документов имеет возможность указывать уже введенные данные в выпадающих списках, либо с помощью форм поиска. При этом значительно увеличивается скорость ввода данных информации и существенно сокращается количество ошибок ввода, за счет унификации данных.

при заполнении основных справочников и документов по выполнению операций. Это такие справочники как:

- Модели оборудования;
- Марки оборудования;
- Типы элементов оборудования;
- Сотрудники предприятия;
- Подразделения предприятия;
- Должности сотрудников предприятия;
- Статусы операций, выполняемых над оборудованием.

Необходимо предусмотреть возможность объединения записей справочника в группы, объединяющие записи по смысловому или функциональному назначению. В этом случае в левой части списка должно отображаться дерево групп, а в списке должны отображаться записи только этой группы, при этом режим отображения списка по группам или без групп должен настраиваться.

Программа должна осуществлять выгрузку списка элементов любого справочника в программу MS Excel. Список должен выводиться в соответствии с установленными настройками фильтра, сортировки и настройки полей. Результатом выгрузки становится сформированный xls-файл, который автоматически открывается в программе MS Excel. Далее пользователь может использовать всю мощь этого программного продукта для дальнейшего редактирования и форматирования данных.

При запуске программы должно проверяться наличие базы данных, с которой ей предстоит работать. Если программа запускается впервые и параметры связи с базой данных не настроены, то необходимо открыть окно, в котором пользователю предлагается задать все параметров связи с базой данных. Эти параметры записываются в ini-фал и считываются оттуда при последующих запусках. При каждом запуске программа проверяет в базе данных наличие каждой из необходимых таблиц и в случае обнаружения отсутствия какой-либо из таблиц автоматически создает ее с нужной структурой при помощи SQL-запроса. Таким образом, при первом запуске

программы будет не только настроена связь с базой данных, но и автоматически создана ее структура и структура всех таблиц.

### **1.3.3 Формализация расчетов подзадач**

Подзадача построения структуры оборудования должна предоставлять возможность отображать состав оборудования предприятия. Состав оборудования должен быть представлен в виде дерева, так чтобы можно было отобразить подчиненность элементов оборудования друг другу.

В рамках подсистемы учета выполненных операций программа должна реализовывать следующие задачи:

- ведение списка операций, списка категорий операций;
- ведение списка операций по умолчанию для типов и моделей оборудования;
- ведение списка допустимых операций для типов оборудования;
- выполнение допустимой операции для конкретного элемента оборудования с заполнением всех данных операции вручную;
- выполнение для конкретного элемента оборудования операцию по умолчанию с заполнением всех данных операции автоматически;
- ведение журнала выполненных операций.

Рассмотрим каждую из задач подробнее.

Задача ведения списка операций и их категорий должна предоставлять возможность предварительного заполнения справочника операций и внесения по каждой операции всех необходимых данных. Однако, ввод новой операции в справочник возможен и при непосредственном оформлении выполнения операции.

Задача ведения операций по умолчанию предназначена для того, чтобы облегчить жизнь пользователя. Дело в том, что в большинстве случаев пользователю бывает лень вводить данные о выполнении операций, не смотря на то какая бы современная и высоко функциональная автоматизированная система для этого не была бы предназначена. Цель данной задачи – довести оформление выполнения операции до «нажатия

одной клавиши». Предполагается, что для каждого типа оборудования существует некоторая операция из списка операций, которая выполняется для него чаще всего, и выражение «выполнить операцию» для оборудования данного типа означает чаще всего выполнить именно определенную операцию из списка операций. В этом случае для операции в списке операций указывается тип оборудования, для которого эта операция является наиболее часто выполняемой или другими словами является операцией по умолчанию. Примером операции по умолчанию может являться операция «Заправка картриджа» для типа оборудования «Картридж».

Ведение списка допустимых операций для типов оборудования призвано решить задачу ограничить количество предлагаемых пользователю операций для конкретного элемента при оформлении операции вручную. Дело в том, что не все операции применимы для всех типов оборудования, а в некоторых случаях выполнение некоторых операций для некоторых типов оборудования не имеет смысла. Например, не имеет смысла выполнение операции «Заправка картриджа» для оборудования типа «Сканер», в то же время операция «Ремонт» имеет смысл для «Оборудования» всех типов.

Оформление операции вручную заключается в том, что для конкретного элемента оборудования пользователь сам выбирает операцию и заполняет все ее реквизиты вручную. Выполнение операции означает занесение информации о факте ее выполнения в журнал операций.

Задача выполнения операции по умолчанию для конкретного элемента оборудования реализует возможность оформления операции «нажатием одной клавиши». Если такая задача выполняется для конкретного элемента оборудования программа должна сама выбрать какую операцию необходимо выполнить и заполнить все ее реквизиты. Пользователь должен иметь возможность изменить все предложенные по умолчанию реквизиты вручную.

Операция по ходу своего выполнения может проходить несколько этапов. Текущий этап выполнения операции определяется статусом операции. Справочник статусов должен позволять вести список состояний

операции начиная от «заявка принята к выполнению», до «операция успешно завершена». Статус операции последовательно меняется по мере прохождения операции по этапам.

Задача ведения журнала операций должна предоставлять возможность просмотра сформированного журнала операций. Каждая запись журнала – это информация о факте выполнения операции, с указанием всех ее реквизитов. Должен формироваться итог по сумме и количеству выведенных в журнал записей. Должна быть предоставлена возможность фильтрации журнала операций по следующим критериям отбора:

- диапазон дат;
- категории операций;
- виды операций;
- сотрудник, выполнивший операцию;
- оборудование, над которым выполнена операция.

Журнал операций может вызываться как в режиме «общий», т.е. для всех элементов, так и для конкретного элемента. Просмотр журнала операций, возможность формировать его в соответствии с критериями отбора, формирование итоговых значений должно стать главным инструментом аналитики при анализе выполненных операций.

Журнал операций является инструментом просмотра истории выполненных над элементом операций.

Журнал операций предоставляет информацию о текущем состоянии элемента оборудования: если для элемента в журнале операций присутствуют записи с незавершенным статусом – значит он находится в стадии выполнения какой-либо операции (или нескольких операций), характер которых определяется содержанием записей в журнале операций. Если таких записей нет – значит никаких операций над элементом оборудования не производится.

Таким образом, основными слабыми местами в процессе выполнения задачи учета компьютерной техники являются затраты времени на

выполнение процесса учета компьютерной техники, высокая стоимость выполнения этого процесса и наличие ошибок, которые могут негативно влиять на процесс принятия решения, а также требовать дополнительного времени для поиска и устранения.

## **1.4 Анализ существующих разработок и обоснование выбора технологии проектирования**

### **1.4.1 Определение критериев анализа**

Определение критериев выбора автоматизированной информационной системы учета компьютерной техники заранее неопределимо при рационализации процесса и увеличении вероятности того, что выбранная АИС поможет компании достичь целей технического обслуживания. Многие факторы влияют на приоритеты обслуживания компании, такие как размер операций, количество технических специалистов, типы техники и т. д. Не существует универсального подхода для выбора подобной АИС, но следует рассмотреть пять ключевых критериев:

- Простота использования;
- Возможность мобильной АИС;
- Составление отчетов;
- Масштабируемость;
- Поддержка и услуги.

Это не единственные критерии, по которым следует оценивать рассматриваемое ПО, но это области, которые оказывают значительное влияние на долгосрочность использования информационной системы.

#### **1. Простота использования.**

Чем сложнее АИС, тем больше вероятность отказа. Сложная система не обязательно та, которая обладает большей функциональностью, это скорее относится к тому, как функциональность представлена в рабочем процессе системы – переход между окнами, интуитивно понятный интерфейс и др.

Чтобы тщательно оценить удобство использования, все пользователи должны быть частью процесса. Если технические специалисты сочтут систему трудной в использовании, они не сразу примут ее, что снижает ценность такой АИС. Следует привлекать техников и другие ключевые заинтересованные стороны к рассмотрению пользовательского интерфейса, рабочего процесса и других ключевых функций, чтобы убедиться, что они включены в систему.

## 2. Мобильные возможности.

Мобильные функции в АИС достаточно важны, так как многие технические специалисты используют мобильное устройство для работы. Мощность, которую обеспечивает мобильная АИС, означает, что при работе на оборудовании в полевых условиях технический специалист может эффективно выполнять необходимые задачи для технического обслуживания и ремонта:

- Работа с документацией;
- Оценка истории обслуживания оборудования;
- Проверка наличия запчастей;
- Инициирование заказов на запчасти;
- Обновление статуса рабочего заказа.

Повышение эффективности труда в сочетании с более точными показателями производительности обслуживания - сильная комбинация, которая повышает общую ценность АИС.

## 3. Отчетность.

Хорошая АИС не только помогает компаниям управлять своими программами технического обслуживания, но также служит хранилищем исторической информации об активах и заказах на работы по техническому обслуживанию.

Эффективные отчеты предоставляют исчерпывающую библиотеку отчетов, которые могут стать хорошей отправной точкой. Однако надежная отчетность также должна предоставлять компаниям возможность гибко

настраивать отчеты в соответствии с конкретными потребностями. Отчеты — это способ повысить ценность АИС, позволяя принимать более обоснованные бизнес-решения.

#### 4. Масштабируемость.

Для оценки необходимо учитывать два основных аспекта масштабируемости. Во-первых, насколько система подходит компании. Во-вторых, масштабируемость — это способность системы расти в соответствии с потребностями организации.

#### 5. Поддержка и услуги.

Хорошие поставщики АИС являются партнерами для своих клиентов по внедрению, обучению и постоянной поддержке. Услуги по внедрению являются естественной частью проекта. Компании должны иметь возможность запросить у поставщиков четкий план внедрения с ожидаемыми сроками.

Обучение может принимать разные формы. Любое предлагаемое формальное обучение должно быть четко изложено поставщиком и должно определять продолжительность, количество пользователей, которые могут быть включены в сеансы, материалы, которые будут предоставлены, и т. д.

После установки АИС могут возникнуть вопросы, и работа команды поддержки поставщика по оказанию помощи будет иметь решающее значение в решении возникающих проблем.

### **1.4.2 Сравнительная характеристика существующих разработок**

Рынок программного обеспечения по учету компьютерной техники представлен весьма большим количеством программ различных фирм. Рассмотрим программы, наиболее подходящие для решения поставленных задач.

Учет оборудования версия Prof для 1c 8.1 и 8.2

Программа (рисунок 7) реализована фирмой TuneSoft на платформе 1c 8.2.

Вся работа в программе идет в нескольких главных формах, в которых можно будет просматривать основные данные, формировать отчеты и вводить новые документы.

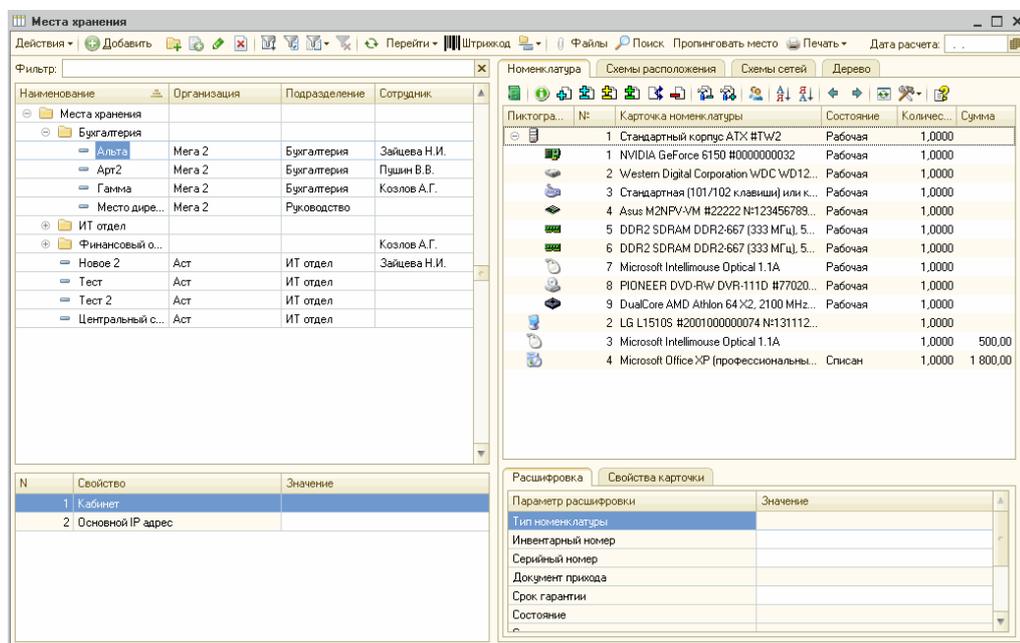


Рисунок 7 – Учет оборудования версия Prof

В журнале учета комплектующих, можно осуществить все операции с комплектующими и расходными материалами, кроме того, осуществить отбор по основным параметрам документов: Рабочим местам, комплектам, комплектующим.

В программе Учет оборудования можно учитывать:

- Учет оборудования, ПО и прочих материальных и нематериальных ценностей. Ведение учета по инвентарным, серийным номерам, номерам лицензий. Возможность ввода штрих-кодов, состояний, материально-ответственных лиц для оборудования;
- Учет заявок поставщикам. Возможность создавать заявки ИТ (заявки на последующую закупку), организовать тендеры, создавать

счета поставщиков. Возможность закрытия заявки частично или полностью;

- Учет заявок пользователей. Возможность с помощью графических схем создавать любой маршрут движения заявки. Возможность добавления собственных дополнительных свойств для заявок. Web интерфейс для ввода и изменения заявок пользователей;
- Оборудование в ремонте. Учет как внутреннего так и внешнего ремонта. Ремонт на месте. Возможность замены и списания документом ремонта комплектующих. Отчеты по затратам на ремонт, отчеты оборудованию в ремонте;
- Импорт данных из WMI и Эвереста Сканирование с помощью WMI, импорт из файлов отчета Эвереста. Возможность ввода синонимов для оборудования. Автоматическая инвентаризация. Импорт свойств оборудования (диагонали, объема памяти, чипсета, скорости и прочего);
- Штрих-кодирование. Возможность ввода данных с помощью сканера штрих-кода, а так же терминала сбора данных;
- Учет ответственных лиц. Учет ответственных лиц по местам хранения с помощью документов;
- Разделение прав пользователей. Разделение прав как по таблицам так и по конкретным записям, таким как: местам хранения, свойствам, паролям, заявкам. Возможность закрытия периода для редактирования документов;
- Бюджетирование. Возможность бюджетирование расходов на любой период в разрезе статей расходов. Возможность вести бюджет как по документам поступления оборудования (услуг) так и по документам списания денежных средств. Возможность вести бюджет по счетам поставщиков.

Стоимость конфигурация ИТ управление, версия 2014.x для 1С:Предприятие 8 версия на 5 пользователей – 12900 рублей.

Стоимость 12 месяцев поддержки и поставки обновлений – 3900 рублей.

Одинескин: Учёт оборудования 2.0.

Программа (рисунок 8) реализована на платформе 1с 8.2.

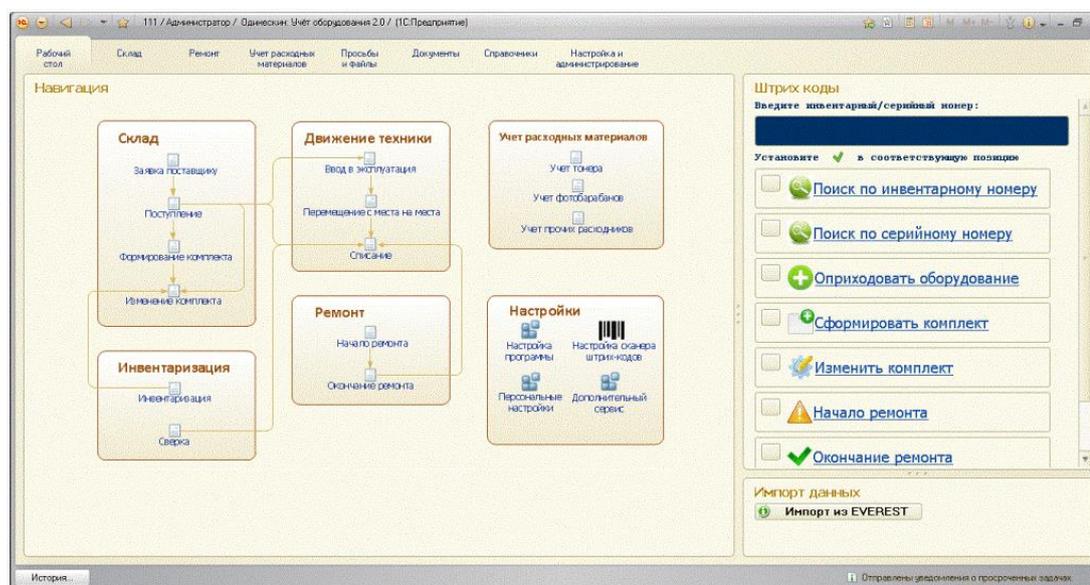


Рисунок 8 – Одинескин: Учёт оборудования 2.0

Решение «Одинэскин:Учёт оборудования 2.0» позволяет осуществлять автоматизацию учета оборудования, компьютеров, компьютерной техники и любых других материальных ценностей на предприятии.

Конфигурация позволяет:

- Организовать систему подотчёта на предприятии и учет материалов;
- Вести учёт компьютеров, любой техники, мебели, программного обеспечения с использованием технологии штрихкодирования. В том числе с помощью программ-сборщиков данных;
- Вести учёт заявок пользователей и получать отчётность по выполненным заявкам, времени их выполнения и т.д.

Позволяет вести упрощенный документооборот на предприятии, учитывать заявки пользователей, контролировать задания работникам.

- Вести учёт заявок поставщикам оборудования. Возможность отслеживать выполнение заявок. Встроенный почтовый клиент, позволяет отправлять заявки напрямую из 1С;
- Получать отчётность в количественном и суммовом выражениях. Получение остатков, оборотов по любым отборам и группировкам данных. Большое количество печатных форм для документов и элементов справочников;
- Возможность отслеживать оборудование, находящиеся в ремонте. Возможность отслеживать время в ремонте. Исполнителей ремонта: гарантия, внутренний ремонт и т.д.;

Стоимость ПО – 11800 рублей.

PrintStore — учет техники и расходных материалов.

Программа PrintStore представляет собой учетный инструмент для учета компьютерной техники, а также оборудования и расходных материалов, которым достаточно легко и просто пользоваться. Распределенный характер системы позволяет отслеживать наличие расходных материалов в каждом устройстве, подключенном к системе. Система может автоматически определять момент, когда необходимо заменять расходные материалы, и высылать письмо либо каким-либо другим способом предупреждать о необходимости их замены. Программа поддерживает работу как с картриджами производителей, так и с заменяемыми расходными материалами. В системе по умолчанию уже содержится большое количество внесенной техники различных производителей, что означает что при внедрении системы не появится необходимости заносить их вручную. Если какого-либо принтера или другого устройства нет – его нетрудно внести в систему.

Программа позволяет вести учет не только оборудования, но например серверных шкафов, мебели, и остального сопутствующего оборудования. При этом ведется учет перемещений техники и мебели, манипуляции,

которые были с нею произведены, проводится учет смены ответственных за технику лиц.

Возможности PrintStore:

Учет оборудования: приход, перемещение, ремонты, списание.

Учет картриджей: приход, перемещение, расход на оборудование, списание, отображение оперативных остатков.

Программа позволяет рассчитывать то, как быстро происходит расход расходных материалов, таких например как тонер. Это позволяет проводить планирование замены расходных материалов в устройствах, а также контролировать и оптимизировать количество расходных материалов на складах предприятия.

Также программа позволяет вести учет количества отпечатанных копий, и соотносить эту информацию с операциями замены расходных материалов в принтерах и МФУ, а также учитывать необходимость ремонта.

Стоимость ПО – 67000 рублей за 5 лицензий.

Проведем сравнение рассматриваемых и разрабатываемой системы в виде сравнительной таблицы.

Таблица 1 – Сравнительный анализ систем учета оборудования

	Учет оборудования версия Prof	Одинескин: Учёт оборудования 2.0.	PrintStore
Ведение дерева оборудования	-	-	-
Учет операций	-	+	+
Журнал работ	+	+	+
Документы перемещения	+	+	+
Документы списания	-	+	+
Возможность работы в сети	+	+	+

Рассмотрев документацию по имеющемуся на рынке программному обеспечению по учету компьютерной техники, ознакомившись с демо-

версиями ряда программ и получив отзывы об имеющихся программах от пользователей, можно прийти к следующим выводам:

- Имеющиеся программные комплексы не могут решать все поставленные задачи;
- Приобретение программных комплексов потребует значительных финансовых затрат;
- Техническая сложность и высокая цена обновления программ;
- Модификация и доработка имеющихся программ под нужды предприятия своими силами практически невозможна;
- Пользовательский интерфейс многих программ очень неудобен, ряд программ вообще не имеет графического интерфейса;
- Настройка программ сложна и часто неудобна;
- Выходные формы отчетов многих программ не полностью устраивают пользователей;
- На региональном уровне у всех фирм-разработчиков отсутствует прямая персональная поддержка пользователей программ.

Исходя из вышеперечисленного и из сведений сравнительной таблицы – ни одна из программ не предоставляет всех возможностей, которые описаны в техническом задании на разработку. Поэтому можно сделать вывод о том, что разработка собственного решения является лучшим выходом для нашего предприятия.

В итоге, можно сделать вывод о том, что использование информационной системы учета компьютерной техники и комплектующих организации поднимает деятельность отдела ИТ на качественно новый уровень и имеет положительный экономический эффект. Также внедрение подобной информационной системы способно ликвидировать основные слабые места в процессе выполнения задачи учета оборудования на предприятии.

## 1.5 Требования к разрабатываемой системе

Требования к разрабатываемой системе следующие:

- Возможность ввода данных о местах нахождения техники;
- Возможность ввода типов техники;
- Возможность ввода данных о технике;
- Возможность ввода видов работ;
- Возможность регистрации работ в журнале работ;
- Возможность формирования документов перемещения;
- Возможность формирования документов списания;
- Графическое отображение дерева оборудования и кабинетов организации;
- Возможность формирования аналитических отчетов.
- Разграничение прав доступа пользователей к системе.

Среда реализации: Delphi XE8.

Используемая СУБД: MS SQL Server 2012.

### **Вывод по главе**

В первой главе дана характеристика предприятия, краткая характеристика исследуемого подразделения, описана задача автоматизации.

Также было проведено концептуальное моделирование предметной области задачи и подробная постановка задачи. Подробно описан процесс учета оборудования.

В заключение проанализировано несколько программных средств по теме исследования и сделан вывод о том, что для исследуемого предприятия требуется разработка собственной системы.

## **Глава 2 Разработка и реализация проектных решений**

### **2.1 Логическое моделирование предметной области**

#### **2.1.1 Логическая модель и ее описание**

В состав большинства информационных систем входят:

- База данных (класс);
- Объект связи (метод);
- Интерфейс информационной системы (интерфейс класса).

Файл данных располагается либо на локальном компьютере, либо на сервере. Файл предназначен для хранения логической структуры базы данных информационной системы. В структуру данных входит всё содержимое базы данных, в частности запросы, представления, хранимые процедуры, триггеры, и т.д.

Объект связи – это объект языка программирования, осуществляющий связь между файлом данных и интерфейсом информационной системы; Интерфейс информационной системы служит для взаимодействия базы данных информационной системы с её пользователем. Он может находиться как на клиентском компьютере, так и на сервере.

Класс – это множество объектов с общей структурой и поведением. Метод – это операция над объектом, определённая как часть описания класса.

Интерфейс класса – это внешний вид класса, объекта или модуля, выделяющий его существенные черты.

Архитектура типовой информационной системы приведена на рисунке 9.

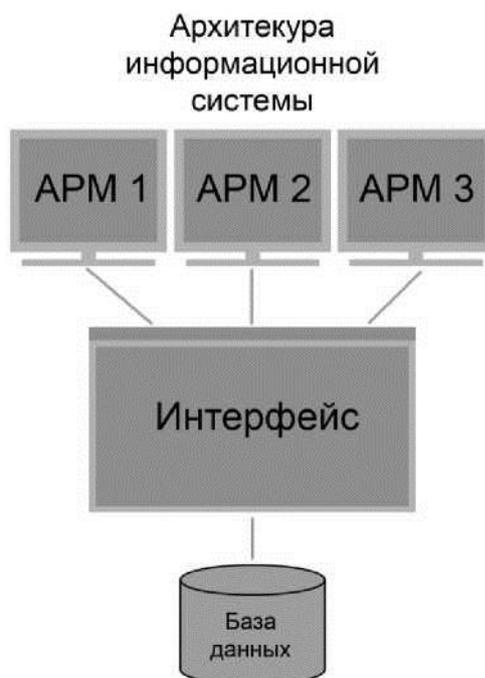


Рисунок 9 – Архитектура информационной системы

В качестве классификаторов в системе используются автоинкрементные поля целочисленного типа.

Входной информацией в системе являются характеристики оборудования, а также данные по местам расположения оборудования.

### **2.1.2 Характеристика базы данных**

Проведем концептуальное проектирование базы данных, описав основные сущности, которые должны в ней присутствовать, а также связи между ними.

С базой данных работают сотрудники. Каждый сотрудник (пользователь), обладает определенной ролью, от которой зависят его права доступа к системе.

Пользователи выполняют операции над оборудованием. Все оборудование привязано к определенным объектам в системе. Иерархия объектов определяет структуру предприятия. На верхнем уровне иерархии находится управление, ниже располагаются отделы, еще ниже – структурные единицы - кабинеты.

Все оборудование относится к определенным видам оборудования (например ноутбук, компьютер, принтер, сканер и т.д.).

По оборудованию ведется журнал работ. Каждая работа в журнале относится к определенному виду работ (например – техническое обслуживание).

На оборудование оформляются также документы перемещения и документы списания, которые заносятся в соответствующие журналы.

Таким образом, в базе данных должны быть реализованы следующие сущности:

- Сотрудники,
- Роли,
- Должности сотрудников,
- Оборудование,
- Виды оборудования,
- Журнал работ,
- Виды работ,
- Документы списания,
- Документы перемещения.

Ниже, на рисунке 10, показана концептуальная модель базы данных.

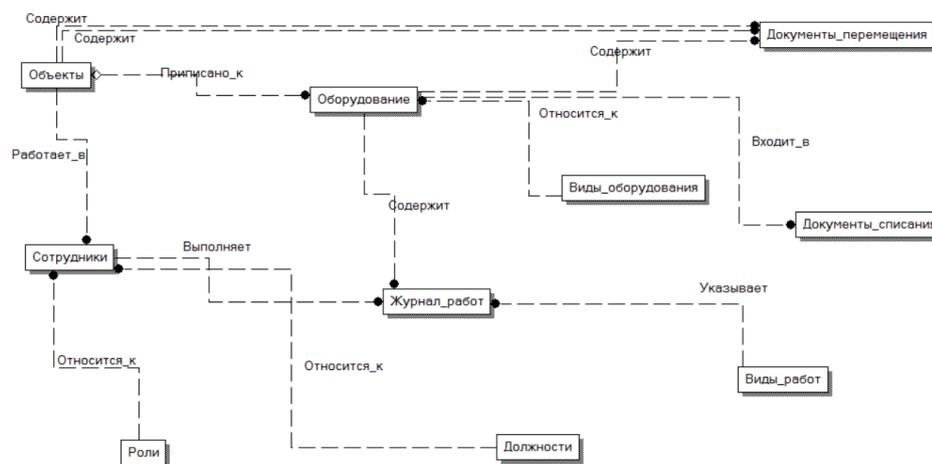


Рисунок 10 – Концептуальная модель базы данных

После концептуального проектирования приступаем к логическому проектированию БД. На этом этапе к сущностям добавляются атрибуты.

На рисунке 11 показана логическая модель базы данных.

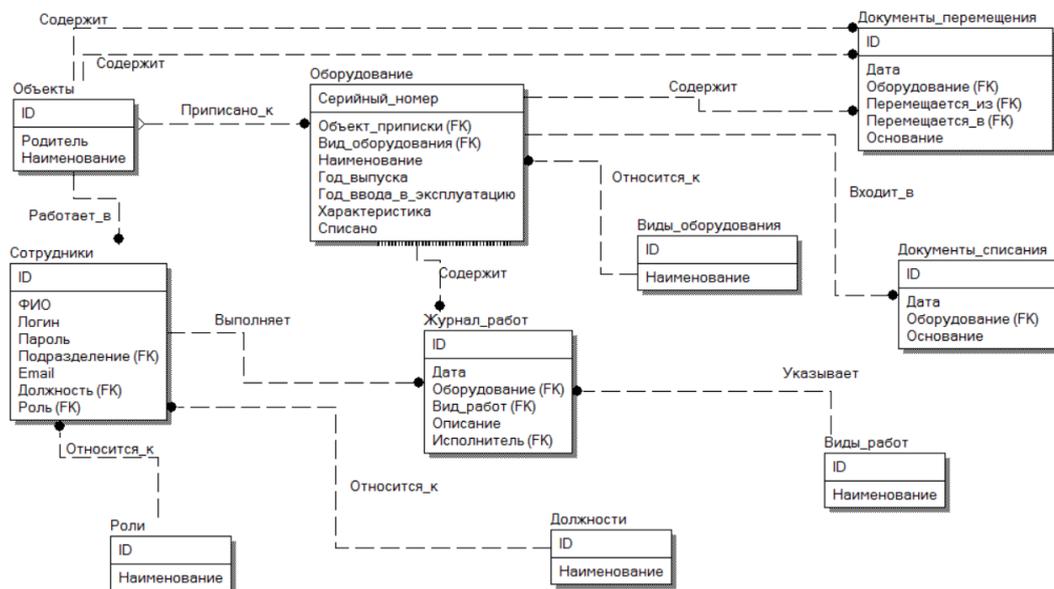


Рисунок 11 – Логическая модель базы данных

Для того, чтобы реализовать базу данных в терминах выбранной СУБД – необходимо присвоить всем атрибутам сущностей определенные типы данных.

Укажем типы данных для каждого из атрибутов сущностей (табл 2-11).

Таблица 2 «Оборудование» служит для хранения информации о ремонтируемом оборудовании. Структура таблицы «Оборудование» следующая.

Таблица 2 – Структура таблицы «Оборудование»

Название поля	Тип данных	Ключ
Серийный номер	Varchar(32)	Первичный ключ
Объект приписки	Integer	Внешний ключ
Вид оборудования	Integer	Внешний ключ
Наименование	varchar(50)	
Год выпуска	Integer	
Год ввода в эксплуатацию	Integer	
Характеристика	Varchar(500)	
Списано	Bit	

Таблица 3 – «Виды оборудования» содержит информацию о доступных видах оборудования.

Таблица 3 – Структура таблицы «Виды оборудования»

Название поля	Тип данных	Ключ
Id	Integer	Первичный ключ
Наименование	Varchar(150)	

Таблица «Сотрудники» содержит информацию о сотрудниках предприятия, с указанием их учетных данных для входа в систему.

Таблица 4 – Структура таблицы «Сотрудники»

Название поля	Тип данных	Ключ
Id	Integer	Первичный ключ
ФИО	Varchar(150)	
Логин	Varchar(32)	
Пароль	Varchar(32)	
Подразделение	Integer	Внешний ключ
Email	Varchar(50)	
Должность	Integer	Внешний ключ
Роль	Integer	Внешний ключ

Таблица 5 – «Должности» содержит информацию о должностях сотрудников.

Таблица 5 – Структура таблицы «Должности»

Название поля	Тип данных	Ключ
Id	Integer	Первичный ключ
Наименование	Varchar(150)	

Таблица 6 – «Роли» содержит информацию о ролях в системе. От роли зависит уровень доступа.

Таблица 6 – Структура таблицы «Роли»

Название поля	Тип данных	Ключ
Id	Integer	Первичный ключ
Наименование	Varchar(150)	

Таблица 7 – «Журнал работ» содержит информацию о работах, проведенных над оборудованием. Таблица содержит следующие поля.

Таблица 7 – Структура таблицы «Журнал работ»

Название поля	Тип данных	Ключ
Id	Integer	Первичный ключ
Дата	Datetime	
Оборудование	Integer	Внешний ключ
Вид работ	Integer	Внешний ключ
Описание	Varchar(250)	
Исполнитель	Integer	Внешний ключ

Таблица 8 – «Виды работ» содержит информацию о видах работ над оборудованием.

Таблица 8 – Структура таблицы «Виды работ»

Название поля	Тип данных	Ключ
Id	Integer	Первичный ключ
Наименование	Varchar(150)	

Таблица 9 – «Документы перемещения» содержит информацию об операциях перемещения, проводившихся с оборудованием.

Таблица 9 – Структура таблицы «Документы перемещения»

Название поля	Тип данных	Ключ
Id	Integer	Первичный ключ
Дата	Datetime	
Оборудование	Integer	Внешний ключ
Перемещается из	Integer	Внешний ключ
Перемещается в	Integer	Внешний ключ
Основание	Varchar(250)	

Таблица 10 – «Документы списания» содержит информацию об операциях списания, проводившихся над оборудованием. Таблица содержит следующие поля.

Таблица 10 – Структура таблицы «Документы списания»

Название поля	Тип данных	Ключ
Id	Integer	Первичный ключ
Оборудование	Integer	Внешний ключ
Основание	Varchar(250)	

На рисунке 12 представлена физическая схема базы данных.

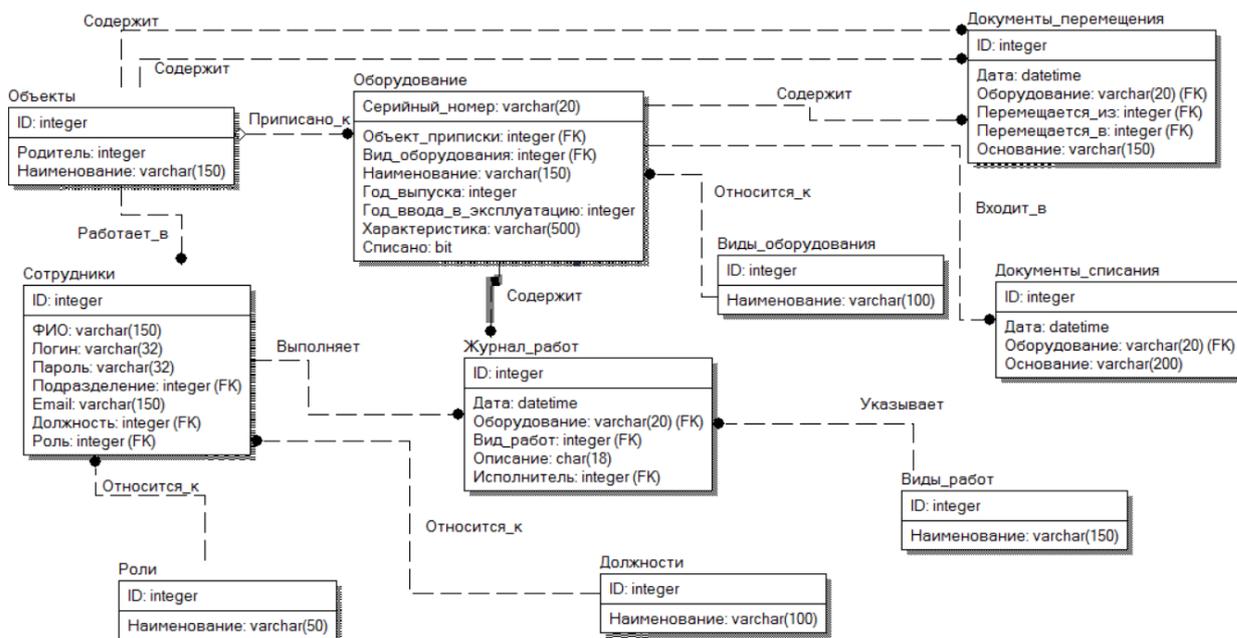


Рисунок 12 – Физическая схема базы данных

С помощью программного средства ErWin нами был сгенерирован код создания базы данных, который приведен в Приложении А.

Выходной информацией в системе являются различные отчеты, в частности отчет по перемещению оборудования.

## 2.2 Физическое моделирование АИС

### 2.2.1 Выбор архитектуры АИС

В качестве архитектуры сети будем использовать клиент-серверную архитектуру. Это вычислительная или сетевая архитектура, в которой сетевая нагрузка распределена между поставщиками услуг, называемыми серверами и заказчиками услуг, называемыми клиентами. Нередко клиенты и серверы взаимодействуют через компьютерную сеть и могут быть как различными физическими устройствами, так и программным обеспечением.

Преимущества этой архитектуры:

- отсутствие дублирования кода программы-сервера программами-клиента

- так как все вычисления выполняются на сервере, то требования к компьютерам, на которых установлен клиент, снижаются.
- все данные хранятся на сервере, который защищен гораздо лучше большинства клиентов. На сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.

Протокол используемый в сети - TCP/IP. Это набор сетевых протоколов передачи данных, используемых в сетях, включая сеть Интернет. Название TCP/IP происходит из двух важнейших протоколов семейства Transmission Control Protocol и Internet Protocol, которые были разработаны и описаны первыми в данном стандарте.

TCP протокол базируется на IP для доставки пакетов, но добавляет две важные вещи:

- установление соединения — это позволяет ему, в отличие от IP, гарантировать доставку пакетов;
- порты — для обмена пакетами между приложениями, а не просто узлами.
- Протокол TCP предназначен для обмена данными — это «надежный» протокол, потому что:
  - обеспечивает надежную доставку данных, так как предусматривает установления логического соединения;
  - нумерует пакеты и подтверждает их прием квитанцией, а в случае потери организует повторную передачу;
  - делит передаваемый поток байтов на части — сегменты - и передает их нижнему уровню, на приемной стороне снова собирает их в непрерывный поток байтов.

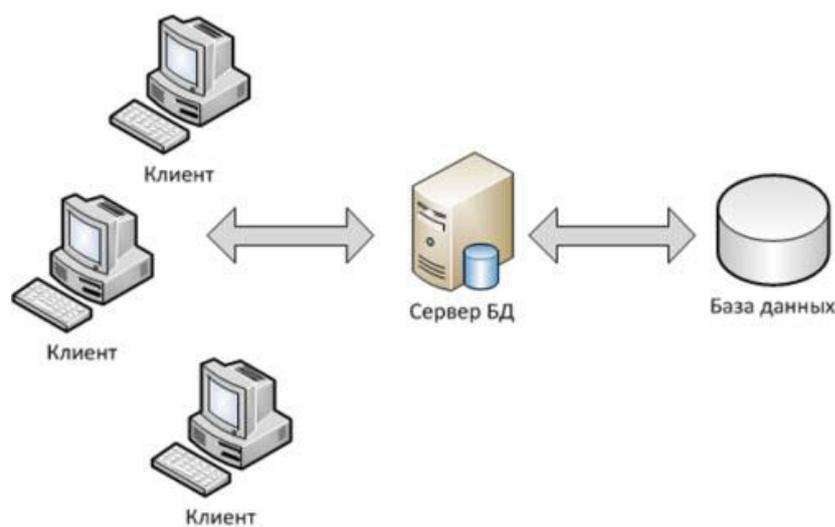


Рисунок 13 – Архитектура «клиент-сервер»

Клиент-серверная архитектура является наиболее приемлемым вариантом для нашей разработки, поскольку поддерживает многопользовательский доступ к базе данных, а также обеспечивает надежную защиту и разграничение доступа пользователей к информационной системе.

### **2.2.2 Структурная схема проекта**

Функциональная схема информационной системы приведена на рисунках ниже.

Функциональная схема работы АИС представлена на рисунке 14.

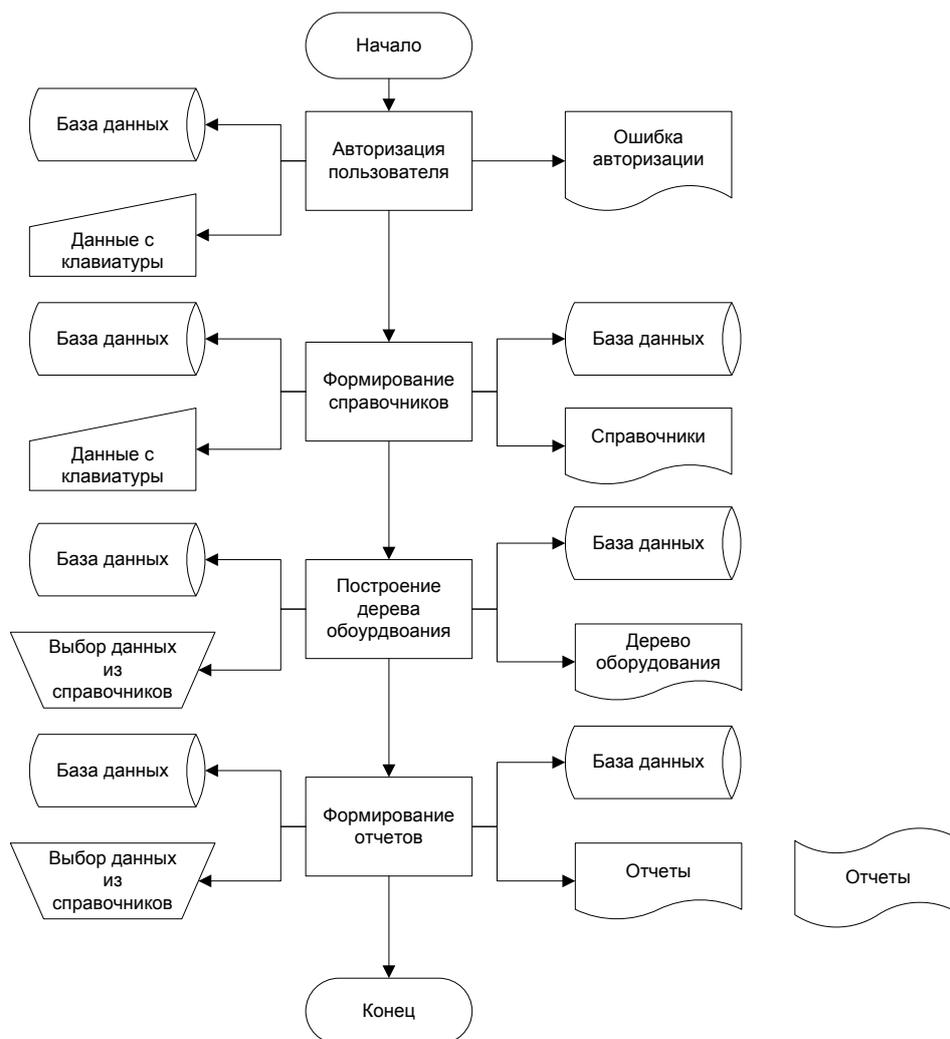


Рисунок 14 – Функциональная схема АИС

Согласно функциональной схеме – на основании имеющихся в БД данных формируется список структуры оборудования, после чего формируются и выводятся на печать отчеты.

### 2.2.3 Описание программных модулей

Одной из основных процедур в приложении является процедура построения дерева объектов LoadTree.

Данная процедура предназначена для формирования дерева всего оборудования и всех объектов учета в левой части формы. Для построения данного дерева используется стандартный компонент DelphiTreeView.B

качестве данных построения дерева используются таблицы «Объекты» и «Оборудование».

На рисунках 15 и 16 приведен алгоритм функционирования данной процедуры.

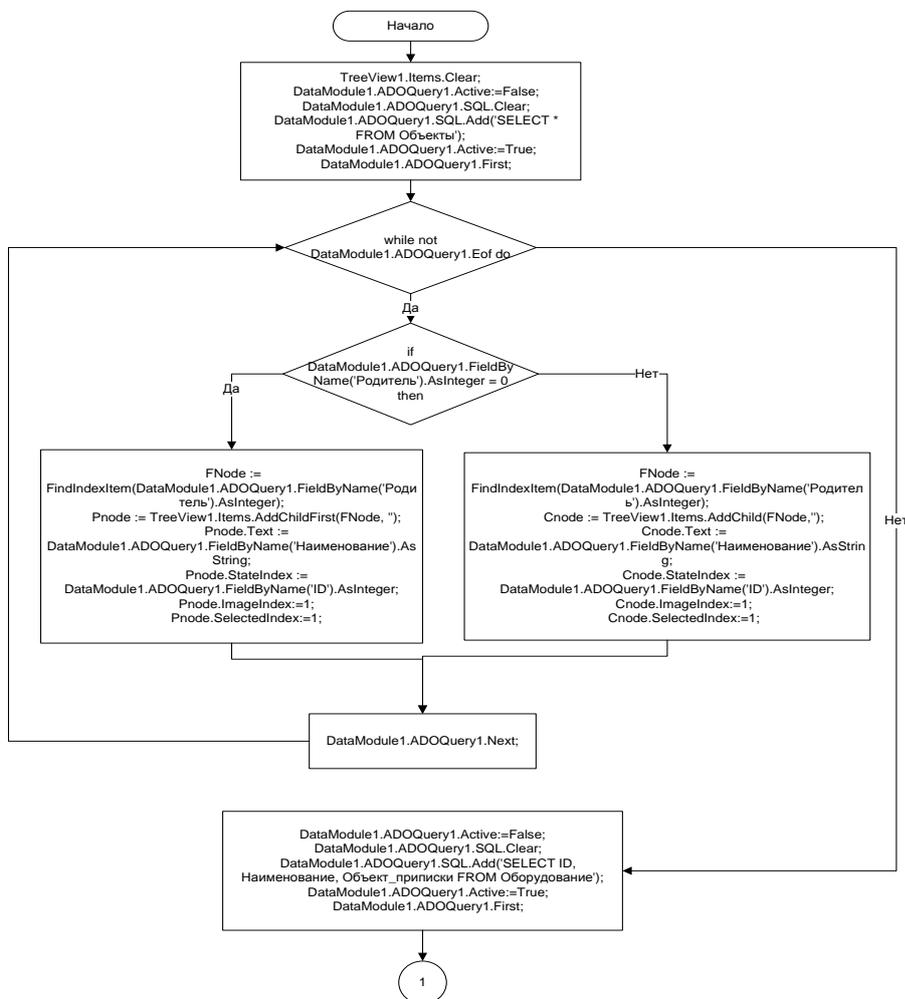


Рисунок 15 – Блок-схема работы процедуры LoadTree

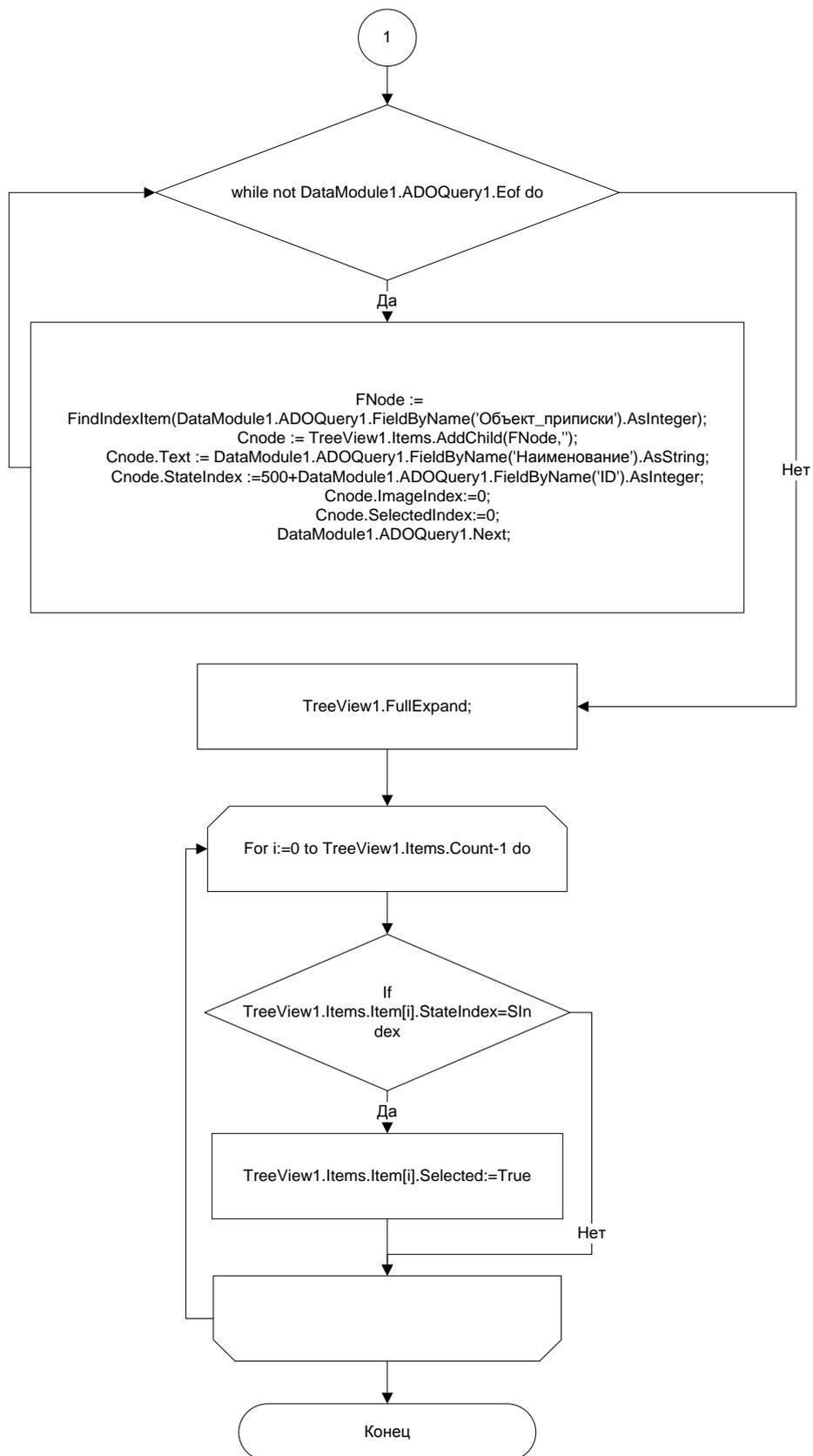


Рисунок 16 – Продолжение блок-схемы работы процедуры LoadTree

Опишем работу данной процедуры словесно.

В начале работы процедуры происходит очистка списка на случай, если там уже есть данные. После этого в запрос выбираются все данные из таблицы «Объекты».

Далее последовательно происходит перебор всех строк полученной таблицы и построение дерева. Для этого проверяется, не равно ли значение «Родитель» в строке таблицы нулю. Если это так, то в таблицу добавляется корневой элемент и в его свойства заносятся все необходимые данные.

Если значение поля «Родитель» не равно единице, то к объекту, указанному в качестве родительского, добавляется потомок. Этому потомку присваиваются все необходимые атрибуты.

Для однозначной идентификации какой объект добавляется, используется свойство узла дерева StateIndex. Если значение этого поля больше 500, то это оборудование, если меньше – то объект. Соответственно при добавлении оборудования значение этого поля вычисляется как 500 прибавить IDоборудования. Такое разделение позволяет идентифицировать объекты и оборудование.

После того, как построено дерево объектов, начинается добавление в дерево оборудования.

Для этого из таблицы «Оборудование» в запрос выбираются поля «ID», «Объект приписки» и «Наименование». После этого происходит последовательный перебор всех строк полученной таблицы. К родительским узлам дерева добавляется текущее оборудование и ему присваиваются необходимые атрибуты.

После этого последовательно перебираются все элементы дерева и делается активным тот элемент, IDкоторого занесен в поле SIndex. Это обеспечивает выделение одного из элементов списка.

Таким образом происходит построение списка оборудования и объектов.

## 2.2.4 Схема взаимосвязи программных модулей и информационных файлов

Иерархическая схема модулей представлена на рисунке 17.

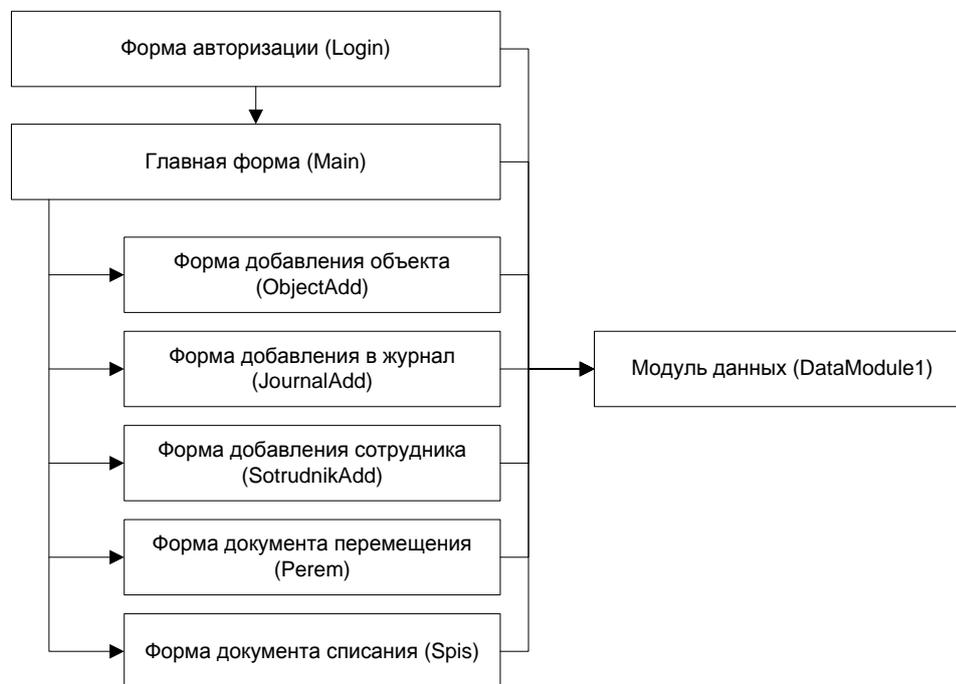


Рисунок 17 – Иерархическая схема модулей

За счет использования интерфейса MDI, как было указано выше – нам удалось достигнуть малого количества модулей в программе.

Ниже приведено описание всех модулей системы.

Модуль UMain.pas – главный модуль системы, в котором содержится процедура построения дерева объектов и оборудования, функции справочников, а также все остальные функции системы.

Модуль ULogin.pas – содержит процедуру авторизации пользователя в системе. В процессе работы определяется, существует ли сотрудник с указанным логином и паролем в базе данных. Если существует – на экран выводится главный модуль системы. Если не существует – выдается сообщение об ошибке. Также в процессе работы процедуры записываются права доступа сотрудника для дальнейшей работы приложения.

UDataModule.pas – стандартный модуль объектов доступа к базе данных Delphi. Подобные модули используются только для доступа к базе данных и не могут содержать никаких интерфейсных элементов. Данный модуль не выводится на экран в процессе работы программы и используется только для хранения объектов базы данных и их свойств, таких как таблицы, запросы и прочее.

USpis.pas – интерфейсный модуль документа списания.

UPerem.pas – интерфейсный модуль документа перемещения.

## **2.3 Технологическое обеспечение задачи**

### **2.3.1 Организация технологии сбора, передачи, обработки и выдачи информации**

Технологический процесс машинной обработки системы учета компьютерной техники представляет собой совокупность операций, осуществляемых в строго определенной последовательности с начального момента до окончательного получения заданных результатов. Его можно подразделить на четыре укрупненных этапа: первичный, подготовительный, основной и заключительный.

На первичном этапе работы осуществляется сбор исходных данных, их регистрация и передача для ввода в ЭВМ. Подготовительный этап охватывает операции по приему, контролю и регистрации входной информации и переносу ее на машинные носители. Основной этап обеспечивает контроль, выпуск и передача результатной информации пользователю.

В условиях диалоговой обработки внемашинная и внутримашинная технологии работы приложения тесно связаны друг с другом и не имеют четкой границы, также как и нет четкого разграничения подготовительного, основного и заключительного этапа технологического процесса. Это

происходит по причине того, что работа в диалоговом режиме и не имеет заранее определенной последовательности действий.

Отдельные части (блоки) схемы работы программы реализуют выполнение пунктов меню, соответствующих схеме диалога.

### 2.3.2 Схема технологического процесса сбора, передачи, обработки и выдачи информации

Укрупненная схема алгоритма построения прогноза продаж представлена на рисунке 18.

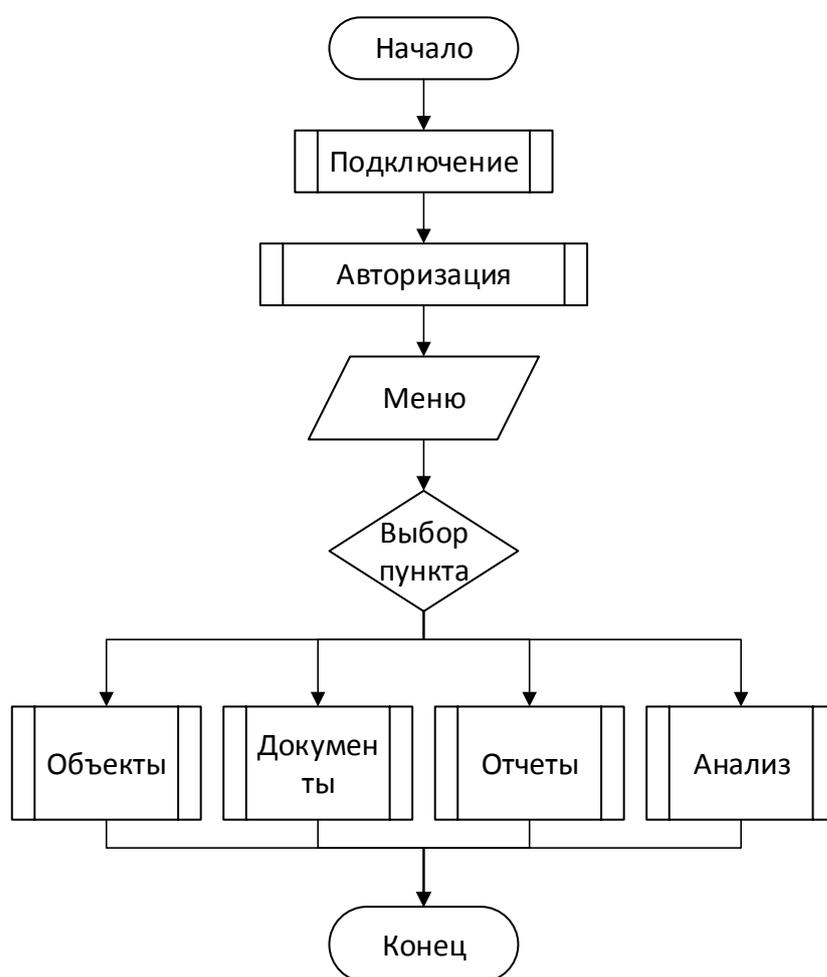


Рисунок 18 – Укрупнённая схема алгоритма АС

Согласно укрупненной схеме работы системы – система содержит несколько различных модулей.

В частности один модуль отвечает за объекты в описываемой системе (например кабинеты, единицы техники и т.д.)

Также в системе формируются отчеты, заполняются документы, происходит анализ на основе полученных данных.

## 2.4 Контрольный пример реализации проекта и его описание

В начале работы с программой происходит отображение формы входа. Здесь необходимо ввести логин и пароль пользователя (Рисунок 19).

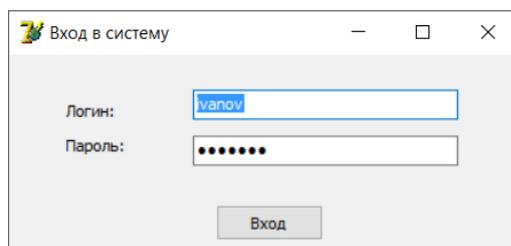
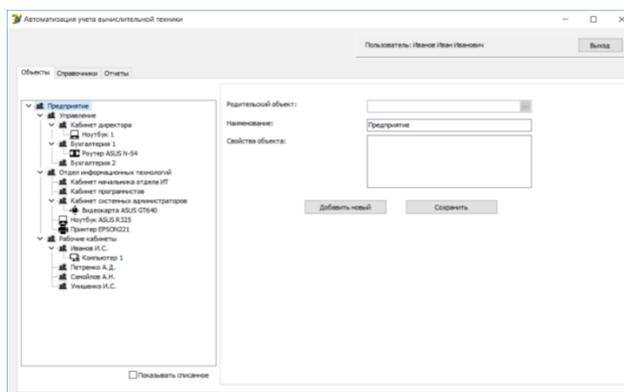


Рисунок 19 – Форма входа

Если данные введены правильно – происходит переход к главному окну.

Приложение разработано таким образом, чтобы в нем можно было как можно меньше форм. Для удобства пользователя – работа с программой в основном происходит в основном в одном окне.

Ниже показано главное окно программы (Рисунок 20).



## Рисунок 20 – Главное окно программы

В верхней части окна находится ФИО пользователя, вошедшего в систему, а также кнопка выхода из программы.

По центру слева находится дерево оборудования. Здесь представлена структура предприятия, а также все оборудование, которое находится в каждой структурной единице.

При выборе оборудования либо объекта, в зависимости от того, какой тип выбран – происходит динамическое изменение состава вкладок в правой части окна. Для объекта происходит отображение формы просмотра и редактирования данных об объекте. Для оборудования происходит отображение вкладки редактирования и добавления данных об оборудовании, а также дополнительных вкладок «Журнал работ», «Документы перемещения» и «Документы списания».

Закладка редактирования данных об оборудовании выглядит следующим образом (Рисунок 21).

The screenshot shows a software window with four tabs: 'Свойства оборудования' (Equipment Properties), 'Журнал работ' (Work Journal), 'Документы перемещения' (Movement Documents), and 'Документы списания' (Disposal Documents). The 'Свойства оборудования' tab is active. The form contains the following fields:

- Серийный номер: AD88821
- Инвентарный номер: AD231
- Наименование: Роутер ASUS N-54
- Объект приписки: Кабинет директора (dropdown menu)
- Год выпуска: 2016
- Год ввода в эксплуатацию: 2016
- Вид оборудования: Сетевое оборудование (dropdown menu)
- Характеристика: Двухдиапазонный

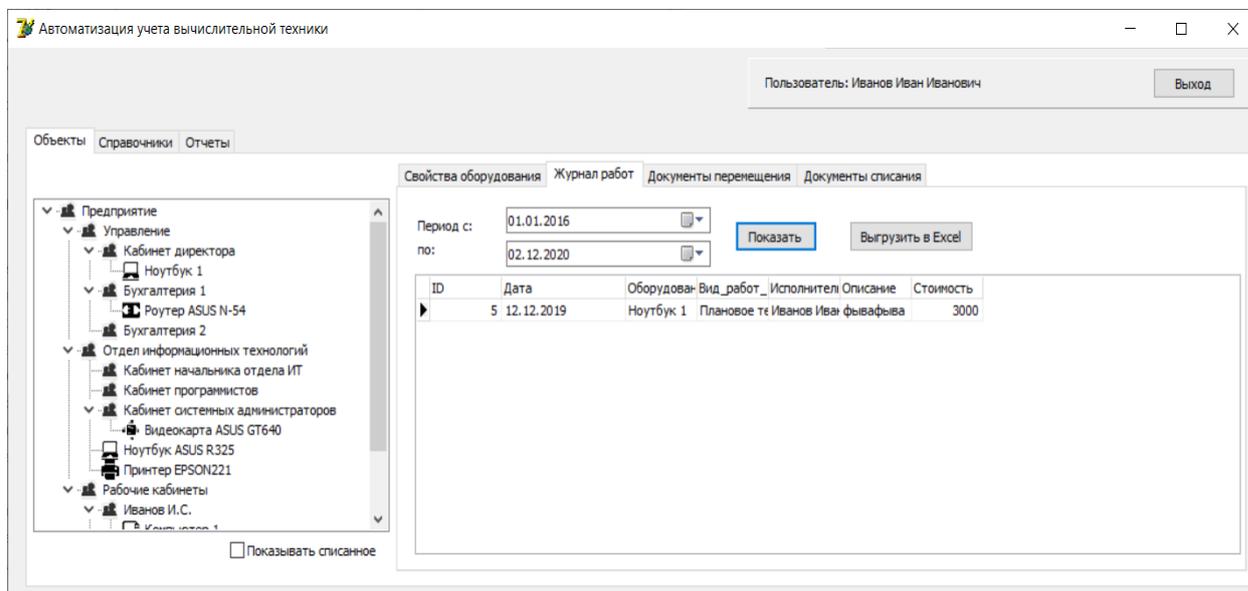
At the bottom of the form, there are two buttons: 'Добавить новое' (Add new) and 'Сохранить' (Save).

## Рисунок 21 – Редактирование данных об оборудовании

Для добавления новой единицы оборудования необходимо нажать на кнопку «Добавить новое» и ввести нужные данные.

Ниже показана закладка журнала работ. Для того, чтобы добавить новую работу в журнал необходимо нажать правой кнопкой на форме, после чего выбрать «Добавить» и ввести данные в появившуюся форму.

Для отбора данных за период по данной единице оборудования необходимо указать начальную и конечную дату и нажать на кнопку «Показать». (Рисунок 22).



## Рисунок 22 – Журнал работ по оборудованию

Для просмотра документов перемещения необходимо выбрать соответствующую закладку.

После выбора происходит отображение следующей формы (Рисунок 23).

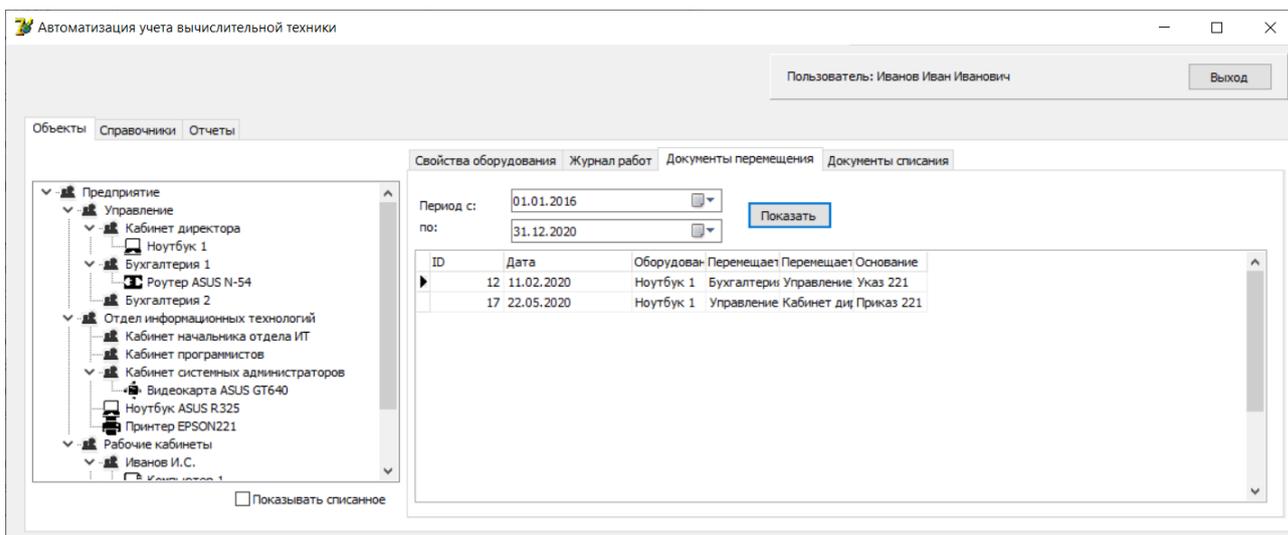


Рисунок 23 – Документы перемещения

В данной форме также доступно контекстное меню добавления нового документа.

Работа с документами списания происходит аналогичным образом.

В верхней части главной формы находятся закладки, с помощью которых можно перейти к справочникам, а также к отчетам в системе.

Журнал справочника сотрудников выглядит следующим образом (Рисунок 24).

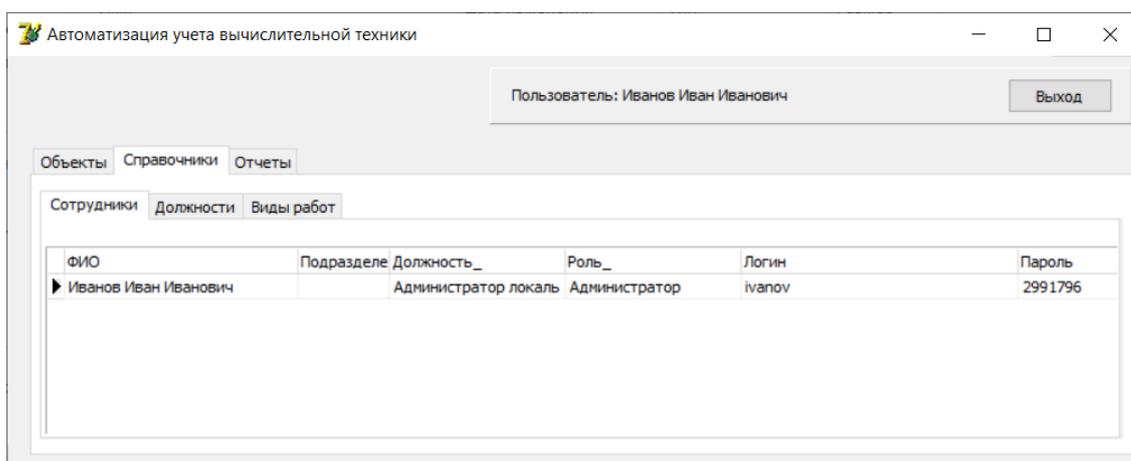


Рисунок 24 – Журнал справочника сотрудников

Для добавления нового сотрудника необходимо правой кнопкой мыши в окне, после чего выбрать «Добавить».

На экране отобразится форма добавления сотрудника (Рисунок 25).

Добавление/изменение сотрудника

ФИО: Иванов Иван Иванович

Подразделение: Рабочие кабинеты

Должность: Администратор локальной сети

Роль: Администратор

Логин: ivanov

Пароль: 2991796

Email: ivanov@mail.ru

OK

Рисунок 25 – Форма добавления сотрудника

Работа с остальными справочниками происходит похожим образом.

Отчеты в системе выглядят следующим образом (Рисунок 26).

Автоматизация учета вычислительной техники

Пользователь: Иванов Иван Иванович

Выход

Объекты | Справочники | **Отчеты**

Работы по сотруднику | Перемещения за период | Списание за период | Средняя стоимость по сотрудникам | Средняя стоимость по видам | Анализ

Сотрудник: Иванов Иван Иванович

Начальная дата: 01.01.2016

Конечная дата: 11.02.2020

Вывести

Экспорт в Excel

Дата	Оборудование	Исполнитель	Вид_работ	Описание
12.12.2019	Ноутбук ASI	Иванов Иван	Замена комп	Заменена на
12.12.2019	Ноутбук ASI	Иванов Иван	Замена комп	Заменен про
12.12.2019	Ноутбук 1	Иванов Иван	Плановое те	физафыва
12.02.2019	Роутер ASUS	Иванов Иван	Плановое те	Заменена си

Рисунок 26 – Отчет «Работы по сотруднику»

Отчеты в системе формируются динамически на основании введенных пользователем параметров. Формирование отчетов происходит с использованием запросов SQL и представлений.

### **Выводы по главе 2**

В процессе написания данной главы достигнуты следующие результаты.

Была описана логическая модель системы, используемые классификаторы и системы кодирования.

Подробно описана и реализована база данных информационной системы.

После этого проведены проектирование и разработка системы. Подробно описан интерфейс и выполняемые функции.

## **Глава 3 Оценка и обоснование экономической эффективности проекта**

### **3.1 Выбор и обоснование методики расчёта экономической эффективности**

Предметная область проекта – процесс учета оборудования имеет ряд недостатков, обусловленных ручной обработкой данных, отсутствием контроля результатных данных. В результате на обработку данных в информационной системе тратятся очень большие объемы ресурсов, в частности временных и трудовых. В настоящее время большинство рабочих мест на предприятии автоматизированы, но учет компьютерной техники ведется без широкого использования компьютерной техники и информационных систем. Учет ведется либо в устаревших системах вроде MS Excel, либо не ведется вообще.

В связи с этим возникает проблема больших затрат ресурсов на выполнение задач, в результате имеются ошибки и неточности, а также относительно высока длительность процесса учета компьютерной техники.

Так, в месяц работники выполняет обработку порядка 200 операций. На обработку этой информации тратится в среднем 24 часа. Т.е. в месяц на процесс учета оборудования сотрудникам затрачивается около 288 часов. При этом около 50% трудовых затрат (144 часа в месяц) считаются избыточными, что в стоимостном выражении с учетом тарифа оплаты труда сотрудника (85,23 руб./час) составит около 12 273 руб./мес.

На основе проведенного анализа, основной проблемой предметной области является избыточная трудоемкость процесса, которая оценивается как 144 час/мес или 12 273 руб/мес. Таким образом, можно определить основную цель проекта, как сокращение трудовых затрат на размещение оборудования на 50% на базе автоматизированных решений. В результате в качестве основного фактора экономической эффективности проекта

выделена экономия на оплату труда сотрудников. Ожидаемые результаты проекта в предметной области отражены в таблице 11.

Таблица 11 – Результаты проекта в предметной области

Факторы улучшения бизнес-процессов в результате проекта	Факторы экономической выгоды, получаемые в результате проекта	Количественное выражение
Снижение количества ошибок	Экономия на оплату труда работников в части размещения оборудования (экономия трудозатрат)	Не менее 12 000 руб./мес (144 час/мес)
Уменьшение времени на ввод данных		

На основе типовой методики разработано информационно-методическое обеспечение для оценки экономической эффективности проекта, которое включает следующие показатели.

К трудовым показателям относятся следующие:

1. Трудозатраты на размещение оборудования по базисному варианту (T0):

$$T0 = [\text{Средняя продолжительность по базисному варианту, час/начисл}] * [\text{Количество начислений в мес}]$$

(3.1)

2. Трудозатраты на размещение оборудования по проектному варианту (T1):

$$T1 = [\text{Средняя продолжительность по проектному варианту, час/начисл}] * [\text{Количество начислений премии}]$$

(3.2)

3. Абсолютное снижение трудовых затрат ( $\Delta T$ ) на размещение оборудования:

$$\Delta T = T0 - T1$$

(3.3)

при этом значение показателя должно быть не меньше 144 час/мес.

4. Коэффициент относительного снижения трудовых затрат (Kт):

$$K_T = \Delta T / T_0 * 100\% \quad (3.4)$$

5. Индекс снижения трудовых затрат или повышение производительности труда ( $Y_T$ ):

$$Y_T = T_0 / T_1 \quad (3.5)$$

К стоимостным показателям относятся: абсолютное снижение стоимостных затрат ( $\Delta C$ ), коэффициент относительного снижения стоимостных затрат ( $K_C$ ), индекс снижения стоимостных затрат ( $Y_C$ ), рассчитываемые аналогично.

1. Стоимостные затраты на размещение оборудования по базисному варианту ( $C_0$ ), руб/мес:

$$C_0 = C_c * T_0, \quad (3.6)$$

где  $C_c$  – тарифная ставка оплаты труда сотрудника отдела, руб./час.

2. Стоимостные затраты на размещение оборудования по базисному варианту ( $C_1$ ), руб/мес:

$$C_1 = C_c * T_1, \quad (3.7)$$

3. Абсолютное снижение стоимостных затрат ( $\Delta C$ ) на размещение оборудования, руб/мес:

$$\Delta C = C_0 - C_1 \quad (3.8)$$

при этом значение показателя должно быть не меньше 12000 руб/мес.

4. Коэффициент относительного снижения стоимостных затрат ( $K_C$ )

$$K_C = \Delta C / C_0 * 100\% \quad (3.9)$$

5. Индекс снижения стоимостных затрат ( $Y_C$ ):

$$Y_C = C_0 / C_1 \quad (3.10)$$

Для определения трудовых и стоимостных затрат в базовом и проектном вариантах используем данные таблицы 3.3.

Таблица 12 – Показатели эффективности проекта

Показатель	До проекта	После проекта
Средняя продолжительность операций по учету оборудования	24	4
Тарифная ставка оплаты труда сотрудника отдела, руб.	85,23	
Количество операций в месяц	12	

Представленные данные получены в результате анализа предметной области до внедрения информационной подсистемы и на основе результатов тестовой эксплуатации ИПС. Определим показатели изменения трудовых затрат на начисление дополнительной премии сотрудникам:

1. Определим общие трудовые затраты работников отдела по базисному варианту:

$$T_0 = 24 \times 12 = 288 \text{ чел.-часа}$$

2. Общие трудовые затраты отдела поддержки специальных программ по проектному варианту:

$$T_1 = 4 \times 12 = 48 \text{ чел.-часов}$$

3. Рассчитаем абсолютное снижение трудовых затрат на выполнение задачи (формула 3.3.):

$$\Delta T = 288 - 48 = 240 \text{ чел.-часов}$$

4. Коэффициент относительного снижения трудовых затрат (формула 3.4) составит:

$$K = 240 / 288 \times 100\% = 83,3\%$$

5. Индекс снижения трудовых затрат или повышение производительности труда (формула 3.5):

$$Y = 288 / 48 = 6$$

Определим показатели изменения стоимостных затрат:

1. Определим общие стоимостные затраты работников отдела по базисному варианту:

$$C_0 = 288 \times 85,23 = 24546,24 \text{ руб/мес}$$

2. Общестоймостные затраты работников по проектному варианту:

$$C_1 = 48 \times 85,23 = 4091,04 \text{ руб/мес}$$

3. Рассчитаем абсолютное снижение трудовых затрат на выполнение задачи:

$$\Delta C = 24546,24 - 4091,04 = 20455,2 \text{ руб/мес}$$

4. Коэффициент относительного снижения стоимостных затрат составит:

$$K_c = 20455,2 / 24546,24 \times 100\% = 83,3\%$$

5. Индекс снижения трудовых затрат или повышение производительности труда:

$$Y_c = 24546,24 / 4091,04 = 6$$

Таким образом, наблюдается явное снижение трудовых и стоимостных затрат на размещение оборудования (на 83,3%) при использовании предлагаемой информационной системы.

### 3.2 Расчет показателей экономической эффективности

К показателям экономического эффекта и экономической эффективности относятся:

1. Годовой экономический эффект ( $\mathcal{E}$ ):

$$\mathcal{E} = \Delta C - E_n * K_n, \quad (3.11)$$

где  $K_n$  – затраты на создание и внедрение проекта, руб,  $E_n$  – нормативный коэффициент эффективности единовременных затрат на создание ИП (0,15).

2. Срок окупаемости затрат на создание и внедрение проекта ( $T_{ок}$ ), рассчитываемый в годах:

$$T_{ок} = K_n / \Delta C. \quad (3.12)$$

При этом затраты на создание и внедрение проекта ( $K_n$ ) определяются:

Кп =

[Стоимость лицензионного ПО, руб.] +

[Время разработки и отладки программы, час] \*

[Тариф оплаты труда программиста, руб/час]

(3.13)

Срок окупаемости должен быть не более трех лет. Информационно-методическое обеспечение оценки экономической эффективности проекта в части критериев отражено в таблице 13.

Таблица 13 – Показатели экономической эффективности проекта

Показатели эффективности	Критерии эффективности
Экономия на оплату труда работников ( $\Delta C$ )	Не менее 12 000 руб./мес
Абсолютное снижение трудовых затрат работников ( $\Delta T$ )	Не менее 144 чел-час./мес

Продолжение таблицы 13

Годовой экономический эффект ( $\Xi$ )	Больше 0
Срок окупаемости затрат на создание и внедрение проекта (Ток)	Не более 3 лет

Представленные критерии оценки базируются на стоимостной оценке проблемной области и общеэкономической логике показателей.

Далее необходимо определить затраты на разработку информационной системы. К ним относятся:

- время разработки и отладки программы;
- затраты на операционную систему;
- заработная плата программиста в день.

Также необходимо выполнить расчет основных показателей на отладку и разработку программного средства. К данным показателям относятся время отладки и разработки. Это время умножается на тарифную ставку специалиста-программиста, при этом также необходимо учесть дополнительные затраты на приобретение программного и аппаратного обеспечения.

Стоит отметить, что организации имеется рабочее место, оснащенное операционной системой. Однако пришлось приобрести лицензионную версию программы.

Таблица 14 – Основные показатели для вычисления затрат на разработку и отладку программы

Показатель	Значение показателя
Время разработки и отладки программы, дн.	14
Часовая тарифная ставка программиста, руб.	105
Стоимость лицензионной версии программы SQLServer, руб	128 300

Основываясь на этих базовых показателях, мы можем рассчитать стоимость работы разработчика. Она определяется как время разработки, умноженное на зарплату в месяц:

$$K = 14 \times 8 \times 105 = 11\,760 \text{ руб.}$$

Общие затраты на создание проекта (проектирование и внедрение) составят:

$$K_{п} = 11\,760 + 128\,300 = 140\,060 \text{ руб.}$$

Экономический эффект рассчитываем по формуле 3.6:

$$\varepsilon = 12 \times 20\,455,2 - 0,15 \times 58\,060 = 224\,453,4 \text{ руб.}$$

Срок окупаемости единовременных затрат, рассчитываем по формуле 3.7:

$$T_{ок} = 140\,060 / 224\,453,4 = 0,62 \text{ года} = 7,5 \text{ месяцев}$$

Полученные показатели можно представить графически в виде диаграмм.

На рисунке 27 представлена диаграмма, показывающая изменение трудозатрат в связи с внедрением проекта.



Рисунок 27 – Трудозатраты до и после внедрения проекта

На рисунке 28 представлена диаграмма, показывающая изменение стоимостных затрат предприятия в связи с внедрением проекта.

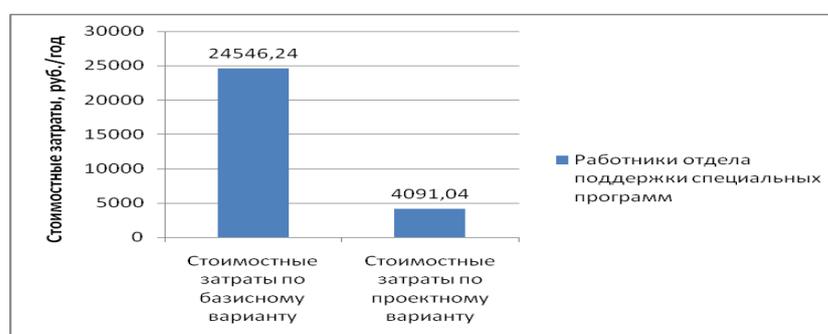


Рисунок 31 – Стоимостные затраты предприятия до и после внедрения проекта

Таким образом, все полученные показатели соответствуют норме и свидетельствуют об экономической выгоде проекта.

Показатели экономии на оплату труда работников по проектному варианту составят 20 455,2 руб./мес. По этим показателям проект полностью удовлетворяет поставленным перед ним требованиям (не менее 12 000 руб/мес).

Показатели трудозатрат также снизились, они составляют 48 чел-час/мес, снижение произошло на 240 чел-часов. Данные показатели также соответствуют поставленным критериям (не менее, чем на 144 чел-часа).

Снижение трудовых затрат персонала позволит сократить время внеурочной работы, снизить утомляемость, а значит повысить эффективность труда работников.

#### **Выводы по главе**

Экономический эффект от внедрения программного продукта составил 224 453,4 руб., что также удовлетворяет предъявленным требованиям (больше нуля). Срок окупаемости единовременных затрат составит 7,5 месяцев. Этот срок удовлетворяет предъявленным требованиям (не более трех лет).

## Заключение

В процессе работы над дипломным проектом были получены следующие результаты:

- Была дана общая характеристика предприятия и описана его организационная структура;
- Описана сущность задачи автоматизации предприятия, построены схемы работы «КАК ЕСТЬ» и «КАК БУДЕТ»;
- Проведена постановка задачи на разработку систему учета компьютерной техники;
- Проанализированы существующие разработки по теме исследования;
- Проведено концептуальное, логическое и физическое проектирование базы данных информационной системы;
- Выбрана архитектура ИС и разработано программное средство;
- Описана технология организации сбора, передачи и обработки информации в системе;
- Приведен контрольный пример реализации проекта;
- Обоснована экономическая эффективность разработки системы.

Разработанная АИС является средством достижения высокого уровня автоматизации на предприятии. Вследствие ее внедрения повышается производительность труда, снижается время, затрачиваемое на поиск нужной информации, сотрудники освобождаются от ряда продолжительных операций по вычислению и созданию информационных отчетов. При помощи данной разработки снижается уровень трудовых затрат на оформление однотипной документации и занесения данных в ЭВМ. Качественно меняется большинство процессов, связанных с поиском, занесением и редактированием информации.

Благодаря удобному и интуитивному интерфейсу, в котором используется большинство полезных функций ОС Windows, с программой

может работать даже неопытный пользователь. Графическая оболочка имеет простую навигационную структуру, что облегчает доступ необходимой информации.

Таким образом, данную разработку можно порекомендовать к использованию на предприятиях и учреждениях, где возникает необходимость в надежном автоматизированном средстве управления учетом компьютерной техники, в котором предусмотрены большинство необходимых функций обработки и хранения информации.

## Список используемой литературы и источников

1. Александров Д. Инструментальные средства информационного менеджмента. М.: Финансы и статистика, 2012. – 224 с.
2. Белов, В.В. Программирование в Delphi: процедурное, объектно-ориентированное, визуальное: Учебное пособие для вузов / В.В. Белов, В.И. Чистякова. - М.: ГЛТ , 2014. - 240 с.
3. Боковой Ю. Особенности методологии проектирования информационных систем для малого и среднего бизнеса. М.: Изд. дом «Синергия», 2016. – 145 с.
4. Бондарь А. Microsoft SQL Server 2012. СПб.: БХВ, 2013. – 608 с.
5. Виейра Р. Программирование баз данных Microsoft SQL Server. Базовый курс. М.: Диалектика, 2009 – 812 с.
6. Исаев Г. Проектирование информационных систем. М.: Омега-Л, 2012. – 431 с.
7. Ицик Бен-Ган. Microsoft SQL Server 2012. Основы. М.: Эксмо, 2015. – 400 с.
8. Кошелев В.Е. Access 2007. М.: Бином-Пресс, 2008. – 592 с., ил
9. Кузан, Д.Я. Программирование Win32 API в Delphi / Д.Я. Кузан. - СПб.: ВHV, 2013. - 368 с.
10. Культин Н. Основы программирования в Delphi XE. СПб.: БХВ, 2011. – 416 с.
11. Маклаков С.В. Создание информационных систем с AllFusion Modelling Suite. – М.: Диалог-МИФИ, 2003. – 432 с.
12. Осипов Д. Л. Базы данных и Delphi. Теория и практика. СПб.: ВHV, 2011.
13. Осипов Д. Delphi XE2. СПб.: БХВ, 2011. – 626 с.
14. Осипов, Д. Delphi. Профессиональное программирование / Д. Осипов. - СПб.: Символ-плюс, 2015. - 1056 с.

15. Петров В.Н. Информационные системы. СПб.: Питер, 2003. – 688 с., ил.
16. Петкович Д. Microsoft SQL Server 2012. Руководство для начинающих. СПб.: БХВ, 2014. – 567 с.
17. Пирогов В. Информационные системы и базы данных: организация и проектирование. СПб.: БХВ, 2015. – 341 с.
18. Понамарев В. Самоучитель Delphi 7 Studio. СПб.: БХВ, 2003.
19. Романова Т.Г., Романова Т.В., Белоусова А.Г. Финансовый анализ - Схемы и таблицы: учеб. пособие, 2010 – 63 с.
20. Рудикова Л.В. Базы данных. Разработка приложений. СПб.: БХВ-Петербург, 2012. – 496 с.: ил.
21. Рыбальченко М. Архитектура информационных систем. М.: Эксмо, 2016. – 221 с.
22. Стасышин Д. Проектирование информационных систем и баз данных. СПб.: Питер, 2017. – 453 с.
23. Хойтек А. Microsoft SQL Server 2012. Реализация и обслуживание. Учебный курс Microsoft. М.: Русская редакция, 2012. – 576 с.
24. Санников, Е. Курс практического программирования в Delphi. Объектно - ориентированное программирование / Е. Санников. - М.: Солон-пресс, 2013. - 188 с.
25. Станек У.Р. Microsoft SQL Server. Справочник администратора. СПб.: БХВ, 2012.
26. Стивенс Р. Программирование баз данных. Изд. 2-е. Пер. с англ. – М.: Бином-Пресс, 2013. – 384 с., ил.
27. Тейлор, Аллен SQL для чайников / Аллен Тейлор. - М.: Вильямс, 2014. - 416 с.
28. Фараонов В. DELPHI. Программирование на языке высокого уровня: Учебник для вузов. СПб.: Питер, 2015.
29. Хотек М. Microsoft SQL Server. Реализация и обслуживание. СПб.: БХВ, 2011.

## Приложение А

### Сценарий создания базы данных

```
CREATE TABLE Виды_оборудования
(
    ID          integer IDENTITY ( 1,1 ) ,
    Наименование  varchar(100) NULL
)
go
ALTER TABLE Виды_оборудования
    ADD CONSTRAINT ХПКВиды_оборудования PRIMARY KEY
CLUSTERED (ID ASC)
go
CREATE TABLE Виды_работ
(
    ID          integer IDENTITY ( 1,1 ) ,
    Наименование  varchar(150) NULL
)
go
ALTER TABLE Виды_работ
    ADD CONSTRAINT ХПКВиды_работ PRIMARY KEY CLUSTERED (ID
ASC)
go
CREATE TABLE Документы_перемещения
(
    ID          integer IDENTITY ( 1,1 ) ,
    Основание    varchar(150) NULL ,
    Оборудование  varchar(20) NULL ,
    Перемещается_из integer NULL ,
    Перемещается_vinteger NULL ,
```

Продолжение Приложения А

```
        Дата datetime NULL
    )
go
ALTER TABLE Документы_перемещения
    ADD CONSTRAINT ХПКДокументы_перемещения PRIMARY KEY
CLUSTERED (ID ASC)
go
CREATE TABLE Документы_списания
(
    ID          integer IDENTITY ( 1,1 ) ,
    Основание   varchar(200) NULL ,
    Оборудование varchar(20) NULL ,
    Дата        datetime NULL
)
go
ALTER TABLE Документы_списания
    ADD CONSTRAINT ХПКДокументы_списания PRIMARY KEY
CLUSTERED (ID ASC)
go
CREATE TABLE Должности
(
    ID          integer IDENTITY ( 1,1 ) ,
    Наименование varchar(100) NULL
)
go
ALTER TABLE Должности
    ADD CONSTRAINT ХПКДолжности PRIMARY KEY CLUSTERED (ID
ASC)
go
CREATE TABLE Журнал_работ
(
    ID          integer IDENTITY ( 1,1 ) ,
    Дата        datetime NULL ,
```

Продолжение Приложения А

```
        Описание      char(18) NULL ,
        Оборудование   varchar(20) NULL ,
        Вид_работ      integer NULL ,
        Исполнитель    integer NULL
    )
go
ALTER TABLE Журнал_работ
        ADD CONSTRAINT XPKЖурнал_работ PRIMARY KEY CLUSTERED (ID
ASC)
go
CREATE TABLE Оборудование
(
        Серийный_номер  varchar(20) NOT NULL ,
        Наименование    varchar(150) NULL ,
        Характеристика  varchar(500) NULL ,
        Год_выпуска     integer NULL ,
        Год_ввода_в_эксплуатацию integer NULL ,
        Объект_приписки integer NULL ,
        Вид_оборудования integer NULL ,
        Списано         bit NULL
)
go
ALTER TABLE Оборудование
        ADD CONSTRAINT XPKОборудование PRIMARY KEY CLUSTERED
(Серийный_номер ASC)
go
CREATE TABLE Объекты
(
        ID              integer IDENTITY ( 1,1 ) ,
        Родитель        integer NULL ,
        Наименование    varchar(150) NULL
)go
```

Продолжение Приложения А

```
ALTER TABLE Объекты
    ADD CONSTRAINT ХПКОбъекты PRIMARY KEY CLUSTERED (ID ASC)
go
CREATE TABLE Роли
(
    ID          integer IDENTITY ( 1,1 ) ,
    Наименование  varchar(50) NULL
)
go
ALTER TABLE Роли
    ADD CONSTRAINT ХПКРоли PRIMARY KEY CLUSTERED (ID ASC)
go
CREATE TABLE Сотрудники
(
    ID          integer IDENTITY ( 1,1 ) ,
    ФИО        varchar(150) NULL ,
    Логин      varchar(32) NULL ,
    Пароль     varchar(32) NULL ,
    Подразделение  integer NULL ,
    Должность  integer NULL ,
    Email      varchar(150) NULL ,
    Роль       integer NULL
)
go
ALTER TABLE Сотрудники
    ADD CONSTRAINT ХПКСотрудники PRIMARY KEY CLUSTERED (ID
ASC)
go
```

## Приложение Б

### Исходный код приложения

Листинг модуля UMain.pas

```
unit UMain;

interface

uses

  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DB, ADODB, ComCtrls, ImgList, UDataModule, DBCtrls,
  Mask, Grids, DBGrids, Menus, ExtCtrls, IniFiles, word2000, excel2000, ComObj,
  System.ImageList ;

type

  TNodeData = class
    sText : string;
  end;

  TMainForm = class(TForm)
    ImageList1: TImageList;
    PopupMenu1: TPopupMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    Panel1: TPanel;
    Label17: TLabel;
    Button6: TButton;
    PopupMenu2: TPopupMenu;
    N3: TMenuItem;
    N4: TMenuItem;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TreeView1: TTreeView;
    PageControl2: TPageControl;
    TabSheet5: TTabSheet;
    Label2: TLabel;
```

## Продолжение Приложения Б

Label3: TLabel;  
Label4: TLabel;  
Label5: TLabel;  
Label6: TLabel;  
Label12: TLabel;  
DBEdit1: TDBEdit;  
DBLookupComboBox1: TDBLookupComboBox;  
DBEdit2: TDBEdit;  
DBEdit3: TDBEdit;  
DBEdit4: TDBEdit;  
DBMemo1: TDBMemo;  
Button8: TButton;  
Button9: TButton;  
TabSheet6: TTabSheet;  
Label8: TLabel;  
Label11: TLabel;  
DBEdit8: TDBEdit;  
DBLookupComboBox2: TDBLookupComboBox;  
Button10: TButton;  
Button11: TButton;  
TabSheet8: TTabSheet;  
Label14: TLabel;  
Label15: TLabel;  
DateTimePicker1: TDateTimePicker;  
DateTimePicker2: TDateTimePicker;  
Button3: TButton;  
Button4: TButton;  
DBGrid2: TDBGrid;  
TabSheet3: TTabSheet;  
PageControl4: TPageControl;  
TabSheet13: TTabSheet;  
DBGrid6: TDBGrid;  
TabSheet14: TTabSheet;

Продолжение Приложения Б

DBGrid7: TDBGrid;  
DBNavigator1: TDBNavigator;  
TabSheet15: TTabSheet;  
DBGrid8: TDBGrid;  
DBNavigator2: TDBNavigator;  
TabSheet4: TTabSheet;  
PageControl5: TPageControl;  
TabSheet16: TTabSheet;  
Button12: TButton;  
DBGrid9: TDBGrid;  
Button16: TButton;  
TabSheet17: TTabSheet;  
Button13: TButton;  
DBGrid10: TDBGrid;  
Button14: TButton;  
Button15: TButton;  
Label1: TLabel;  
DBLookupComboBox3: TDBLookupComboBox;  
TabSheet9: TTabSheet;  
TabSheet2: TTabSheet;  
TabSheet7: TTabSheet;  
CheckBox1: TCheckBox;  
Label7: TLabel;  
Label9: TLabel;  
DateTimePicker5: TDateTimePicker;  
DateTimePicker6: TDateTimePicker;  
Button1: TButton;  
PopupMenu3: TPopupMenu;  
N5: TMenuItem;  
N6: TMenuItem;  
DBGrid1: TDBGrid;  
Label10: TLabel;  
Label13: TLabel;

## Продолжение Приложения Б

```
DateTimePicker7: TDateTimePicker;
DateTimePicker8: TDateTimePicker;
Button2: TButton;
DBGrid3: TDBGrid;
PopupMenu4: TPopupMenu;
N7: TMenuItem;
Label16: TLabel;
DBLookupComboBox4: TDBLookupComboBox;
DateTimePicker9: TDateTimePicker;
DateTimePicker10: TDateTimePicker;
Label18: TLabel;
Label19: TLabel;
Button5: TButton;
Button7: TButton;
DBGrid4: TDBGrid;
DateTimePicker3: TDateTimePicker;
Label22: TLabel;
Label23: TLabel;
DateTimePicker4: TDateTimePicker;
DateTimePicker11: TDateTimePicker;
DateTimePicker12: TDateTimePicker;
Label20: TLabel;
Label21: TLabel;
function FindIndexItem(Item: Integer): TTreeNode;
...
private
  { Private declarations }
public
  { Public declarations }
end;

var
  MainForm: TMainForm;
```

## Продолжение Приложения Б

```
SIndex:Integer;  
Num:string;  
ActiveUser:Integer;  
CurrentRole:integer;  
z:boolean;  
IniFile:TIniFile;  
MyData: PChar;  
ss:string;
```

implementation

```
uses UPPRAdd, UJournalAdd, UDocAdd, ULogin, USotrudnikAdd, UPerem, USpis;
```

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

```
procedure TMainForm.ExportDBGrid(DBGrid:TDBGrid; toExcel: Boolean; s:string);
```

```
var
```

```
  bm: TBookmark;  
  col, row: Integer;  
  sline: string;  
  str:string;  
  mem: TMemo;  
  ExcelApp: Variant;
```

```
begin
```

```
  Screen.Cursor := crHourglass;  
  DBGrid.DataSource.DataSet.DisableControls;  
  bm := DBGrid.DataSource.DataSet.GetBookmark;  
  DBGrid.DataSource.DataSet.First;
```

```
  // создаём объект Excel
```

```
  if toExcel then
```

```
  begin
```

```
    ExcelApp := CreateOleObject('Excel.Application');
```

## Продолжение Приложения Б

```
ExcelApp.WorkBooks.Add;
ExcelApp.WorkBooks[1].Worksheets[1].name := 'Grid Data';
end;

// Сперва отправляем данные в мемо
// работает быстрее, чем отправлять их напрямую в Excel
mem := TMemo.Create(Self);
mem.Visible := false;
mem.Parent := MainForm;
mem.Clear;
sline := "";

// добавляем информацию для имён колонок
for col := 0 to DBGrid.FieldCount-1 do
  if DBGrid.Columns[col].Visible=True then
    begin
      str:=DBGrid.Columns[col].Title.Caption;
      sline := sline + str + #9;
    end;
  sline:=copy(sline,0,Length(sline)-1);
  mem.Lines.Add(sline);

// получаем данные из мемо
for row := 0 to DBGrid.DataSource.DataSet.RecordCount-1 do
  begin
    sline := "";
    for col := 0 to DBGrid.FieldCount-1 do
      if DBGrid.Columns[col].Visible=True then
        sline := sline + DBGrid.Fields[col].AsString + #9;
      sline:=copy(sline,0,Length(sline)-1);
    end;
    mem.Lines.Add(sline);
    DBGrid.DataSource.DataSet.Next;
```

## Продолжение Приложения Б

```
end;
// копируем данные в clipboard
mem.SelectAll;
mem.CopyToClipboard;

// если необходимо, то отправляем их в Excel
// если нет, то они уже в буфере обмена
if toExcel then
begin
    ExcelApp.Range['A1:A1'].Value:=s;
    ExcelApp.Range['A3:A3'].Select;
    ExcelApp.Workbooks[1].WorkSheets['Grid Data'].Paste;
    ExcelApp.Visible := true;
    ExcelApp.Selection.Columns.AutoFit;
    ExcelApp.Selection.Font.Name:=DBGrid.Font.Name;
    ExcelApp.Selection.Borders[xlEdgeBottom].LineStyle:=-4119;
    ExcelApp.Selection.Borders[xlEdgeTop].LineStyle:=-4119;
    ExcelApp.Selection.Borders[xlEdgeLeft].LineStyle:=-4119;
    ExcelApp.Selection.Borders[xlEdgeRight].LineStyle:=-4119;

end;

DBGrid.DataSource.DataSet.GotoBookmark(bm);
DBGrid.DataSource.DataSet.FreeBookmark(bm);
DBGrid.DataSource.DataSet.EnableControls;
Screen.Cursor := crDefault;
end;

function TMainForm.FindIndexItem(Item: Integer): TTreeNode;
var
    i: Integer;
begin
```

## Продолжение Приложения Б

```
Result := nil;
for i:=0 to TreeView1.Items.Count-1 do
  if TreeView1.Items[i].StateIndex=Item then
    begin
      Result:=TreeView1.Items[i];
      break;
    end;
end;

procedure TMainForm.TreeView1Click(Sender: TObject);

begin

  SIndex:=TreeView1.Selected.StateIndex;
  if (TreeView1.Selected.Data)<>nil then
    Num:=TNodeData(TreeView1.Selected.Data).sText;

    If (TreeView1.Selected.Data)<>nil then
      Begin
        TabSheet5.TabVisible:=True;
        TabSheet6.TabVisible:=False;
        TabSheet7.TabVisible:=True;
        TabSheet8.TabVisible:=True;
        TabSheet9.TabVisible:=True;
        PageControl2.TabIndex:=0;
        ss:=TNodeData(TreeView1.Selected.Data).sText;
        DataModule1.TOborudovanie.Locate('Серийный_номер',TNodeData(TreeView1.Select
ed.Data).sText,[]);
        ReloadPPR;
        ReloadJournal;
      end
    else
```

## Продолжение Приложения Б

```
begin
    If TreeView1.Selected.StateIndex=1 then DBLookupComboBox2.Enabled:=False else
    DBLookupComboBox2.Enabled:=True;
    TabSheet5.TabVisible:=True;
    TabSheet6.Visible:=True;
    TabSheet7.TabVisible:=False;
    TabSheet8.TabVisible:=False;
    TabSheet9.TabVisible:=False;
    PageControl2.TabIndex:=0;
    TabSheet5.TabVisible:=False;
    DataModule1.TObjects.First;
    DataModule1.TObjects.Locate('ID',TreeView1.Selected.StateIndex,[]);
    DataModule1.QObjects.Edit;
end;
LoadTree;
end;
procedure TMainForm.LoadTree;
var
    Pnode: TTreeNode;
    Cnode: TTreeNode;
    Fnode: TTreeNode;
    i:integer;

Begin

    TreeView1.Items.Clear;
    DataModule1.ADOQuery1.Active:=False;
    DataModule1.ADOQuery1.SQL.Clear;
    DataModule1.ADOQuery1.SQL.Add('SELECT * FROM ОБЪЕКТЫ');
    DataModule1.ADOQuery1.Active:=True;
    DataModule1.ADOQuery1.First;
    while          not          DataModule1.ADOQuery1.Eof          do
```

## Продолжение Приложения Б

```

begin
  if DataModule1.ADOQuery1.FieldByName('Родитель').AsInteger = 0 then
    begin
      FNode :=
FindIndexItem(DataModule1.ADOQuery1.FieldByName('Родитель').AsInteger);
      Pnode := TreeView1.Items.AddChildFirst(FNode, "");
      Pnode.Text :=
DataModule1.ADOQuery1.FieldByName('Наименование').AsString;
      Pnode.StateIndex := DataModule1.ADOQuery1.FieldByName('ID').AsInteger;
      Pnode.ImageIndex:=0;
      Pnode.SelectedIndex:=0;
    end else
      // иначе
      begin
        FNode :=
FindIndexItem(DataModule1.ADOQuery1.FieldByName('Родитель').AsInteger);
        Cnode := TreeView1.Items.AddChild(FNode,"");
        Cnode.Text :=
DataModule1.ADOQuery1.FieldByName('Наименование').AsString;
        Cnode.StateIndex := DataModule1.ADOQuery1.FieldByName('ID').AsInteger;;
        Cnode.ImageIndex:=0;
        Cnode.SelectedIndex:=0;
      end;
      DataModule1.ADOQuery1.Next;
    end;
    // -----
    DataModule1.ADOQuery1.Active:=False;
    DataModule1.ADOQuery1.SQL.Clear;
    DataModule1.ADOQuery1.SQL.Add('SELECTСерийный_номер,      Наименование,
Вид_оборудования, Объект_приписки, СписаноFROMОборудование');
    DataModule1.ADOQuery1.Active:=True;
    DataModule1.ADOQuery1.First;

```

## Продолжение Приложения Б

```
while not DataModule1.ADOQuery1.Eof do
begin
    FNode :=
FindIndexItem(DataModule1.ADOQuery1.FieldByName('Объект_приписки').AsInteger);
    if (DataModule1.ADOQuery1.FieldByName('Списано').AsBoolean=False) then
begin
    Cnode := TreeView1.Items.AddChild(FNode,"");
    Cnode.StateIndex := 1;
    Cnode.Text :=
DataModule1.ADOQuery1.FieldByName('Наименование').AsString;
    Cnode.Data := TNodeData.Create;
        {Set the nodes date time}
    TNodeData(Cnode.Data).sText :=
DataModule1.ADOQuery1.FieldByName('Серийный_номер').AsString;;
    //Cnode.Data:=MyData;

Cnode.ImageIndex:=DataModule1.ADOQuery1.FieldByName('Вид_оборудования').AsInteger;

Cnode.SelectedIndex:=DataModule1.ADOQuery1.FieldByName('Вид_оборудования').AsInteger;
end;
    If DataModule1.ADOQuery1.FieldByName('Вид_оборудования').AsInteger=0
then
begin
    Cnode.ImageIndex:=1;
    Cnode.SelectedIndex:=1;
end;
end else if CheckBox1.Checked then
begin
    Cnode := TreeView1.Items.AddChild(FNode,"");
    Cnode.StateIndex := 1;
    Cnode.Text :=
DataModule1.ADOQuery1.FieldByName('Наименование').AsString;
```

## Продолжение Приложения Б

```
Cnode.Data := TNodeData.Create;
    {Set the nodes date time}
    TNodeData(Cnode.Data).sText :=
DataModule1.ADOQuery1.FieldByName('Серийный_номер').AsString;;
    //Cnode.Data:=MyData;

Cnode.ImageIndex:=DataModule1.ADOQuery1.FieldByName('Вид_оборудования').AsInteger;

Cnode.SelectedIndex:=DataModule1.ADOQuery1.FieldByName('Вид_оборудования').AsInteger;

    If    DataModule1.ADOQuery1.FieldByName('Вид_оборудования').AsInteger=0
then
    begin
    Cnode.ImageIndex:=1;
    Cnode.SelectedIndex:=1;
    end;
    end;
    DataModule1.ADOQuery1.Next;
end;
TreeView1.FullExpand;
For i:=0 to TreeView1.Items.Count-1 do
begin
if TreeView1.Items.Item[i].Data=Nil then
begin
    If          (TreeView1.Items.Item[i].StateIndex=SIndex)          then
TreeView1.Items.Item[i].Selected:=True
    end else
    if TNodeData(TreeView1.Items.Item[i].Data).sText=Num then
    begin
    TreeView1.Items.Item[i].Selected:=True;
    Num:='none';
    end;
```

## Продолжение Приложения Б

```
end;
end;

procedure TMainForm.FormShow(Sender: TObject);
begin
  Num:='none';
  IniFile:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'settings.ini');
  DataModule1.ADOConnection1.ConnectionString:=IniFile.ReadString('Server
Info','ConnString', 'Фигня');
  try
    DataModule1.ADOConnection1.Connected:=True;
  except
    ShowMessage('Ошибка подключения!');
  exit;
end;
with DataModule1 do
  begin
    TOborudovanie.Active:=True;
    TObjects.Active:=True;
    TRaboty.Active:=True;
    TDoljnosti.Active:=True;
    TVidy.Active:=True;
    TSotrudniki.Active:=True;
    TDoljnosti.Active:=True;
    TRoli.Active:=True;
    TPerem.Active:=True;
    TSpis.Active:=True;
    TDokumenty.Active:=True;

    end;
  Login.ShowModal;
  SIndex:=0;
```

## Продолжение Приложения Б

```
LoadTree;
PageControl2.TabIndex:=0;
TabSheet5.TabVisible:=False;
TabSheet8.TabVisible:=False;
TabSheet9.TabVisible:=False;
TabSheet6.TabVisible:=True;
TabSheet5.Visible:=False;
PageControl2.TabIndex:=1;
TabSheet16.TabVisible:=True;
end;

procedure TMainForm.ReloadPPR;
begin
  DataModule1.QPPR.Active:=False;
  DataModule1.QPPR.SQL.Clear;
  DataModule1.QPPR.SQL.Add('SELECT * FROM Журнал_работ WHERE
Оборудование =
'+DataModule1.TOborudovanie.FieldByName('Серийный_номер').AsString+'
Дата>='+DateToStr(DateTimePicker1.Date)+' AND
Дата<='+DateToStr(DateTimePicker2.Date)+'');
  DataModule1.QPPR.Active:=True;
end;

procedure TMainForm.ReloadJournal;
begin
  DataModule1.QPPR.Active:=False;
  DataModule1.QPPR.SQL.Clear;
  DataModule1.QPPR.SQL.Add('SELECT * FROM Журнал_работ WHERE
Оборудование =
'+DataModule1.TOborudovanie.FieldByName('Серийный_номер').AsString+'
Дата>='+DateToStr(DateTimePicker1.Date)+' AND
Дата<='+DateToStr(DateTimePicker2.Date)+'');
```

## Продолжение Приложения Б

```
DataModule1.QPPR.Active:=True;
end;

procedure TMainForm.ReloadPerem;
begin
  DataModule1.QPerem.Active:=False;
  DataModule1.QPerem.SQL.Clear;
  DataModule1.QPerem.SQL.Add('SELECT * FROM Документы_перемещения
WHERE Оборудование =
'+DataModule1.TOborudovanie.FieldName('Серийный_номер').AsString+'
AND
Дата>="'+DateToStr(DateTimePicker6.Date)+'"
AND
Дата<="'+DateToStr(DateTimePicker5.Date)+'"');
  DataModule1.QPerem.Active:=True;
end;

procedure TMainForm.ReloadSpis;
begin
  DataModule1.QSpis.Active:=False;
  DataModule1.QSpis.SQL.Clear;
  DataModule1.QSpis.SQL.Add('SELECT * FROM Документы_списания WHERE
Оборудование =
'+DataModule1.TOborudovanie.FieldName('Серийный_номер').AsString+'
AND
Дата>="'+DateToStr(DateTimePicker7.Date)+'"
AND
Дата<="'+DateToStr(DateTimePicker8.Date)+'"');
  DataModule1.QSpis.Active:=True;
end;

procedure TMainForm.CheckBox1Click(Sender: TObject);
begin
  LoadTree;
end;
```

## Продолжение Приложения Б

```
procedure TMainForm.ComboBox1Change(Sender: TObject);
begin
  ReloadPPR;
end;

procedure TMainForm.N1Click(Sender: TObject);
begin
  DataModule1.TJournal.Append;
  DataModule1.TJournal.FieldByName('Оборудование').AsString:=DataModule1.TOboru
dovanie.FieldByName('Серийный_номер').AsString;
  JournalAdd.Show;
end;

procedure TMainForm.Button3Click(Sender: TObject);
begin
  ReloadPPR;
end;

procedure TMainForm.Button5Click(Sender: TObject);
begin
  DataModule1.TDokumenty.Append;
  DataModule1.TDokumenty.FieldByName('Оборудование').AsInteger:=SIndex-500;
  DocAdd.Show;
end;

procedure TMainForm.Button6Click(Sender: TObject);
begin
  Application.Terminate;
end;

procedure TMainForm.N3Click(Sender: TObject);
begin
  DataModule1.TSotrudniki.Append;
```

## Продолжение Приложения Б

```
SotrudnikAdd.Show;
end;

procedure TMainForm.N4Click(Sender: TObject);
begin
  DataModule1.TSotrudniki.Edit;
  SotrudnikAdd.Show;
end;

procedure TMainForm.N5Click(Sender: TObject);
begin
  DataModule1.TPerem.Append;
  Perem.DBLookupComboBox1.KeyValue:=DataModule1.TOborudovanie.FieldByName('
Серийный_номер').AsString;
  Perem.DBLookupComboBox2.KeyValue:=DataModule1.TOborudovanie.FieldByName('
Объект_приписки').AsInteger;
  DataModule1.TPerem.FieldByName('Дата').AsDateTime:=Date;
  DataModule1.TPerem.FieldByName('Оборудование').AsString:=DataModule1.TOboru
dovanie.FieldByName('Серийный_номер').AsString;
  DataModule1.TPerem.FieldByName('Перемещается_из').AsInteger:=DataModule1.TO
borudovanie.FieldByName('Объект_приписки').AsInteger;
  Perem.Show;
end;

procedure TMainForm.N7Click(Sender: TObject);
begin
  DataModule1.TSpis.Append;
  DataModule1.TSpis.FieldByName('Дата').AsDatetime:=Date;
  DataModule1.TSpis.FieldByName('Оборудование').AsString:=DataModule1.TOborudo
vanie.FieldByName('Серийный_номер').AsString;
  Spis.Show;
end;
```

## Продолжение Приложения Б

```
procedure TMainForm.DBLookupComboBox10Click(Sender: TObject);  
begin  
z:=false;  
end;
```

```
procedure TMainForm.Button8Click(Sender: TObject);  
begin  
DataModule1.TOborudovanie.Append;  
end;
```

```
procedure TMainForm.Button9Click(Sender: TObject);  
begin  
DataModule1.TOborudovanie.Post;  
LoadTree;  
end;
```

```
procedure TMainForm.Button10Click(Sender: TObject);  
begin  
DataModule1.TObjects.Append;  
end;
```

```
procedure TMainForm.Button11Click(Sender: TObject);  
begin  
DataModule1.TObjects.Post;  
LoadTree;  
end;
```

```
procedure TMainForm.Button12Click(Sender: TObject);  
var month,s:string;  
begin  
DataModule1.QRabSot.Active:=False;
```

## Продолжение Приложения Б

```
DataModule1.QRabSot.SQL.Clear;
DataModule1.QRabSot.SQL.Add('SELECT * FROM Журнал_работ WHERE
Исполнитель='+IntToStr(DBLookupComboBox4.KeyValue)+' AND
Дата>="'+DateToStr(DatetimePicker9.Date)+'" AND
Дата<="'+DateToStr(DatetimePicker10.Date)+'"');
DataModule1.QRabSot.Active:=True;
end;

procedure TMainForm.Button2Click(Sender: TObject);
begin
ReloadSpis;
end;

procedure TMainForm.Button4Click(Sender: TObject);
begin
ExportDBGrid(DBGrid2, true, 'Журналработпообъекту');
end;

procedure TMainForm.Button15Click(Sender: TObject);
begin
ExportDBGrid(DBGrid10, true, 'Отчет по заявкам');
end;

procedure TMainForm.Button16Click(Sender: TObject);
begin
ExportDBGrid(DBGrid9, true, 'Отчет норма/факт');
end;

procedure TMainForm.Button1Click(Sender: TObject);
begin
ReloadPerem;
end;
end.
```