



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра прикладной информатики
Институт информационных систем и геотехнологий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)
по направлению подготовки 09.03.03 Прикладная информатика
(квалификация – бакалавр)

На тему «Проектирование информационной аналитической системы»

Исполнитель Аллало Тимур Сергеевич

Руководитель старший преподаватель ПИ Чернышев Дмитрий Русланович

«К защите допускаю»

и.о. заведующий кафедрой _____

доктор технических наук, профессор

Истомин Евгений Петрович

« ____ » _____ 2020 г.

Санкт-Петербург
2020

ОГЛАВЛЕНИЕ

Введение.....	3
1 Аналитическая часть.....	6
1.1 Техничко-экономическая характеристика предметной области и постановка задачи	6
1.2 Постановка задачи и анализ существующих разработок	16
1.3 Обоснование проектных решений по видам обеспечения:	25
2 Проектная часть.....	32
2.1 Информационное обеспечение задачи.....	32
2.2 Программное обеспечение задачи.....	45
2.3 Технологическое обеспечение задачи	49
3 Обоснование экономической эффективности проекта	58
3.1 Эффективность применения информационных технологий	58
3.2 Расчет экономической эффективности разработки информационной системы	62
Заключение	65
Список использованной литературы.....	67
Приложение	Ошибка! Закладка не определена.

Введение

Транспортировка и экспедирование груза, а также перевозка пассажиров - сложный и многосторонний процесс, распределенный во времени и пространстве, конечно, в него входит множество этапов, в которых задействовано значительное число участников. Чтобы планирование, организация и осуществление перевозок было результативным используют такое мощное средство как модули управления. Несомненно, на сегодняшний день автоматизация автотранспорта развивается высокими темпами. Растущая сложность задач, которые выполняют машины и механизмы, а также быстро повышающиеся запросы потребителей к электронному оборудованию транспорта, что, несомненно, приводит к постоянному развитию информационных решений, которые повышают уровень управления транспортным средством. К таким решениям можно отнести установку бортового компьютера и конечно систему навигации, также могут быть оснащение любыми другими техническими средствами.

Выпускная квалификационная работа направлена на разработку информационной системы для автотранспортного предприятия. Организации, функционирующие в течение продолжительного времени, сталкиваются с ростом объёма хранимых данных и соответственно увеличением трудовых затрат на их обработку. Так же существуют требования контролирующих органов, и налоговых служб, которые устанавливают требования наличия информационных систем.

Актуальность работы связана с тем, что информатизация становится основой дальнейшего развития транспортных и логистических систем. Все большее применение находят информационные технологии для обслуживания пассажиров на транспорте общего пользования. Рассмотрим внимательней. Создание системы автоматизации с непосредственным последующим её внедрением на предприятии обеспечит наибольшее повышение производительности, улучшит качество обслуживания и сэкономит большое

количество затрачиваемого на операции времени работникам, тем самым, несомненно, выведет предприятие на новый уровень развития, что немаловажно.

Непосредственное использование программно-аппаратных средств, безусловно, автоматизирующие бизнес-процессы, являются одним из главных инструментов увеличения результативности функционирования учетного процесса.

Очень важно отметить, что выполненный анализ дал возможность, несомненно, найти комплекс противоречий, создающие потребность в безусловном совершенствовании системы управления складским хозяйством, транспортировкой и учетом оборудования, а также конечно расходных материалов в современных условиях. К ним можно, несомненно, отнести противоречия между следующими пунктами:

- требованиями оперативности логистических операций и состоянием технологий управления АИС складским хозяйством, а также, конечно, транспортировкой;

- необходимостью, безусловно, усовершенствования автоматизированной информационной системы управления складским хозяйством и учета оборудования, а также расходных материалов и недостаточным развитием научно – методической базы по подтверждению методов ее построения и задач.

Объектом исследования является акционерное общество «Туапсинское Автотранспортное предприятие». Предмет исследования – автоматизированная экономическая информационно-аналитическая система учета оборудования и расходных материалов.

Целью в данной выпускной квалификационной работы будет являться создание информационно-аналитической системы автоматизации, дающей возможность вести учет оборудования и расходных материалов автотранспортного предприятия.

В рамках комплексного достижения обозначенной цели были определены следующие задачи исследования:

1. Определение бизнес-процессов необходимых для автоматизации;
2. Обоснование необходимости автоматизации бизнес-процессов, а также описание постановки задачи и аналогов автоматизации;
3. разработка инфологической модели данных в рамках выбранного комплекса автоматизируемых бизнес-процессов;
4. Расчет целесообразности и экономической эффективности разработки проекта.

Выбор методов исследования был обусловлен требованиями наиболее адекватного и полного решения задач на каждом этапе работы.

Методами исследования в данной выпускной работе, в зависимости от особенностей решаемых задач использовались будут являться следующие:

– абстрактно-логический метод (при постановке целей и задач работы, а также при разработке теоретических и практических основ проектирование аналитической информационной системы).;

– монографический метод (в процессе изучения тенденций развития современных проблем автоматизированных информационных системам и поиска аналогов разработки);

– экономико-математический метод (при определении экономической эффективности проекта);

– сравнительный анализ (при выборе схем функционирования АИС);

Структура и объем работы. Работа состоит из введения, трех разделов, заключения, списка использованной литературы и приложений.

1 Аналитическая часть

1.1 Техничко-экономическая характеристика предметной области и постановка задачи

Объектом данного исследования является акционерное общество «Туапсинское автотранспортное предприятие». Рассматриваемое предприятие выступает в форме юридического лица, в соответствии с Гражданским кодексом РФ которое имеет самостоятельный баланс, расчетный и иные счета в учреждениях банков.

В рамках своей деятельности АО Туапсинское АТП руководствуется следующими нормативными актами: Конституцией Российской Федерации, законами и иными законодательными актами, действующими на территории России, а также в соответствии с действующим Уставом предприятия.

Срок функционирования предприятия в различных формах собственности в соответствующей отрасли составляет 84 года. Основными конкурентами являются малые предприятия и индивидуальные предприниматели.

Характеристика использования современных информационных технологий в решении задач управления: Учет и контроль финансов, Анализ экономических процессов, Автоматизация документооборота, Анализ и построение перевозочного процесса.

Предприятие специализируется на предоставлении услуг по грузовым перевозкам, ремонту и техническому обслуживанию автотранспортных средств, оказанию услуг по проведению предрейсового и послерейсового медосмотра, оказанию услуг по проведению контрольного осмотра автотранспортных средств перед выпуском на линию, оказывает услуги по аренде помещений, платной автостоянки.

Специализация предприятия основана на непосредственном предоставлении услуг по грузовым перевозкам, ремонту и техническому обслуживанию автотранспортных средств. Также осуществляет оказание услуг по проведению предрейсового и послерейсового медосмотра, вместе с этим

осуществляются услуги по проведению контрольного осмотра автотранспортных средств перед выпуском на линию, также оказывает услуги по аренде помещений, платной автостоянки.

Согласно Уставу, основными видами деятельности АО «Туапсинское автотранспортное предприятие» являются:

- городские перевозки автобусами общего пользования;
- пригородные перевозки автобусами общего пользования;
- предрейсовый и послерейсовый технический осмотр транспортных средств;
- услуги автомобилей малой вместимости;
- предрейсовый и послерейсовый медицинский осмотр;
- диспетчерское обслуживание;
- предоставление мест для стоянки транспорта;
- заказные перевозки;
- ремонт транспорта;
- аренда помещений.

Несомненно, организацией руководит директор. Две основные службы, входящие в структуру управления это служба эксплуатации и техническая служба.

В отдел эксплуатации входят такие подотделы как отдел эксплуатации, кассу пересчета денежных средств и охрану. Основная работа отдела эксплуатации заключается в непосредственном планировании работы рейсов, маршрутов, расписания, необходимого количества автобусов, осуществление плана - графиков выхода автобусов на линии для предприятия.

Безусловно, выполнение плана доходов от перевозки пассажиров, а также поиск других возможных источников прибыли является главной задачей этой службы. Технической службой управляет главный инженер. К ее структуре относятся: производственно-технический отдел, ремонтная мастерская, также промежуточный склад запасных частей и агрегатов.

Возлагаемая рабочая функция на техническую службу заключается в

слежении за техническим состоянием подвижного состава организации, безусловно, также на базе технической службы оказывают услуги по ремонту ТО автомобилей для населения и организаций, что позволяет получить некоторый дополнительный доход.

Стоит отметить, что в структуру предприятия также входит отдел материально-технического обеспечения, функция которого заключается в снабжении расходными материалами и запасными частями, несомненно, при достаточном финансировании. Несомненно, за контролем качества поступающих запчастей и материалов выборочно следят отдел технического контроля и конечно технические работники производственных мастерских. Также за качество осуществляемых работ по техническому обслуживанию и ремонту подвижного состава и изделий собственного производства ответственны работники ОТК и контролеры технического состояния.

Для дифференциации и координации деятельности всех подразделений и задач организации необходима система управления, одним из элементов которой является структура управления организацией (рисунок 1.1).



Рисунок 1.1 - Структурная схема системы управления АО «Туапсинское АТП»

В АО «Туапсинское АТП» предприятии, безусловно, основным

источником информации является служба авто-транспорта, ведь именно там непосредственно циркулирует вся первичная информация - о состоянии парка транспортных средств, выполнении производственного задания и тому подобное. Безусловно, во многом конечно именно от полноты и достоверности этих данных зависят те или иные управленческие решения. В таблице 1 непосредственно представлена классификация основных информационных связей между субъектами системы управления АО Туапсинское АТП по типу источник-получатель. Как видно из представленной таблицы 1.1, несомненно, именно служба автотранспорта имеет наибольшее количество информационных связей с другими подразделениями предприятия.

Таблица 1.1 - Информационные связи по типу источник-получатель среди субъектов управленческой системы Акционерного бщества Туапсинское АТП

Получатель источник	Автотранспортная служба	Служба снабжения горюче-смазочным и материалам и	Финансовая служба	Служба кадров и управления персоналом	Служба технического контроля ремонта
Автотранспортная служба		Данные по расходу ГСМ	Результаты работ для расчета тарифов	Результаты работ для расчета заработной платы	Данные для планирования ТО
Служба снабжения горюче-смазочными материалами			Данные по списанию ГСМ		
Финансовая служба	Данные об объемах перевозок.				
Служба кадров и управления персоналом	Учетная информация по водителям		Данные по начислению зарплаты		
Служба технического контроля ремонта	Данные о готовности ТС к работе		Данные по использованию запчастей, планированию закупок	Данные для расчета заработной платы водителей	

Таким образом, можно смело говорить, что для непосредственной автоматизации процессов обмена информацией на автотранспортных предприятиях, несомненно, требуется разработка решений, конечно учитывающих особенности этой отрасли.

Выполненный в процессе выполнения данной диссертационной работы анализ информационной среды управления нескольких автотранспортных предприятий, также структуры потоков информации, дал возможность в подробном виде составить классификацию информационных потоков, которая непосредственно представлена на рисунке 1.2.



Рисунок 1.2 - Классификация информационных потоков

Несомненно, также основным источником управленческой информации на автотранспортном предприятии является служба автотранспорта, которая

непосредственно предоставляет данные о состоянии подвижного состава, также количестве отработанных часов водителем, объемах выполненных работ, а также соответственно другие данные. Конечно, другие подразделения непосредственно преобразовывают и обрабатывают эту информацию в зависимости от своих прямых задач и потребностей. Из этого следует, что служба автотранспорта имеет конечно наибольшее количество информационных связей внутри системы управления.

Информационные взаимосвязи объекта разработки с другими объектами, а также ряд задач, решаемых службой информационных технологий АО Туапсинское АТП представлены в таблице 1.2.

Таблица 1.2 - Информационные взаимосвязи подразделениями предприятия

Компонент ИС	Функция компонента	Куда передается	Что передается
1	2	3	4
Финансово-экономический отдел и служба информационных технологий	Отчетность о выполнении заказа Отчетность о выполнении заказа; Отчет о расходе материалов и оборудования;	Бухгалтерия Хозяйственный отдел	Рапорт о расходе материалов
Хозяйственный отдел	Формирование документов о наличии оборудования Формирование документов о отпуске компьютерного оборудования, оргтехники и расходных материалов для работы Формирование сопроводительных документов	Бухгалтерия	Приходный ордер; Ведомость наличия материалов и оборудования; Транспортная накладная;
		Финансово-экономический отдел	Комплектовочная ведомость;
Бухгалтерия	Формирование документов на покупку, модернизацию или списание оборудования и расходных материалов	Финансово-экономический отдел	Спецификация
		Хозяйственный отдел	Расходные накладные; Товарно-транспортная накладная;

Рассматриваемая технология внутримашинной организации, конечно,

будет задаваться определенной последовательностью реализуемых процедур, другими словами схем взаимосвязи программных модулей, а также, несомненно, информационных массивов. Подобная схема является декомпозицией общего процесса решения поставленной задачи на отдельные процедуры преобразования массивов, которые, в свою очередь, называются модулями. [14,с.211].

Таким образом, структуру программного продукта можно представить в виде основных блоков, которые можно рассмотреть на нижеприведенном рисунке 1.3.

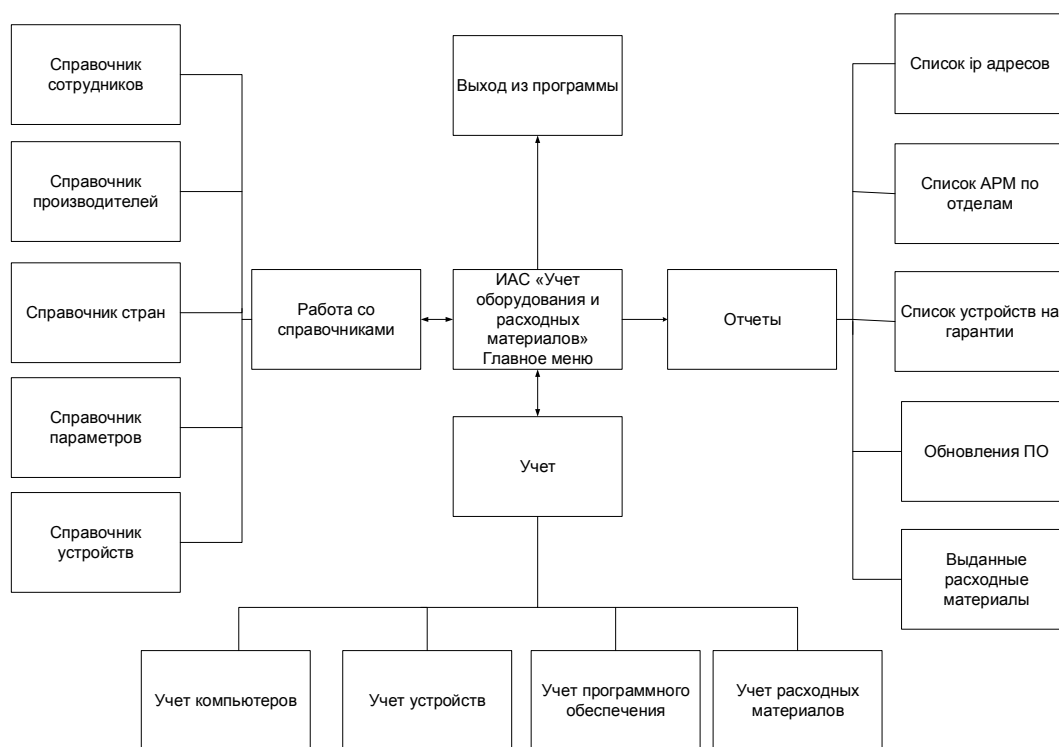


Рисунок 1.3 - Общая схема основных модулей программного продукта

Диаграмма прецедентов представлена на рисунке 1.4.

Закупкой компьютерной и копировально-множительной техники занимается бухгалтерия.

В связи с ростом объемов компьютеризации, было принято решение часть функций по мониторингу и обслуживанию оборудования офиса и контроля обеспечения расходными материалами передать хозяйственному отделу и

финансово-экономическому отделу.[15,с.135].

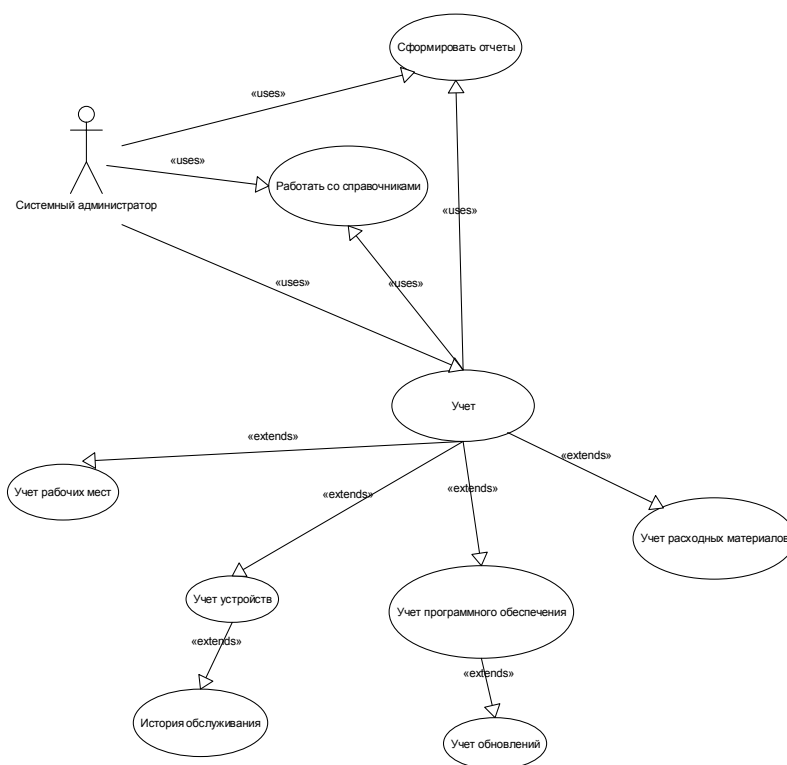


Рисунок 1.4 – Диаграмма прецедентов

Сотрудниками склада хозяйственного отдела является технический персонал, руководит работой отдела старший специалист. Старший специалист подчиняется Директору АО ТАТП и ведет количественный учет оборудования и расходных материалов. Также в обязанности старшего специалиста входит работа с персоналом предприятия по формированию заявки на обеспечение оборудованием и расходными материалами, подготовка отчетов для финансово-экономического отдела, и подготовка заявок и отчетов для бухгалтерии.

Поставка необходимых комплектующих и расходных материалов для нужд склада происходит независимо от закупок оборудования для целей монтажа на объектах портовой инфраструктуры.

Автоматизация работы данного подразделения была осуществлена путем установки уже работающей в организации системы учета. Но данная система

имеет недостатки, которые негативно сказываются на работе предприятия. Основным недостатком существующей системы является то, что весь учет ведется в суммовом выражении. Это не позволяет видеть количество запасов по всему ассортименту.

Специалист склада должен постоянно видеть остатки комплектующих на складе организации по всем необходимым наименованиям, чтобы регулировать наличие тех или иных комплектующих в своем отделе [6].

Количественный учет осуществляется вручную, что приводит к снижению эффективности работы сотрудников данного подразделения.

Таким образом, мы видим систему количественного учета традиционного типа, в которой служащий вручную помещает данные в скоросшиватель.

В результате можно сделать вывод, что отсутствие автоматизации количественного учета комплектующих и расходных материалов на складе приводит к нерациональному использованию трудовых и финансовых ресурсов сразу в нескольких подразделениях организации.

Проведя анализ работы склада оборудования, можно сделать вывод о необходимости автоматизации количественного учета комплектующих, электронно-вычислительной техники и расходных материалов в отделах. Автоматизация количественного учета позволит повысить его оперативность и эффективность.

Таким образом, в результате анализа предметной области было принято решение о создании новой автоматизированной системы, задачей которой станет ведение количественного учета компьютерной техники, комплектующих и расходных материалов, а также учёта проектов и выполнения работ по ним. Также должна быть предусмотрена возможность подготовки отчетных и статистических данных для дальнейшей обработки.

Учет с применением правильно спроектированной базы данных, несомненно, имеет свои особые особенности и, конечно, имеет радикальные отличия от используемого учета. Конечно, использование базы данных, несомненно, облегчает учет, непосредственно, значительно уменьшая

затрачиваемое время, которое может потребоваться для того чтобы оформить документы и обобщение накопленных данных для анализа хода деятельности, несомненно, необходимого для управления ею [18,с.109]. Конечно, отчеты о текущем состоянии, возможно, непосредственно получить и без помощи компьютера. Однако если рассмотреть этот вопрос более внимательно, то можно, безусловно, убедиться в нерациональности такого метода. Потому как непосредственно, затрачиваемое на расчет время может быть настолько велико, что они, несомненно, будут уже не достоверны, что конечно абсолютно не приемлемо; также есть вероятность того, что будет задействовано, конечно, очень большое количество расчетчиков, что приведет к значительным затратам превышающие доход в результате их расчетов, что также не пойдет на пользу. Несомненно, непосредственное использование компьютера «количество переходит в качество»: значительное уменьшение затрачиваемого времени на расчеты дает возможность более качественно улучшить саму систему управления предприятием, что нам на руку [31].

Безусловно, стоит отметить, что уже достаточно длительное время программное обеспечение на электронно-вычислительных машинах непосредственно применяется для работы с базами персональных данных. Несомненно, данный вид программ имеет ряд определенных недостатков, потому что данные программы были либо, как правило, элементарными диспетчерами хранения данных и в них отсутствовали средства разработки приложений, либо они были настолько значительно сложными в эксплуатации, что даже самый опытный пользователь персонального компьютера сторонится проводить с с ними работу, пока не получал возможности воспользоваться полными, ориентированными на конечного пользователя приложений [25].

Так, если учет оборудования и расходных материалов в автотранспортном предприятии ведется на элементарной основе в таблицах пакета приложений MS Office, таких как Excel или Access или аналоговым способом на бумаге, вследствие чего, практически невозможно верно определить нахождение какого-либо определенного компьютера, единицы оргтехники или конкретного

ноутбука ноутбук сотрудника, а также определить их конкретное состояние и принадлежность техники определенному отделу. Именно это обуславливает необходимость разработки удобной и надёжной программы, дающей возможность вести учет компьютерного оборудования и офисной техники на предприятии. Структуризированная база данных даст возможность оперативно получать доступ к необходимым администратору параметрам устройств, помимо этого, позволит отслеживать производимые с каждой учетной техники единицей изменения. Видя полную, целостную картину информации об оборудовании, можно с легкостью планировать и учитывать такие технические операции, как закупки, замены, обслуживания, ремонты, работы, а также списания персональных компьютеров, ноутбуков и любого другого оборудования относящегося к отделу информационных технологий рассматриваемого предприятия [32].

1.2 Постановка задачи и анализ существующих разработок

В данной работе ставится задача разработки информационно-аналитической системы по автоматизации элементов рабочего процесса в Акционерном обществе Туапсинское Автотранспортное предприятие. В современные условия одной из важнейших задач бесперебойного функционирования предприятия порта является обеспечение всех его структурных подразделений, всех рабочих мест работоспособной, отвечающей всем современным стандартам, вычислительной техникой, своевременно обслуживать и ремонтировать её, укомплектовывать необходимым программным обеспечением и расходными материалами.

В данной работе не будет затронута финансовая, бухгалтерская сторона приемки, перемещения и списания оборудования, так как для этого существуют серьезные специализированные программные комплексы, как, например, 1С Предприятие и 1С бухгалтерия, конкурировать с которыми практически невозможно. С другой стороны, в этих продуктах не делается упор на

управленческий учет оборудования и материалов, а также анализа их состояния и наличия. Разрабатываемая программа, в рамках данного проекта, позволит осуществлять мониторинг состояния и наличия оборудования (в конкретном случае компьютерной техники, оргтехники) и расходных материалов на рабочих местах, анализировать её рабочий ресурс, выставлять заявки на пополнение необходимых для функционирования материалов.

Разработка системы должна включать следующие этапы:

- анализ предметной области;
- разработка технического задания;
- разработка функциональной модели;
- разработка инфологической модели;
- разработка программных модулей автоматизированной системы;
- тестирование и отладка системы;
- оформление документации.

Не стоит забывать, что непосредственный учет оборудования и необходимых расходных материалов в АО ТАТП является, безусловно, неотъемлемой и очень важной частью непосредственного функционирования организации. Поэтому, конечно, для очень успешного функционирования предприятия, несомненно, необходимо вести непосредственный качественный учет оборудования и используемых расходных материалов, что, конечно, становится особенно актуальным в свете последних событий.

Также, непосредственный учет оборудования и необходимых расходных материалов, несомненно, должен вестись непосредственно по отделам, чтобы в абсолютно любой момент времени можно было без проблем получить сугубо подробную и абсолютно полную необходимую информацию о состоянии имеющегося в непосредственном распоряжении организации оборудования и соответственно людских ресурсах.

Для более значительного увеличения производительности труда и непосредственного уменьшения затрат времени на поиск необходимой для работы информации, безусловно, целесообразно непосредственное

использование определенных справочников компьютерных комплектующих, периферийных устройств, а также программного обеспечения, мастеров, пользователей и другой информации.

Когда пользователь запрашивает данные, конечно, необходимые на данный момент для работы, они, несомненно, должны выводиться в удобном для чтения и редактирования виде в качестве разнообразных отчетов, которые, конечно, экспортируются в систему электронных таблиц Excel.

Несомненно, нужно обратить внимание при разработке системы на легкость и простоту в использовании, конечно, неприхотливости к имеющемуся техническому и информационному обеспечению, а также отсутствие требований подключения дополнительных модулей и библиотек. Конечно, особое внимание требует непосредственная разработка интерфейса системы, так как он должен быть, конечно, простым и интуитивно понятным для пользователей с различной квалификацией [30,с.125].

По итогам проведенного анализа предметной области для разрабатываемой аналитической информационной системы было, несомненно, выделено несколько выполняемых функций:

- корректное ведение базы данных: обеспечение ввода, удаления, хранения, редактирования информационных блоков, содержащихся в таблицах данной базы данных, без нарушения ссылочной целостности;
- мониторинг правильности ввода информации;
- формирование итоговых отчетов по имеющимся данным в различных аспектах.

Настоящее техническое задание распространяется на разработку программы автоматизации учета оборудования и расходных материалов руководящими подразделениями АО ТАТП. Программное приложение предназначено для учета наличия, перемещения, прихода и расхода оборудования и расходных материалов на складе и в подразделениях организации, а также получения статистики по проделанным операциям и вывода ее на принтер.

Безусловно, предлагаемое программное приложение должно быть очень легким в простоте освоения и конечно будет рассчитано непосредственно для пользователей разной квалификации, работающих под операционной системой Windows. Несомненно, удобный интуитивно понятный интерфейс в сочетании с очень мощной системой вспомогательных рисунков, а также с помощью некоторых всплывающих подсказок предоставит возможность непосредственно работать с программой без предварительной подготовки [26,с.37].

Стоит отметить, что цветовая палитра интерфейсов, безусловно, должна совпадать со стилем операционной системы Windows и поддерживать разрешение экрана в 1024*768 — средний шрифт.

Также, конечно, должна присутствовать защита интерфейса от, соответственно, неправильных действий пользователя.

Для ввода и корректировки данных в интерактивном режиме, безусловно, необходимо использовать несколько компьютеров, объединенных в локальную сеть.

Конечно, рассмотрим минимальные требования к программному обеспечению — ОС Windows 7 и более поздних версий, Microsoft Office не ниже 2007.

Необходимые требования к техническому обеспечению — компьютер Pentium Core 2 Duo, 2 ГГц или более поздних моделей стандартной конфигурации, лазерный принтер LaserJet 1100.

Информационная система, несомненно, должна обеспечивать следующие требования к надежности:

- устойчивая работа системы в целом;
- контроль на ошибочность вводимых данных;
- проверка на допустимость числовых данных.

Разрабатываемая информационная система должна обеспечивать выполнение следующих основных функций:

- учет состояния вычислительной и копировально-множительной техники предприятия;

- учет программного обеспечения на рабочих станциях;
- учет ответственных сотрудников;
- учет расходных материалов на рабочих местах;
- учет сервисных работ.

Чтобы можно было реализовать построение необходимо провести формализацию задачи, что является нужным этапом разработки задачи и заключающуюся в построении структуры таблиц для хранения информации, схемы их взаимосвязей и описания алгоритмов обработки. Такая взаимосвязь будет показана в схеме данных на нежеприведенном рисунке 1.5.

Обычно, работа над проектом реализации задачи осуществляется в несколько этапов, для начала анализируют информацию, которая является выходной (формы и отчеты для Access). После того как будет определена структура и состав этой информации, формулировки запросов для получения отчетов можно будет сделать вывод о структуре и составе таблиц данных. Стоит отметить что, нормализация базы данных является необходимым этапом формализации задачи, которая, представляет собой процесс оптимизации хранения и использования информации в таблицах.

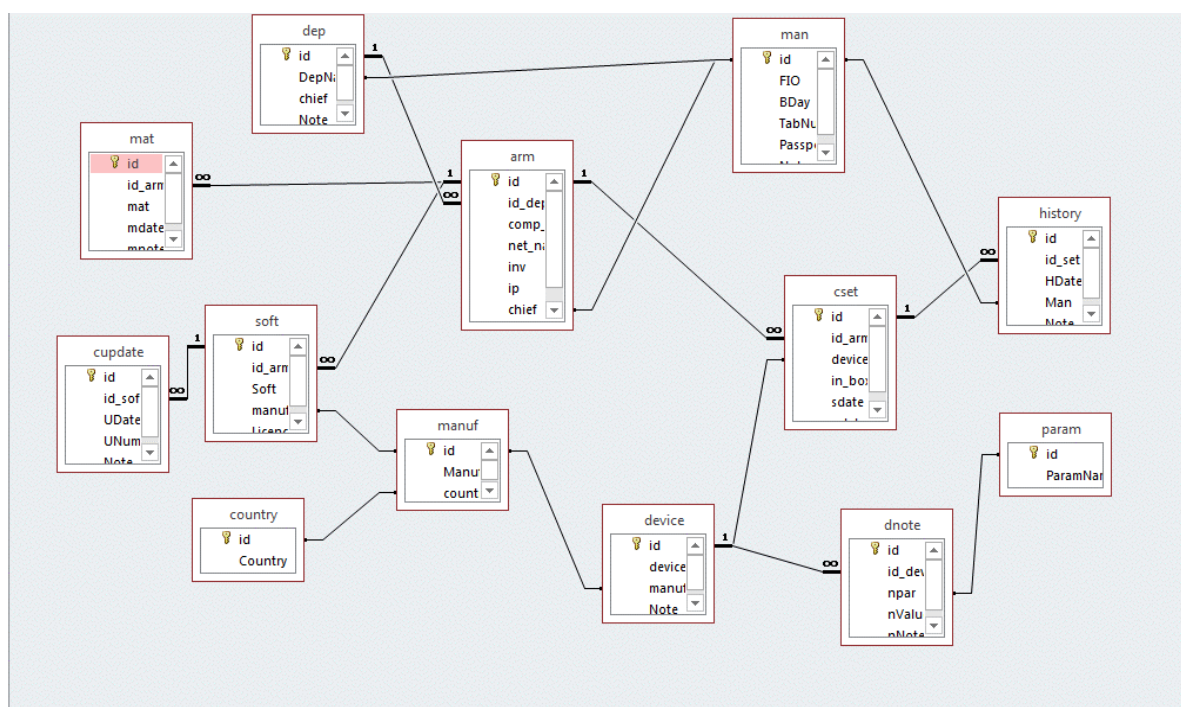


Рисунок 1.5 – Формализация расчетов

Чаще всего применяется приведение к третьей нормальной форме (3NF). Это помогает ликвидировать избыточность данных и позволяет упростить процесс изменения структуры БД.

Вначале совершается приведение к первой нормальной форме (1NF) – для обеспечения уникальности каждой записи, каждой из таблиц назначают первичный ключ. Так же вместе с этим удаляются все повторяющиеся группы (точнее создается новая таблица для повторяющихся групп) [5,с.109].

В процессе приведения ко второй нормальной форме производится ликвидация избыточности данных – любое поле в таблице, которое является избыточным или всегда остается неизменным, основанным на значении другого поля, необходимо перенести в другую таблицу.

Стоит отметить, что в 3NF все поля каждой таблицы непосредственно связаны с полем первичного ключа, и они не зависимы от других полей. Данная процедура осуществляется следующим образом – любое поле не связанное с полем первичного ключа, необходимо перенести в отдельную таблицу.

Когда идет разработка структуры данных рассматриваемой задачи изначально предполагалось следование 3NF. В Access имеется мастер анализа таблиц, который позволяет еще больше упростить этот процесс.

Далее на следующем этапе уточняется структура полей в таблицах, и определяются правила ввода.

Затем происходит возврат к проектированию запросов для форм и отчетов, проектируются собственно входные и выходные формы и отчеты.

При окончательном этапе идет создание меню, а также различных интерфейсных промежуточных форм. Конечно, процесс отладки и доработки позволяет перейти на любой из этапов разработки.

В дальнейших разделах пояснительной записки все объекты задачи можно будет рассмотреть более подробно.

Далее проведем анализ существующих разработок, выбор и обоснование стратегии автоматизации и способа приобретения ИС.

1. IT Invent. Программа IT Invent позволяет организовать

инвентаризационный учет компьютеров, учет оборудования в организации, учет офисной техники, учет оргтехники, принтеров, программного обеспечения, комплектующих, расходных материалов.

Ведение учета компьютерной техники производится в рамках организаций и их филиалов. Также есть сетевое сканирование, и инвентаризация сети, что позволяет собирать данные в автоматическом режиме. Программа для сканирования сети может помочь администраторам, сотрудникам ИТ отделов, владельцам компьютерных магазинов, заведующим складам и всем, для кого нужно иметь отчетливую и актуальную картину состояния всего ИТ оборудования (рисунки 1.6-1.7).

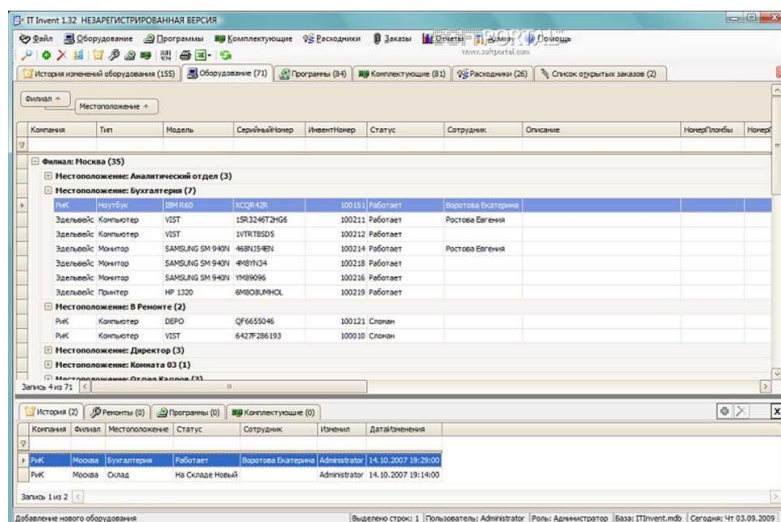


Рисунок 1.6 - Интерфейс программы «IT invent»: «Оборудование»

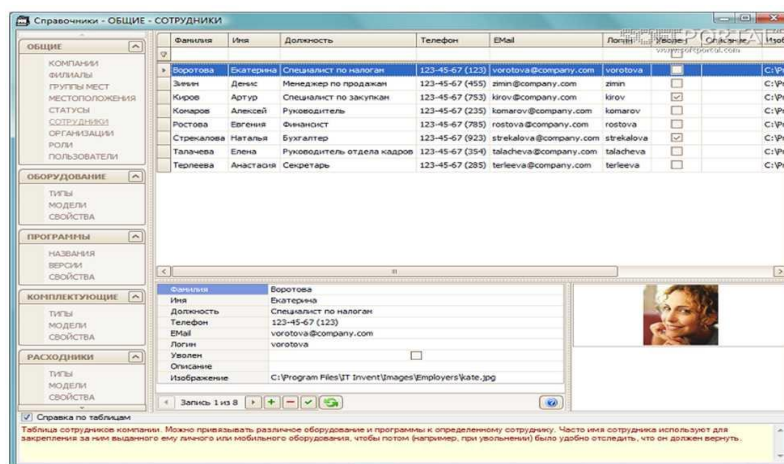


Рисунок 1.7 - Интерфейс программы «IT invent»: «Сотрудники»

2. «Учет компьютеров» - этот программный продукт разработан для учета и каталогизации компьютеров, оргтехники, оборудования и других объектов в организации (рисунки 1.8-1.9).

Есть возможность закреплять объекты учета за каждым сотрудником с разделением на отделы и филиалы.

Также возможно фиксировать производимые ремонты, замены расходных материалов и частей, профилактические работы, установленное программное обеспечение [9,с.56].

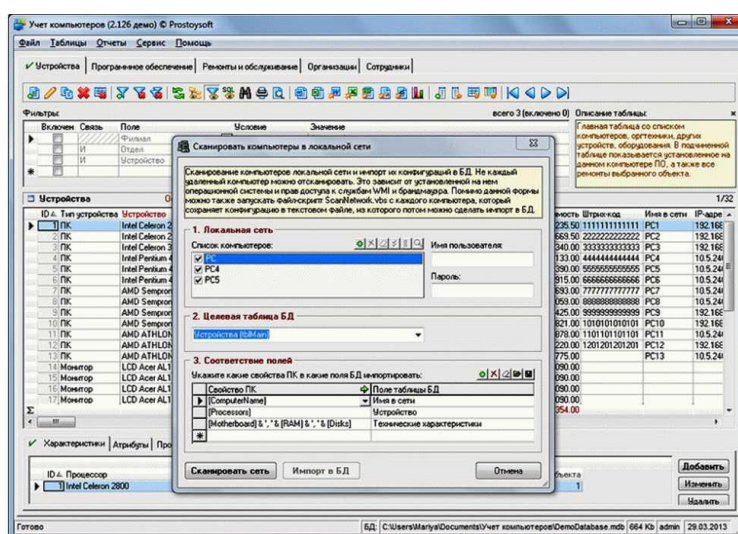


Рисунок 1.8 - Интерфейс программы «Учет компьютеров»: «Устройства»

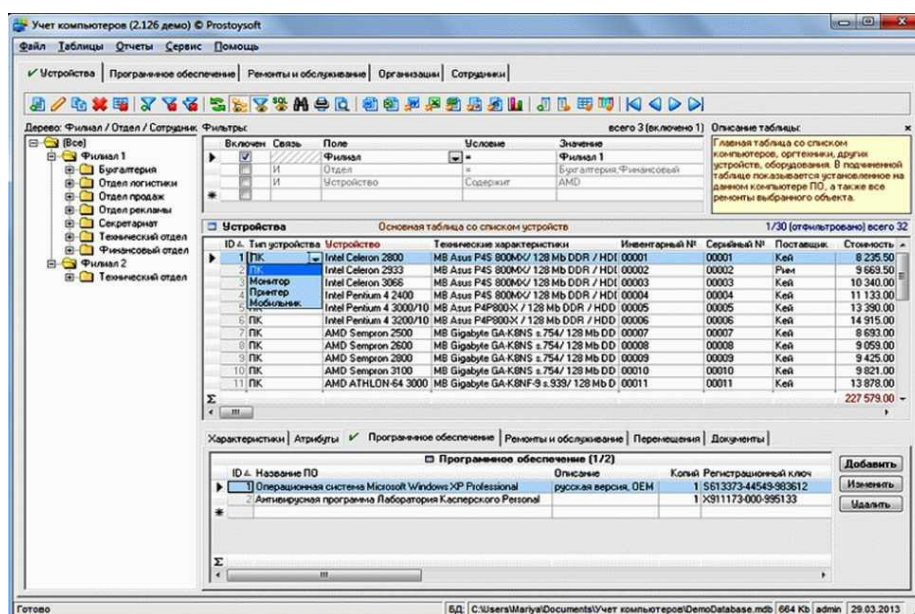


Рисунок 1.9 - Интерфейс программы «Учет компьютеров»: «Устройства»

Эта программа будет полезна системным администраторам, руководителям информационных отделов, владельцам магазинов компьютерных товаров, обладателям большого количества комплектующих и компьютерной и офисной техники, держателям домашних сетей [11].

3. «Инвентаризация сети» - это программный продукт для учёта всей компьютерной и офисной техники и программного обеспечения в сети.

Стоит добавить, что программный продукт реализовывает контроль использования аппаратных средств и лицензий ПО. Инвентаризация сети будет полезна для контроля соответствия установленного программного и аппаратного обеспечения с определенным стандартным набором для каждого рабочего места [13,с.84].

Что немало важно программа поддерживает систему отчётности выполнения установленных плановых мероприятий по всей ИТ-инфраструктуре внутри корпоративной сети (рисунок 1.10).

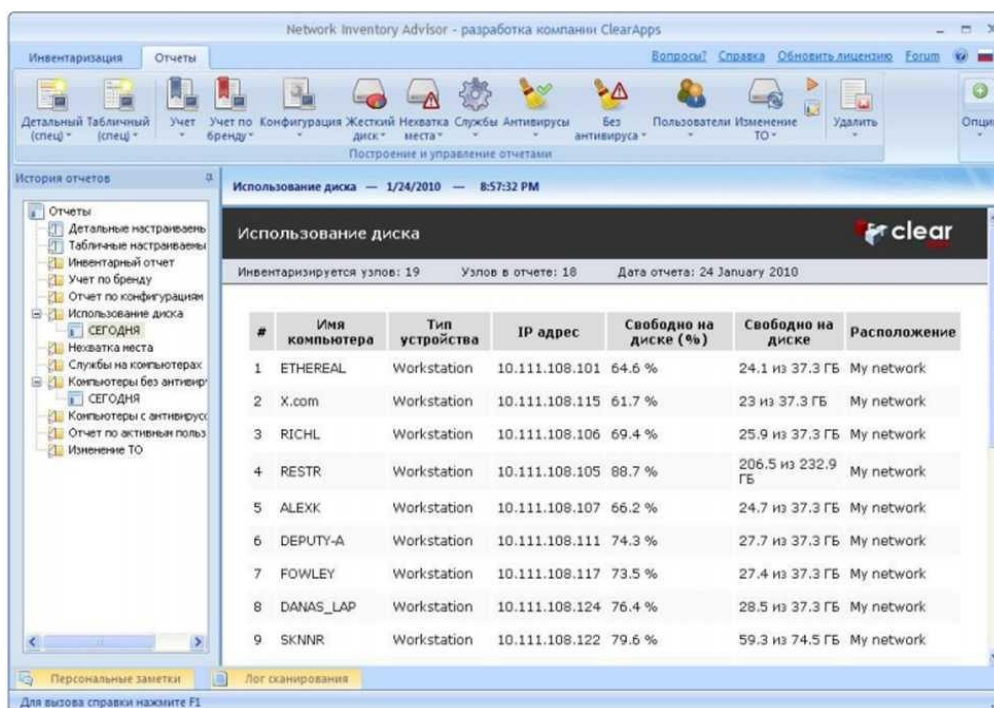


Рисунок 1.10 - Интерфейс программы «Инвентаризации сети и учет компьютеров»

Следует обратить внимание на то, что нет необходимости устанавливать

дополнительное агентское программное обеспечение на удаленный компьютер, так как инвентаризация сети выполняется в автоматическом режиме. Также есть возможность добавлять в отчёты по инвентаризации компьютеры, не подключённые к сети. Для этого достаточно запустить специальный модуль, который соберёт все данные, и их можно будет импортировать в интерфейс программы, конечно установки программы на данном компьютере не требуется.

1.3 Обоснование проектных решений по видам обеспечения:

1. Обоснование проектных решений по техническому обеспечению.

Техническим обеспечением называют комплекс технических средств, рассчитанных для работы информационной системы, а также документация на эти средства и технологические процессы [8].

Разрабатываемая система, кстати, рассчитана исключительно для непосредственного использования на персональных компьютерах, которые оснащены центральным процессором Intel Pentium или АМД, ОЗУ объемом не менее 1024 Мб, свободным дисковым пространством не менее 100 Мб. Обязательным условием является наличие манипулятора типа «мышь», TFT монитора с разрешением не менее 1024x768. Безусловно, для возможности распечатки отчетов необходимо наличие принтера.

2. Рассмотрим обоснование проектных решений по информационному обеспечению. В свою очередь информационным обеспечением называют совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих на предприятии, методология построения баз данных [17].

Информационное обеспечение данной информационной системы составляют:

- данные управленческого учета о компьютерной и офисной технике – наименование, производитель, поставщик, ответственное лицо, сведения о сервисе и ремонте;

– данные об установленном на предприятии программном обеспечении.

В процессе работы над заданием должен быть разработан пакет документов, содержащий:

- техническое задание;
- описание программы;
- программу и методику испытаний;
- описание применения.

Разрабатываемая система должна, несомненно, выдавать сообщения о возникающих ошибках при неверном задании исходных данных, конечно, поддерживать диалоговый режим в рамках предоставляемых пользователю возможностей.

Конечно, стоит обратить непосредственное внимание на то, что разрабатываемая система должна сугубо корректно реагировать на удаление, изменение данных, другими словами обеспечивать полную целостность данных. В тех случаях, когда пользователь произведет соответственно неправильные действия, система должна выдавать соответствующие предупреждения.

При необходимости модификации программной части системы пользователю необходимо обращаться непосредственно к разработчику, конечно, это означает, что конечный пользователь не должен иметь свободный доступ к исходным файлам системы.

3. Обоснование проектных решений по программному обеспечению. Программное обеспечение является программами, процедурами, правилами и соответствующей документацией системы обработки информации.

Приложение Microsoft Access является высокопроизводительной 32-разрядной системой управления реляционной базой данных (далее СУБД).

Достоинством СУБД является ввод новых данных в таблицы. Существует три критерия:

1. СУБД может легко соединять две таблицы так, что для пользователя они будут представляться одной таблицей;

2. СУБД разрабатывают, чтобы обеспечить результативную обработку крупного объёма информации;

3. СУБД отлично уменьшают общий объём базы данных. Для этого таблицы, содержащие повторяющиеся данные, разделяются на несколько связанных между собой таблиц.

Преимущество Access это несомненно простота использования даже непрофессиональными пользователями.

Таблицы. Информация сохраняется в базе данных в виде двумерных таблиц. В одно и то же время открытых таблиц может быть 1024.

В состав Access входит масса мастеров, строителей и надстроек, что позволяет упростить процесс разработки объектов базы данных.

Исходя из анализа, выбранная СУБД, удовлетворяет поставленным требованиям, поэтому выбор был остановлен на данной системе программирования.

Обратим непосредственное внимание, на которое в качестве среды для работы над разработкой нашего приложения была выбрана среда Borland Delphi 7.0, реализующая современные подходы к программированию: объектно-ориентированный подход и визуальные средства для быстрой разработки приложений (RAD-Rapid Application Development).

Среда Delphi 7.0 даёт возможность создавать приложения для непосредственно работы в Windows 95, 98, 2000, ME, XP. Версия Delphi 7.0 – отличный инструмент, который позволяет программисту любой квалификации создавать одинаково профессионально выглядящий интерфейс пользователя к прикладным программам самых различных классов [7,с.39].

Немаловажно отметить, что Delphi поддерживает основные принципы объектно-ориентированного программирования – полиморфизм, инкапсуляцию и множественное наследование. Визуальное наследование форм воплощает самый важнейший аспект объектно-ориентированного программирования в простом для использования инструменте визуального проектирования. Кстати, характеристики новой формы приложения, также могут быть непосредственно

унаследованы от любой другой существующей формы, что, несомненно, обеспечивает централизованную репродукцию изменений пользовательского интерфейса, кроме этого позволяет существенно облегчить возможность контроля за кодом, что также влияет на значительное уменьшение затрачиваемого времени, на непосредственное введение новых качественных атрибутов.

Delphi 7.0 поддерживает связь с различными базами данных: dBASE, Paradox, Oracle, InterBase, Access, FoxPro [2,с.561].

При помощи средств управления проектами, двусторонней интеграции приложения и синхронизации между средствами визуального и текстового редактирования, а также встроенному отладчику (с ассемблерным окном прокрутки, пошаговым исполнением, точками останова, трассировкой) – Delphi 7.0 корпорации Borland является неимоверно поразительной средой разработки, что позволит выдержать конкурентную борьбу с такими сильными и мощными продуктами как Developer Studio фирмы Microsoft.

Delphi - язык и среда программирования, которую относят к классу RAD - (Rapid Application Development «Средство быстрой разработки приложений») средств CASE - технологии. Delphi имеет разнообразный набор возможностей, начиная от проектировщика форм и заканчивая непосредственной поддержкой всех форматов популярных баз данных.

Мощность и гибкость языка программирования ObjectPascal – несомненное достоинство Delphi, которое очень выгодно отличает эту среду от других инструментов RAD. ObjectPascal имеет быстрееший среди продуктов подобного рода оптимизирующий компилятор, который позволяет непосредственно создавать достаточно быстрые и относительно компактные программы, что весьма немаловажно.

Безусловно опираясь на схему, приведенную непосредственно в приложении, в представленной программе разработаны программные модули на встроенном языке «Borland Delphi 7.0». В приложении приведены тексты программных модулей и запросов.

Максимального удобства при работе с программой можно достичь путем непосредственного взаимодействия с теми элементами программы, которые конечно необходимы пользователю в данный момент, а также, несомненно, пользователь не должен работать непосредственно с элементами, представляющими собой, саму структуру программы и обеспечивают её стабильную работоспособность [28,с.205].

Рассмотрим внимательней. Полная структура данной базы данных приведена далее, она состоит из справочников, документов, кроме этого её описание, описание форм и отчетов, которые конечно могут быть необходимы пользователю.

Обратим внимание на то, что по сравнению с достаточно дорогими и сложными программами складского учета типа «1С-Склад» и ей подобных, программа «АИС Учет оборудования и расходных материалов» несомненно, подойдет непосредственно представителям малого и среднего бизнеса, имея в себе все необходимое, но без ненужной перегрузки избыточными возможностями. Использование технологии создания программы в визуальных средах программирования делает ее интерфейс универсальным и совместимым с операционными системами Windows XP/Vista/7/8/10.

4. Рассмотрим обоснование проектных решений по технологическому обеспечению. Технологическое обеспечение автоматизации информационных систем состоит из подсистем, которые автоматизируют информационное обслуживание пользователей, решение задач с применением вычислительной техники, кроме этого и другие технические средства управления в установленных режимах работы.

Проектируемая информационная система должна быть спроектирована с использованием CASE-средств BPWin и ERWin [10].

Приложение должно быть разработано с применением среды разработки Delphi 7 и СУБД MS Access.

Приложение должно работать в операционных системах MS Windows 7/10. Отчеты, создаваемые приложением, должны быть совместимы с MS Excel.

Для решений поставленных задач, была выбрана операционная система (ОС) из семейства Microsoft Windows - Windows 7. В связи с быстрым распространением персональных электронно-вычислительных машин в 80-90 года прошлого века, а также привлечение большого количества неопытных пользователей в работу на ЭВМ, ОС типа Microsoft Windows стали быстро развиваться. Работа ОС заключается в процедуре ввода - вывода информации.

Windows 7 подразумевает улучшенные стандарты безопасности и надежности работы персонального компьютера. Данная операционная система - оптимальное решение, для всех пользователей как корпоративных, так и владельцев домашних персональных компьютеров [23].

Необходимость использования операционной системы Microsoft Windows 7 заключается в том, что компания Microsoft перестала поддерживать работу более ранних версий ОС и они могут работать неправильно. А на версиях ОС выше Windows 7 могут не корректно работать программы необходимые для решения поставленных задач.

Программный продукт AllFusion Process Modeler 7 (раннее BPwin) был выбран для моделирования процессов деятельности организации. Описанная функция реализовывается с помощью case-средств.

AllFusion Process Modeler 7 - инструмент для моделирования, документирования, анализа и повышения эффективности бизнес-процессов. AllFusion Process Modeler 7 возможно применить для графического представления бизнес-процессов. Графическое представление схемы работы, обмена информацией, документ показывает модель бизнес-процесса. Следовательно, графическое представление этой информации дает возможность организации перевести задачу управления сложных сделок в области инженерных технологий.

AllFusion Process Modeler 7 (BPwin) помогает отчетливо документировать существенные аспекты на любом этапе бизнес-процессов, которые нужно предпринять, способы их мониторинга и реализации, необходимых для ресурсов и наконец, есть возможность отображать полученные результаты этих действий.

AllFusion Process Modeler 7 несомненно улучшает бизнес-эффективность ИТ-решений, дает возможность аналитикам и проектировщикам моделей сравнивать корпоративные инициативы и задачи с бизнес-требованиями и процессами, информационной архитектуры и дизайна приложений. Что позволяет построить полную картину предприятия: от потоков работающих в небольших офисах до сложных организационных функций [24].

2 Проектная часть

2.1 Информационное обеспечение задачи

Работающая модель, несомненно, рассчитана для подробного описания существующих бизнес-процессов непосредственно на предприятии. Безусловно, она предписывает построение иерархической системы диаграмм – единичных описаний фрагментов системы. Конечно, рассмотрим подробнее. Безусловно, для начала реализовывается непосредственное описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), соответственно далее проводится функциональная декомпозиция, другими словами система разделяется на подсистемы и соответственно каждая подсистема описывается отдельно (диаграммы декомпозиции). Несомненно, дальше каждая подсистема разделяется на более мелкие и, кстати, так продолжается до достижения нужной степени подробности. Безусловно, описанная технология создания модели позволяет непосредственно построить модель, несомненно, адекватную предметной области на всех уровнях абстрагирования. [19,с.123].

Соответственно, взяв за основу результаты анализа предметной области, несомненно, была разработана функциональная модель системы учета компьютера и, конечно, оргтехники на предприятии.

Обратим внимание на то, что непосредственное использование в разрабатываемой системе автоматизируемых информационных потоков, несомненно, позволяет достаточно удобно непосредственно использовать нотацию DFD. Нотация DFD включает в себя понятия, соответственно, такие как внешняя ссылка и, конечно, хранилище данных, что, несомненно, позволяет ей быть более простой (по сравнению с IDEF0) для моделирования информационных потоков.

Хранилище данных, безусловно, даёт возможность непосредственно осуществить описание данных, которые, конечно, имеют необходимость для сохранения в памяти до непосредственного применения в работах. Кстати, в отличие от стрелок, несомненно, описывающих объекты в движении, хранилища

данных непосредственно изображают объекты в покое. Кстати, в материальных системах хранилища данных изображаются там, где соответственно объекты ожидают обработки, например в очереди.

Безусловно, основываясь на результатах анализа предметной области, соответственно была разработана функциональная модель системы учета компьютеров и оргтехники на предприятии. Проектирование проводилось на основе методологий IDEF0 и DFD [22,с.46].

Контекстная диаграмма, изображенная на рисунке 2.1, реализована с помощью методологии IDEF0.

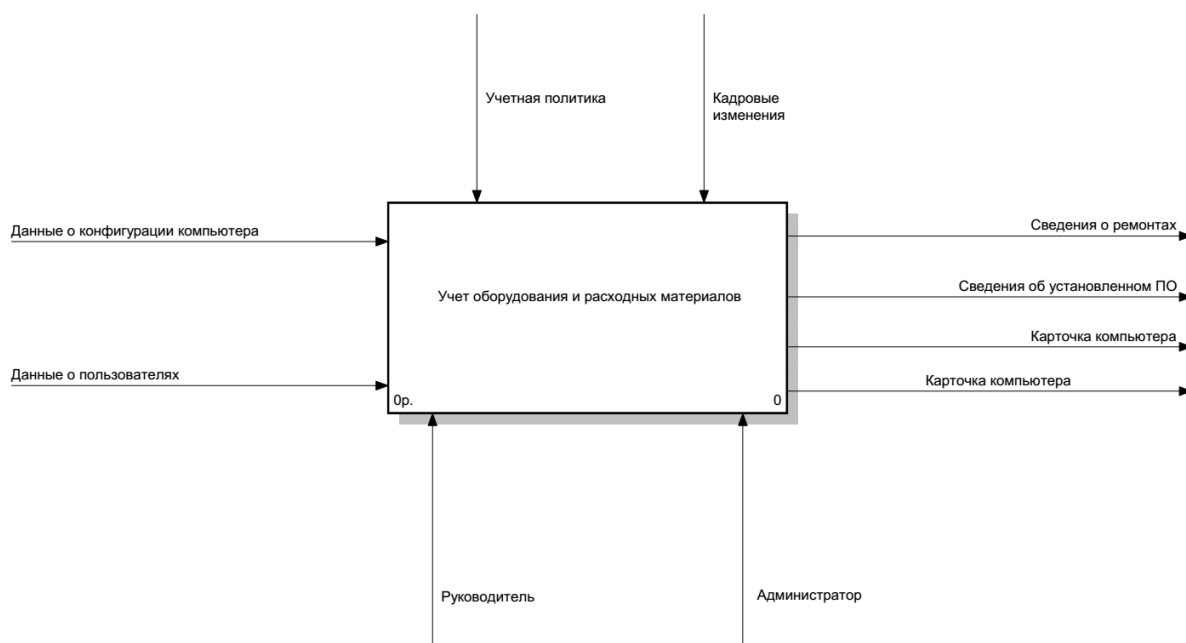


Рисунок 2.1 – Контекстная диаграмма системы

Стоит отметить, что информационное обеспечение системы, безусловно, очень тесно связано с инфологическим проектированием. Соответственно рассмотрим подробнее. Цель инфологического проектирования – конечно, обеспечить наиболее приближенный к натуральному восприятию для человека способ сбора и отображения нужной информации, которую, несомненно, рассчитывается хранить в создаваемой базе данных. Безусловно, основываясь на этих неоспоримых фактах модель данных, несомненно, стараются строить по аналогии с естественным языком (последний, кстати, несомненно, нельзя

использовать в чистом виде из-за сложности обрабатывания текстов компьютером и неоднозначности любого естественного языка). Сущности, связи между ними и их свойства (атрибуты) это важнейшие конструктивные элементы моделей.

Функциональная декомпозиция системы, приведенная на рисунке 2.2, проводится на основе методологии DFD.

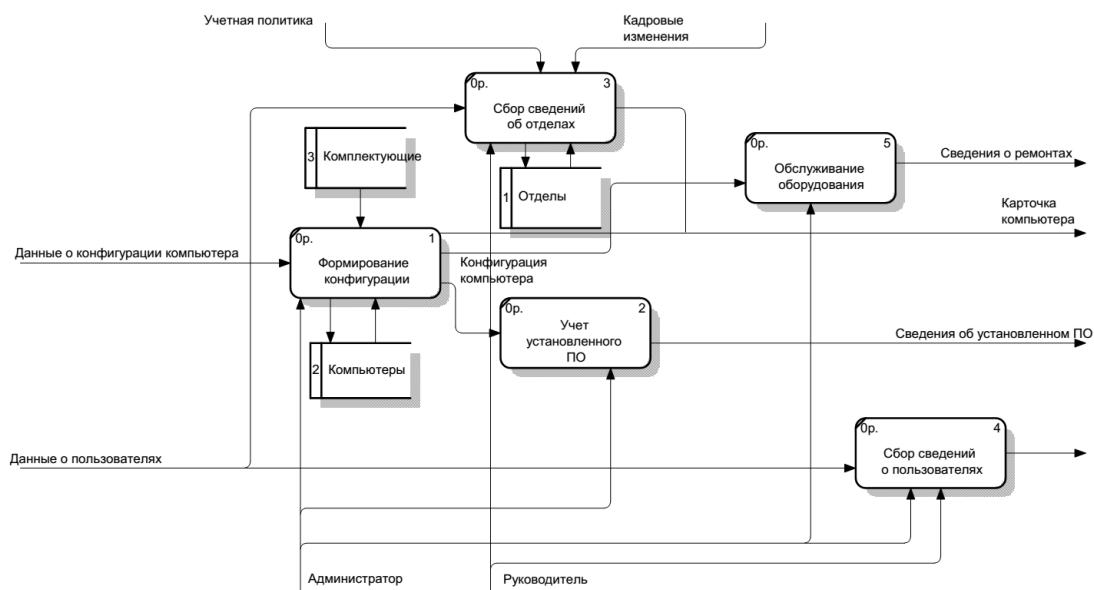


Рисунок 2.2 – Диаграмма декомпозиции системы

Возьмем во внимание разделение на таблицы, которое, несомненно, реализуется в соответствии с семантическим анализом предметной области, причем, кстати, чаще всего, каждому объекту (сущности) предметной области ставится в соответствие таблица, следовательно, атрибутам объекта соответствуют атрибуты таблицы, а идентификатору объекта, несомненно, соответствуют ключ таблицы.

Бывают случаи, что схема БД будет не удачной, другими словами могут появляться избыточность и аномалии (аномалия обновления, аномалия включения, аномалия удаления). Несомненно, нормализация данных подразумевает процедуру, которая, конечно, может обеспечить соответствие информационной модели некоторым стандартам. Это, безусловно, означает

значительную минимизацию дублирования, обеспечение гибкости, которая, несомненно, необходима для поддержки различных функциональных требований, кроме этого, конечно, создание условий для адекватного отображения модели на различные проекты БД [24,с.84].

При проектировании модели использовалась логико-физическая модель. ER-диаграмма системы на уровне сущностей отображена на рисунке 2.3.

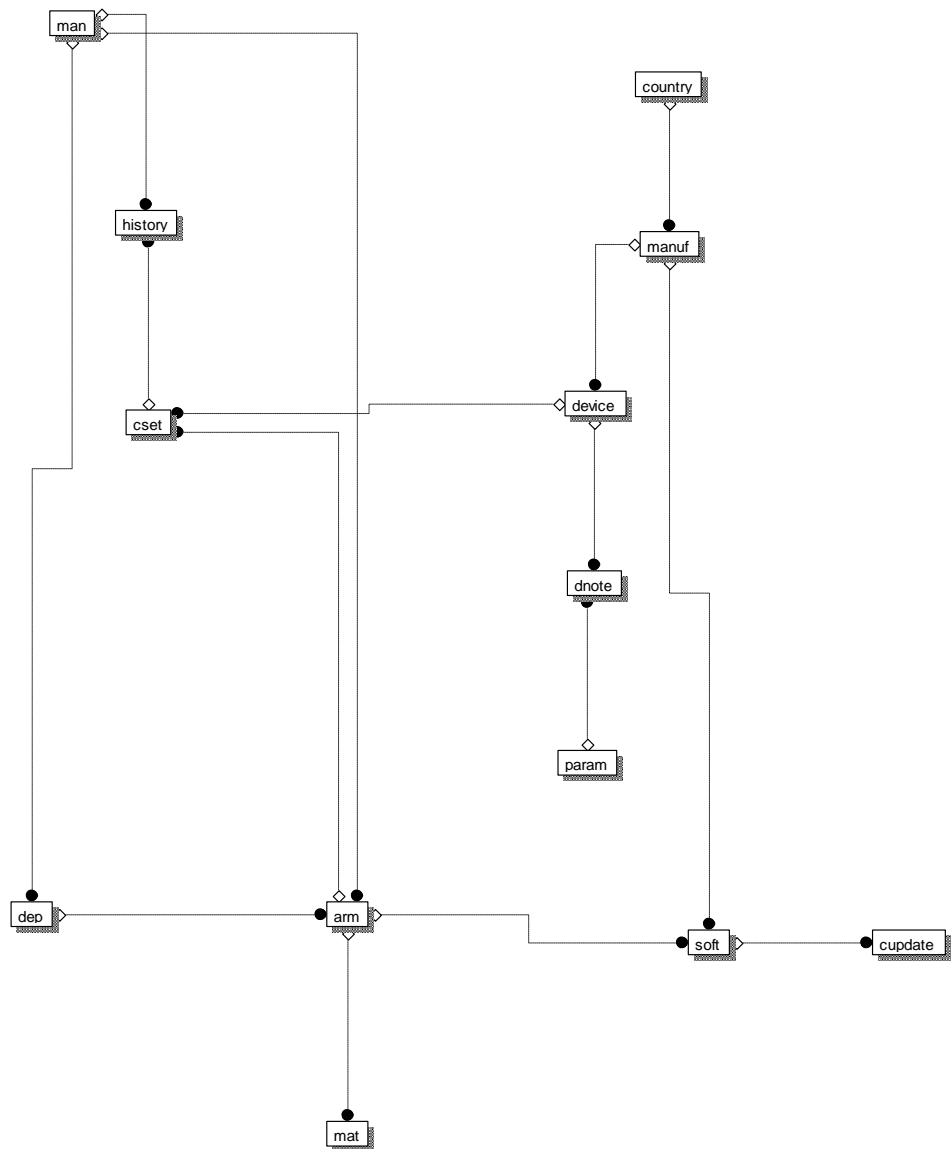


Рисунок 2.3 – ER-диаграмма системы на уровне сущностей

Процесс нормализации, несомненно, идущий параллельно с проектированием, включает в себя следующее:

- обнаружение существенных объектов, информация о которых подлежит выяснению или запоминанию. Эти сущности должны взаимно исключать друг друга;
- добавление связей, представляющих поименованные отношения между сущностями;
- для каждой сущности составляется перечень сведений (атрибутов), которые необходимо знать о ней;
- установить, каким образом каждое вхождение сущности можно идентифицировать без повторов.

Несомненно, это позволяет, разработчику построить логическую модель базы данных, не обращая внимания на детали физической реализации, т. е. непосредственно, концентрируясь на требованиях к информации и бизнес-процессам, которые будет поддерживать будущая база данных.

Во время работы над непосредственным проектированием структуры новой базы данных устанавливаются сущности (объекты, явления) предметной области, они, несомненно, должны найти свое отражение в базе данных. Объектами, несомненно, называют некую абстракцию множества предметов реального мира, где соответственно все экземпляры этого объекта имеют одинаковые характеристики и, конечно, подчиняются одним и таким же правилам поведения. Также, безусловно, объекты обладают определенными свойствами, так называемые атрибуты. Соответственно в свою очередь атрибутами называют абстракцию одной характеристики объекта. Также, безусловно, каждый из атрибутов имеет имя и может получить значения из некоторого множества допустимых значений. В большинстве случаев, каждому объекту в базе данных соответствует таблица, а его атрибутам – поля этой таблицы.

Данную предметную область описывают девять объектов, они были выявлены в процессе анализа.

Кстати, между объектами предметной области существуют связи, которые, конечно, необходимо отразить в виде связей между объектами инфологической

модели. В свою очередь связь является логическим соотношением между сущностями. Соответственно линией обозначают графически связь, соединяя связываемые объекты. Также в любом направлении связи можно отметить главный объект, от которого идет связь, и конечно подчиненный.

Исходя из особенностей предметной области, выделенных объектов и их атрибутов, разработанной структуры связей была построена ER-диаграмма на логическом уровне (рисунок 2.4).

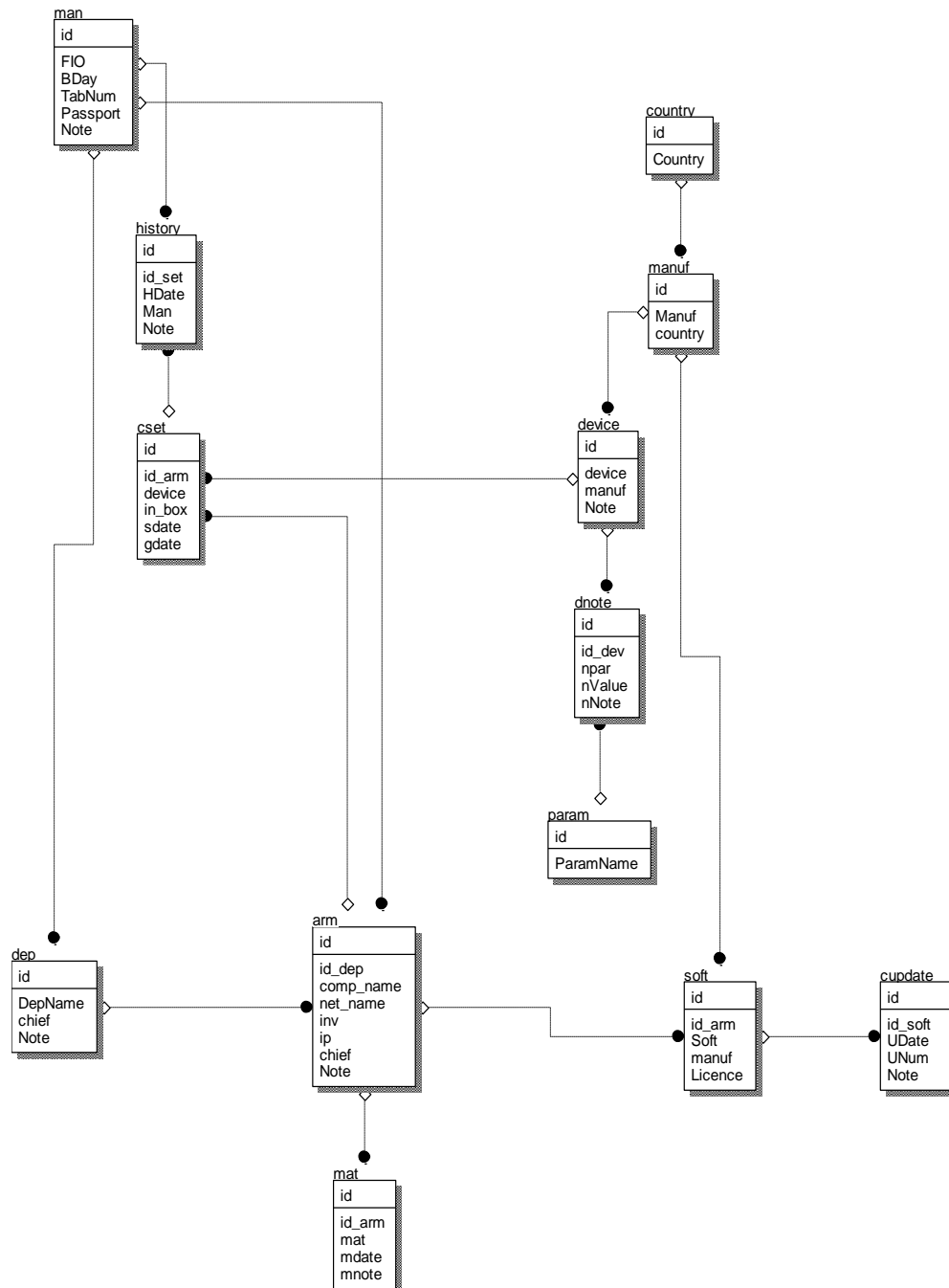


Рисунок 2.4 – ER-диаграмма системы на логическом уровне

Отметим, что сила связи задает наибольшее число экземпляров одной сущности, которые соответственно могут быть связаны с экземплярами другой сущности. В отношении «один-к-одному» один экземпляр сущности, несомненно, связан с одним экземпляром другой сущности. Также в отношении «один-ко-многим» один экземпляр родительской сущности, соответственно, связан со многими экземплярами дочерней сущности.

Следовательно, такие отношения как «многие-ко-многим» появляются, когда один экземпляр одной сущности связан с несколькими экземплярами другой, и один экземпляр этой другой сущности также связан с несколькими экземплярами первой сущности. При проектировании данной системы использовалась связь один-ко-многим.

Схема базы данных может быть неудачной: возникают избыточность и аномалии. Нормализация – это непосредственный процесс проверки и реорганизации сущностей и атрибутов чтобы удовлетворить требования к реляционной модели данных. Целью нормализации является ликвидация избыточности данных и аномалий.

Стоит отметить, в рамках реляционной модели данных имеется, несомненно, некоторое количество нормальных форм отношений (нормальные формы создают ограничение для определенного типа функциональной зависимости и ликвидируют аномалии, в то время когда выполняется операция над отношениями):

Первая нормальная форма, — если все атрибуты отношения являются атомарными (неделимыми). Обратим внимание на то, что понятие атомарности считается условным.

Обычно считают, что если значение атрибута не используется по частям, он является атомарным;

Вторая нормальная форма, — при обстоятельствах, когда отношения находятся в первой нормальной форме и всякий не ключевой атрибут функционально много зависит от составного ключа;

Третья нормальная форма, — при обстоятельствах, когда отношения

находятся во второй нормальной форме и соответственно в них отсутствуют транзитивные зависимости не ключевых атрибутов от ключа.

Конечно, для сохранения целостности данных обычно хватает третьей нормальной формы, но также есть и другие виды нормальных отношений.

Совершенный анализ схемы отношений, который показал отсутствие многозначных зависимостей в существующей инфологической модели, наконец, дает возможность прийти к выводу того, что отношения, несомненно, находятся в третьей нормальной форме.

Структура БД показана физической моделью данных.

Безусловно, физическая модель содержит всю необходимую информацию, которая, несомненно, нужна для реализации конкретной базы данных. Была произведена генерация схемы физической базы данных для модели предметной области в среде ERWin путем выбора целевой СУБД MS Access 2016. При этом сущности логической модели становятся таблицами базы данных, а атрибуты соответственно полями таблиц.

Информационное обеспечение данной информационной системы составляют данные управленческого учета АО ТАТП о компьютерной и копировально-множительной техники (расходных материалах, запчастях) – наименование, производитель, поставщик, ответственное лицо, сведения о сервисе и ремонте, кроме этого данные об установленном программном обеспечении.

Имеется наиболее простое понятие БД как достаточно крупного по объему хранилища, непосредственно в него предприятие вводит все используемые данные и, конечно, из которого, безусловно, разные пользователи могут их непосредственно получать, несомненно, с помощью разных приложений. Конечно, такая единая база данных является, несомненно, сугубо идеальным вариантом, хоть осуществить практически по разным причинам достаточно тяжело.

В процессе создания физической модели данных были скорректированы некоторые типы данных значений полей в соответствии с реализацией базы

данных. Физическая модель данных приведена на рисунке 2.5.

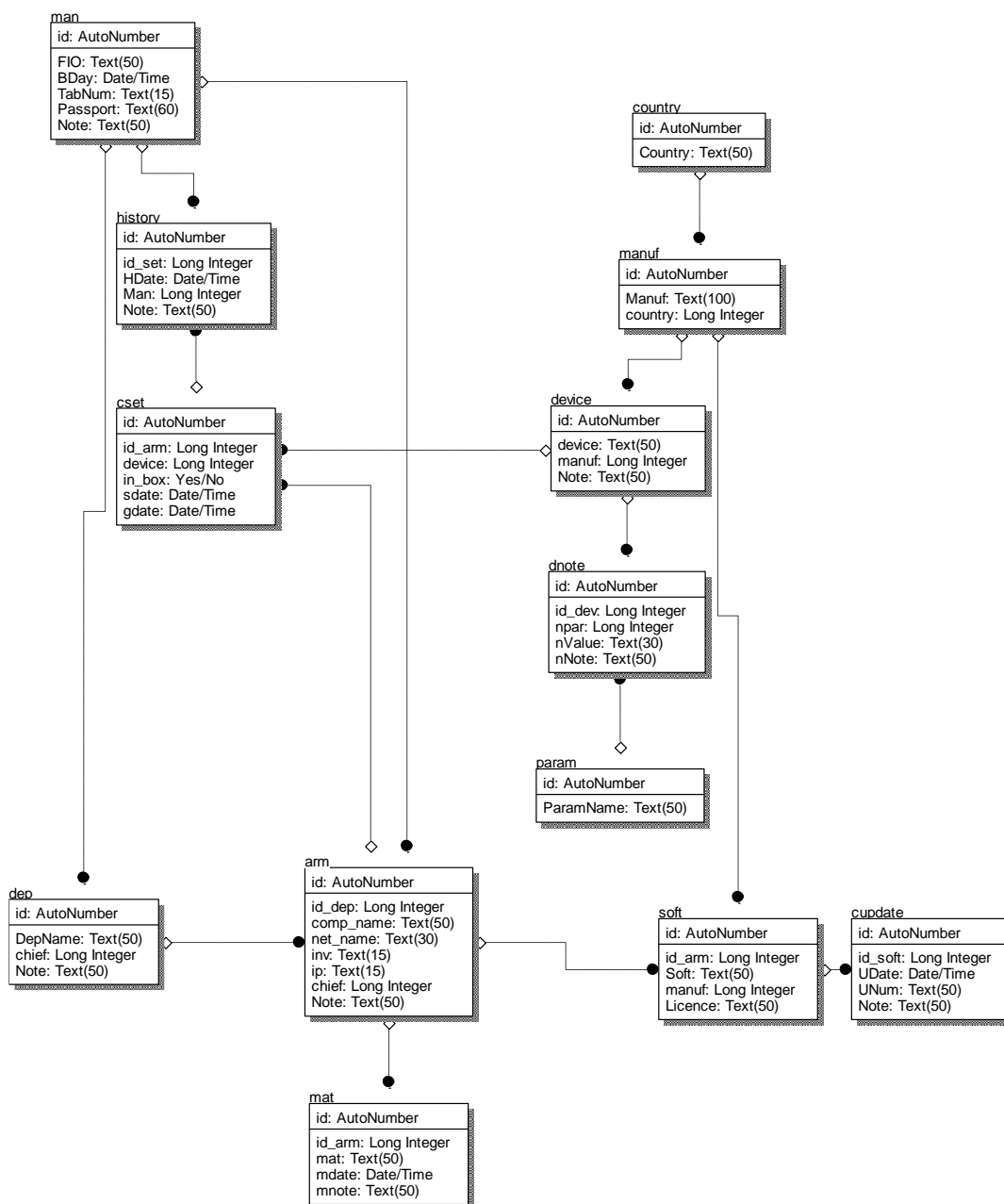


Рисунок 2.5 – ER-диаграмма системы на физическом уровне

В непосредственную основу проектирования БД необходимо заложить безусловные представления непосредственных конечных пользователей определенной организации - концептуальные требования непосредственно к системе. Безусловно, собственно конечный пользователь при выполнении непосредственно своей работы, безусловно, принимает основные решения с учетом получаемой в результате использования непосредственно базы данных

информации. Несомненно, рассматривая запросы непосредственных конечных пользователей нужно, соответственно, учитывать некоторые факты приведенные далее:

- база данных, несомненно, обязана удовлетворять актуальным на данный период времени информационным потребностям организации. Безусловно, информация, которую получает пользователь, должна, несомненно, подходить по структуре и, конечно, содержанию решаемым задачам.

- база данных, безусловно, должна обеспечивать получение требуемых данных за, соответственно, приемлемое время, другими словами, безусловно, соответствовать заданным требованиям производительности.

- база данных, несомненно, должна удовлетворять выявленным и вновь возникающим требованиям непосредственно конечных пользователей.

- база данных должна, безусловно, легко и просто расширяться при реорганизации и, непосредственном расширении предметной области.

- база данных должна, соответственно, легко и просто изменяться при непосредственном изменении программной и аппаратной среды.

- загруженные в базу данных корректные данные должны, несомненно, оставаться, конечно, корректными. Конечно, до того как данные будут включены в базу их необходимо непосредственно, проверить на достоверность.

- к данным, непосредственно располагаемым в базе данных, должны допускаться, конечно, только лица с соответствующими полномочиями.

- база данных должна иметь простейший в управлении интерфейс.

Конечно, в рамках непосредственной разработки данной автоматизированной системы, безусловно, необходимо добиться возможности, несомненно, решать нижеприведенные задачи:

1. непосредственная автоматизация формирования базы данных справочников: несомненно, для осуществления дальнейшей обработки информации, безусловно, необходимо, чтобы первичная (входная) информация была, несомненно, четко структурирована;

2. непосредственная автоматизация формирования отчётных документов.

Конечно, физическое описание модели, несомненно, удобнее всего представить, непосредственно, в виде таблиц. Следовательно, база данных проекта содержит, непосредственно таблицы, названия которых, несомненно, соответствуют именам сущностей инфологической модели. Структура БД описана в таблицах 2.1 – 2.13.

Таблица 2.1 – arm

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
id_dep	Long Integer	
comp_name	Text(50)	
net_name	Text(30)	
inv	Text(15)	
ip	Text(15)	

Таблица 2.2 – history

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
id_set	Long Integer	
HDate	Date/Time	
Man	Long Integer	
Note	Text(50)	

Таблица 2.3 – dep

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
DepName	Text(50)	
chief	Long Integer	
Note	Text(50)	

Таблица 2.4 – manuf

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
Manuf	Text(100)	
country	Long Integer	

Таблица 2.5 – cset

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
id_arm	Long Integer	
device	Long Integer	
in_box	Yes/No	
sdate	Date/Time	
gdate	Date/Time	

Таблица 2.6 – man

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
FIO	Text(50)	
BDay	Date/Time	
TabNum	Text(15)	
Passport	Text(60)	
Note	Text(50)	

Таблица 2.7 – soft

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
id_arm	Long Integer	
Soft	Text(50)	
manuf	Long Integer	
Licence	Text(50)	

Таблица 2.8 – device

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
device	Text(50)	
manuf	Long Integer	
Note	Text(50)	

Таблица 2.9 – dnote

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
id_dev	Long Integer	
npar	Long Integer	
nValue	Text(30)	
nNote	Text(50)	

Таблица 2.10 –update

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
id_soft	Long Integer	
UDate	Date/Time	
UNum	Text(50)	
Note	Text(50)	

Таблица 2.11 –country

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
country	Text(50)	

Таблица 2.12 –param

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
paramName	Text(50)	

Таблица 2.13 –mat

Название столбца	Тип данных	Первичный ключ
id	AutoNumber	Да
id_arm	Long Integer	
mat	Text(50)	
mdate	Date/Time	
mnote	Text(50)	

Конечно, для непосредственного контроля правильной работы системы, безусловно, обязательно необходимо проводить, непосредственное тестирование, в котором, конечно, нужно учесть все возможно вероятные сочетания, непосредственно, исходных данных, соответственно, отражающие работу всех частей системы. Соответственно, пояснительная записка должна, конечно, отвечать ГОСТу. Также для проведения непосредственного тестирования системы, безусловно, необходимо подготовить тестовые примеры, безусловно, охватывающие все функциональные возможности, непосредственно, разрабатываемой системы. Конечно, определить насколько правильно работает непосредственно система, несомненно, можно по результатам выполнения этих примеров.

Безусловно, сдача непосредственно программного продукта, несомненно, должна проводиться при непосредственно представлении работоспособности системы при различных входных данных, при правильном выполнении всех функций, кроме этого при наличии полной документации к программе.

2.2 Программное обеспечение задачи

В качестве среды разработки автоматизированной системы выбрана Delphi 7.0. Она, безусловно, является универсальным средством для непосредственного решения разнообразных проблем программирования, что, несомненно, является общепризнанным фактом. Кстати, также стоит обратить непосредственное внимание на то, что данная система, несомненно, более подходящая, безусловно, по сравнению с остальными средствами визуального проектирования приложений. Кроме этого, Delphi была первой системой ускоренной разработки приложений (RAD — Rapid Application Development), в ней, безусловно, удачно соединили средства визуального проектирования и, несомненно, оптимизирующий компилятор. Несомненно, добавим к этому непосредственно обширную библиотеку компонентов, конечно, полную поддержку Internet, кроме этого, несомненно, многоуровневую архитектуру баз данных. Все это, безусловно, доказывает то, что многие считают Delphi, несомненно, самой мощной системой разработки, конечно, среди всех известных систем аналогичного назначения для Windows. Конечно, Delphi нельзя назвать панацеей, но она, безусловно, приближается к данному именованию.

Разрабатываемая автоматизированная система называется «Учет оборудования и расходных материалов АО ТАТП». Система функционирует в операционных системах операционная система MS Windows XP/VISTA/7. При работе она использует базу данных под управлением СУБД MS SQL ACCESS 2007. База данных содержится в файле usomp.mdb.

Главный исполняемый файл называется UComp.exe. Доступ программы к базе данных реализуется с помощью технологии ADO (Microsoft ActiveX Data

Objects), которая обеспечивает универсальный доступ к источникам данных из приложений БД.

Функции разрабатываемой системы представлены на рисунке 2.6.

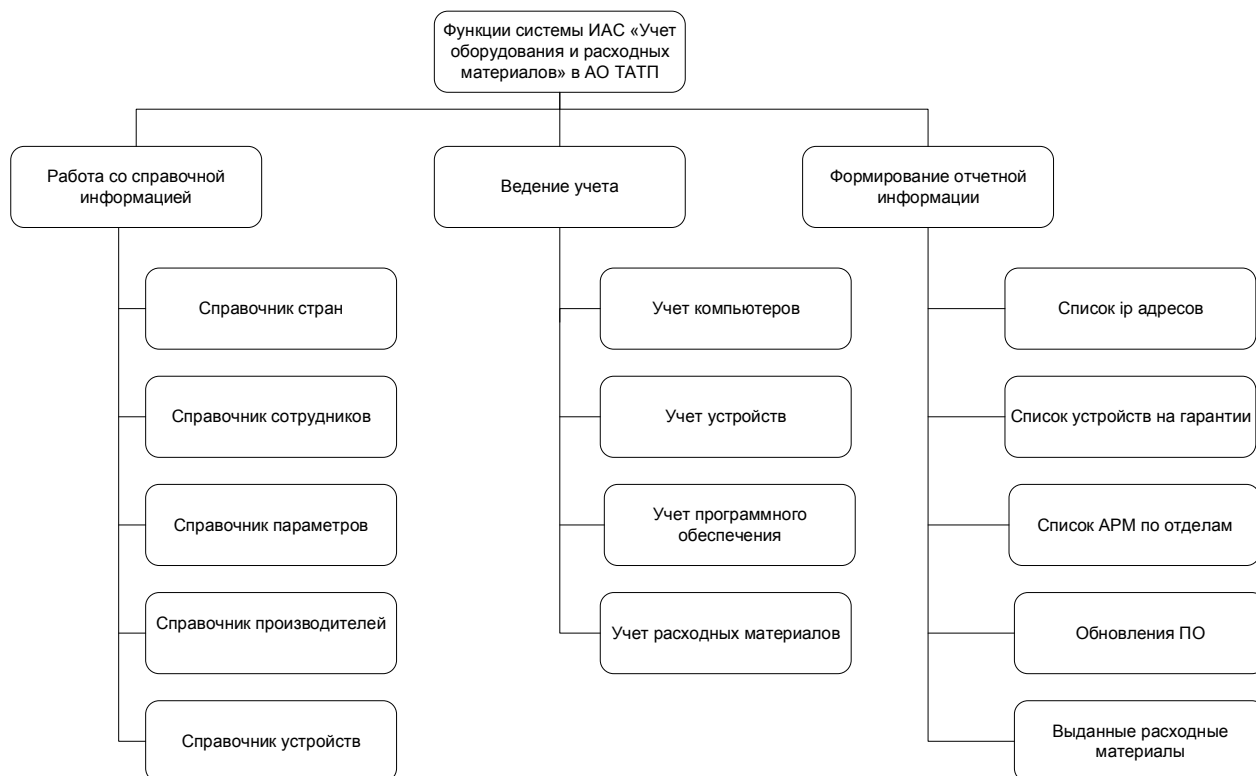


Рисунок 2.6. – Функции информационной системы

Листинги разработанных программных модулей приведены в приложении 1.

Разрабатываемая информационная система должна обеспечивать выполнение следующих основных функций:

- учет состояния вычислительной и копировально-множительной техники предприятия;
- учет программного обеспечения на рабочих станциях;
- учет ответственных сотрудников;
- учет расходных материалов на рабочих местах;
- учет сервисных работ.

Все итоговые отчеты, создаваемые программой должны иметь

возможность выгрузки данных в MS Excel.

Сценарии диалога представлены на рисунке 2.7.



Рисунок 2.7. - Сценарий диалога информационной системы

Программное обеспечение для автоматизации работы фирмы по обслуживанию компьютеров состоит из 14 модулей. Большинство из них служат для работы со справочной информацией. Опишем основные из них:

Модуль главного меню программы UnMain выполняет навигационные функции, т.е. посредством этого модуля вызываются все остальные, в которых происходит выполнение основных функций программы. Также в данном модуле сосредоточен основной функционал программы. Это модуль для учета компьютерной и копировально-множительной техники и предназначен для отображения, добавления, редактирования, удаления комплектующих, с указанием основных характеристик.

Модуль UnCountry служит для ведения справочника стран-производителей.

Модуль UnDep служит для просмотра и редактирования сведений об отделах организации.

Модуль UnDevice служит для просмотра и редактирования сведений об обустройствах и комплектующих.

Модуль UnGDate служит для задания даты для формирования отчета гарантийных устройств.

Модуль UnHistory служит для просмотра и редактирования сведений об истории обслуживания комплектующих.

Модуль UnInt служит для задания дат для формирования отчета об используемых расходных материалах.

Модуль UnMan служит для ведения справочника сотрудников организации.

Модуль UnManuf служит для ведения справочника производителей оборудования.

Модуль UnParam служит для ведения справочника параметров устройств.

Модуль UnDevice служит для просмотра и редактирования сведений об обновлениях программного обеспечения.

Безусловно, отметим, что непосредственное подключение программы к базе данных, соответственно, реализуется с помощью технологии ADO (Microsoft ActiveX Data Objects), которая, отлично, обеспечивает непосредственный универсальный доступ, конечно, к источникам данных из приложений БД. Безусловно, в модуле данных UnDM используются следующие компоненты данной технологии:

- ADOConnection – несомненно, обеспечивает доступ к источнику данных, кроме этого, непосредственно, управляет соединением с объектами хранилища данных.

- ADODataSet – конечно, предназначен для непосредственного представления набора данных из хранилища данных ADO.

- ADOTable – несомненно, обеспечивает непосредственное использование в приложении таблиц БД.

- ADOQuery – безусловно, обеспечивает применение запросов SQL при работе с данными через ADO.

Соответственно, модуль данных UnDM, непосредственно использует данную технологию и, конечно, содержит не визуальные компоненты для

непосредственной работы с базой данных, кроме этого, отвечает за связь приложения, соответственно, с базой данных. Безусловно, указанный модуль отлично используется, при непосредственной работе процедур всех вышеперечисленных модулей программы.

Соответственно, исходные тексты модулей программы кроме этого хранимых процедур приведены в приложении 1.

2.3 Технологическое обеспечение задачи

Безусловно, необходимо учитывать то, чтобы добиться максимального удобства при непосредственной работе с программой пользователю, несомненно, необходимо, конечно, использовать только с элементы программы, которые, безусловно, нужны ему на данный момент, и конечно пользователь, безусловно, не должен работать непосредственно с теми элементами, которые представляют собой саму структуру программы и, конечно обеспечивают её непосредственную работоспособность. Это, соответственно может безусловно навредить целостности структуры.

Соответственно, дальше показана непосредственно полная структура данной базы, которая состоит из справочников, конечно, документов, кроме этого, конечно, несомненно, присутствует её описание, описание форм и отчетов, необходимых непосредственно пользователю.

Кстати, вспомогательные рисунки и, соответственно, интуитивно понятный интерфейс, что немаловажно, кроме этого всплывающие подсказки, безусловно, непосредственно помогают пользователю работать с программой, конечно, не проходя дополнительной подготовки. Безусловно, все это дает непосредственную возможность освоить соответственно программу не профессиональному пользователю Windows. Конечно, этот факт положительно сказывается непосредственно на подготовке персонала.

Безусловно, немаловажно отметить факт, что по сравнению с дорогими и соответственно сложными программами складского учета типа «1С-Склад» и

конечно, соответственно похожих на нее продуктов, программа «АИС Учет оборудования и расходных материалов АО ТАТП» несомненно, подойдет, непосредственно, для представителей малого и среднего бизнеса, несомненно, из-за того, что в ней находится все необходимое для них, соответственно, не смотря на это, она, кстати, не перегружена избыточными возможностями. Непосредственное использование технологии написания программы в визуальных средах программирования, безусловно, даст возможность сделать ее интерфейс соответственно универсальным и совместимым с операционными системами Windows XP/Vista/7/8/10.

Для непосредственного функционирования системы, несомненно, необходимо наличие, соответственно, на персональном компьютере или в локальной сети серверной и клиентской части СУБД Microsoft ACCESS 2007, кроме этого, конечно, операционной системы Windows XP/VISTA/7/10.

Конечно, для непосредственного вызова программы необходимо произвести запуск файла UComp.exe на выполнение.

Соответственно, входными данными являются:

- Данные непосредственно о компьютерной и копировально-множительной техники (расходных материалах, запчастях) – соответственно, наименование, производитель, поставщик, ответственное лицо, кроме этого сведения о сервисе и ремонте;

- данные непосредственно об установленном программном обеспечении

Соответственно, выходными данными являются:

- конечно, история сервиса каждого персонального компьютера;
- несомненно, список всех комплектующих, находящихся на гарантии;
- безусловно, сведения об обновлениях ПО;
- конечно, информация об используемых расходных материалах

Безусловно, все выходные данные, конечно, должны иметь возможность выгрузки в MS Excel.

Несомненно, структура меню программы представлена на рисунке 2.8.

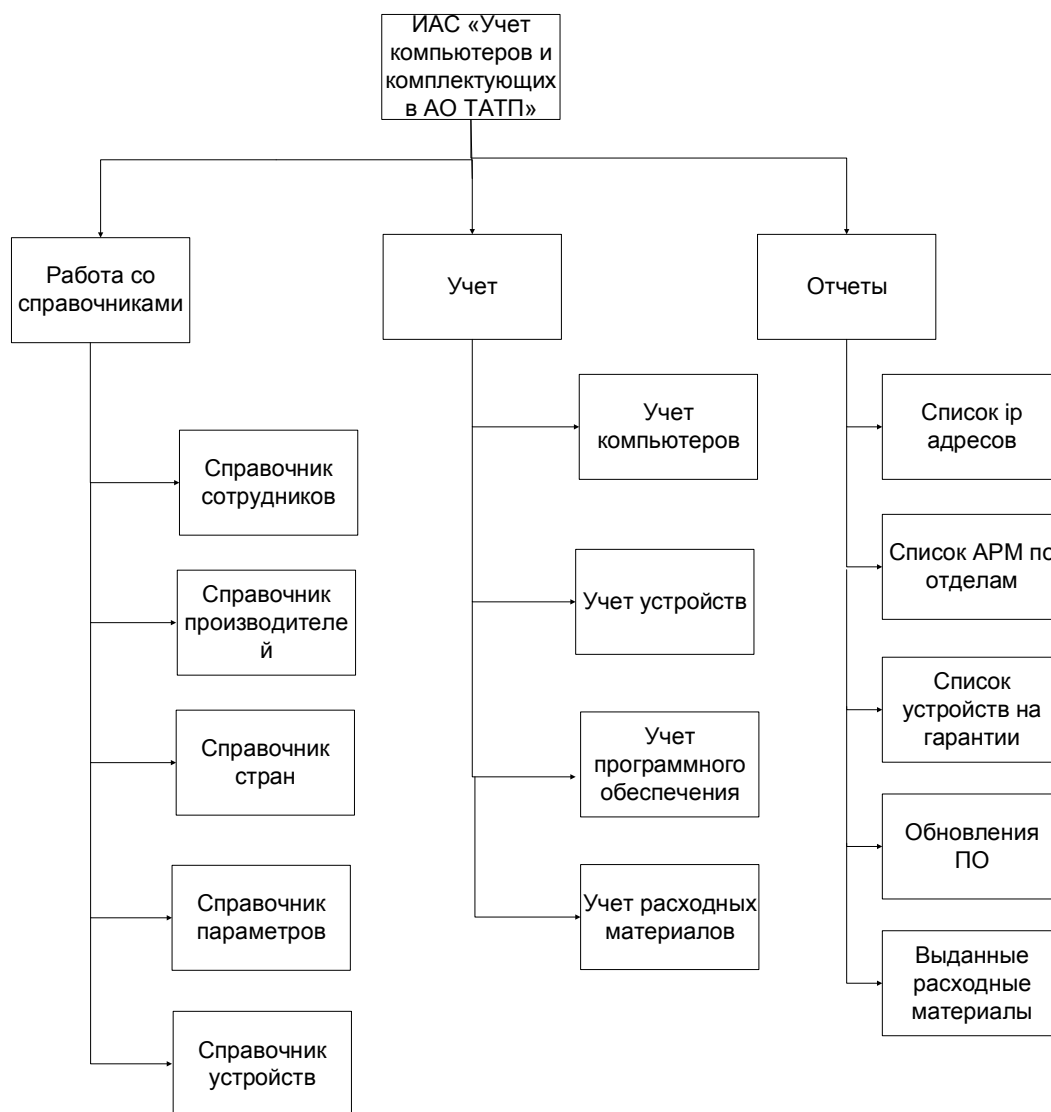


Рисунок 2.8. - Структура меню представления данных

Для вызова программы необходимо произвести запуск файла UComp.exe на выполнение.

Входными данными являются:

- данные о компьютерной и копировально-множительной техники (расходных материалах, запчастях) – наименование, производитель, поставщик, ответственное лицо, сведения о сервисе и ремонте;
- данные об установленном программном обеспечении

Выходными данными являются:

- история сервиса каждого персонального компьютера;
- список всех комплектующих, находящихся на гарантии;

- сведения об обновлениях ПО;
- информация об используемых расходных материалах

Все выходные данные, конечно, должны иметь возможность выгрузки в MS Excel.

Информативность представления информации жизненно важна во многих прикладных областях, к примеру, управление технологическим процессом на ответственных участках производства. Чем быстрее оператор увидит выход какой-либо величины за заданные границы, тем скорее он отреагирует на эту исключительную ситуацию, таким образом, уменьшая возможные последствия. Также, его реакция на подобную ситуацию непременно должна использовать некие органы управления технологическим процессом, находящиеся перед ним, если перефразировать – это также пользовательский интерфейс.

Таким образом, важность пользовательского интерфейса ни в коем случае не стоит приуменьшать.

В настоящее время выработаны определенные нормы человеко-машинного интерфейса.

Обязательно наличие главного меню программы. Главное меню находится в самом верху окна. Оно состоит из пунктов и выпадающих подпунктов. Набор пунктов различается в зависимости от типа решаемых программой задач. Практически в любой программе обязаны быть «Файл / Выход» и «Справка / О программе». Любой пункт желательно должен содержать:

- букву быстрого вызова;
- графическое обозначение (пиктограмму);
- троеточие если данный пункт меню открывает в дальнейшем какие-то дополнительные диалоги;
- обозначение комбинации клавиш быстрого вызова.

Желательно наличие дополнительного графического меню, состоящего из кнопок, дублирующих наиболее часто вызываемых операций.

В нижней части окна должна быть строка состояния, в которой должна отображаться контекстная помощь по выбранному элементу меню или кнопке.

При разработке данной программы особый акцент был сделан на разработку качественного пользовательского интерфейса с использованием самых передовых технологий его создания.

Для организации пользовательского интерфейса используется компонент `tActionList`. Данный компонент представляет собой контейнер и содержит экземпляры класса `tAction`. Здесь программист описывает все пользовательские обрабатываемые события, после чего простым перетаскиванием размещает их в виде кнопки в панели инструментов или пункта меню в верхнем меню. Таким образом, не только экономится время описания визуальных органов управления за счет однократного описания, но и упрощается работа с ними. Так если в программе изменить какое-либо свойство у одного из компонентов типа `tAction`, то это будет применено сразу ко всем связанным с ним визуальным органам управления.

Кроме использования стандартных возможностей, предоставляемых средой разработки, была также добавлена возможность настройки внешнего вида программы. По умолчанию все цвета в программе являются стандартными и соответствуют цветовой схеме Windows. Однако, если у пользователя есть желание настроить внешний вид программы по своему вкусу, он может воспользоваться возможностями тонкой настройки.

В окне «настройки» на закладке «Параметры отображения» можно для каждого из визуальных элементов можно настроить следующие параметры:

- цвет элемента;
- название шрифта;
- размер шрифта.

Для настройки доступны следующие визуальные элементы:

- поле редактирования;
- таблица данных;
- панель;
- панель выбора;
- группирующая панель;

- флажок;
- поле редактирования БД;
- надпись.

Кроме того, есть возможность сбросить параметры отображения в принятые по умолчанию, на случай если внесенные изменения не понравятся. После применения изменений данные сохраняются в БД. При создании любого из окон первым делом вызывается процедура UpdateSettings, которая для текущей формы для всех ее подэлементов применяет введенные ранее параметры.

Для запуска программы необходимо активизировать средствами ОС исполняемый файл «UComp.exe» После запуска программы открывается главное окно, представленное на рисунке 2.9.

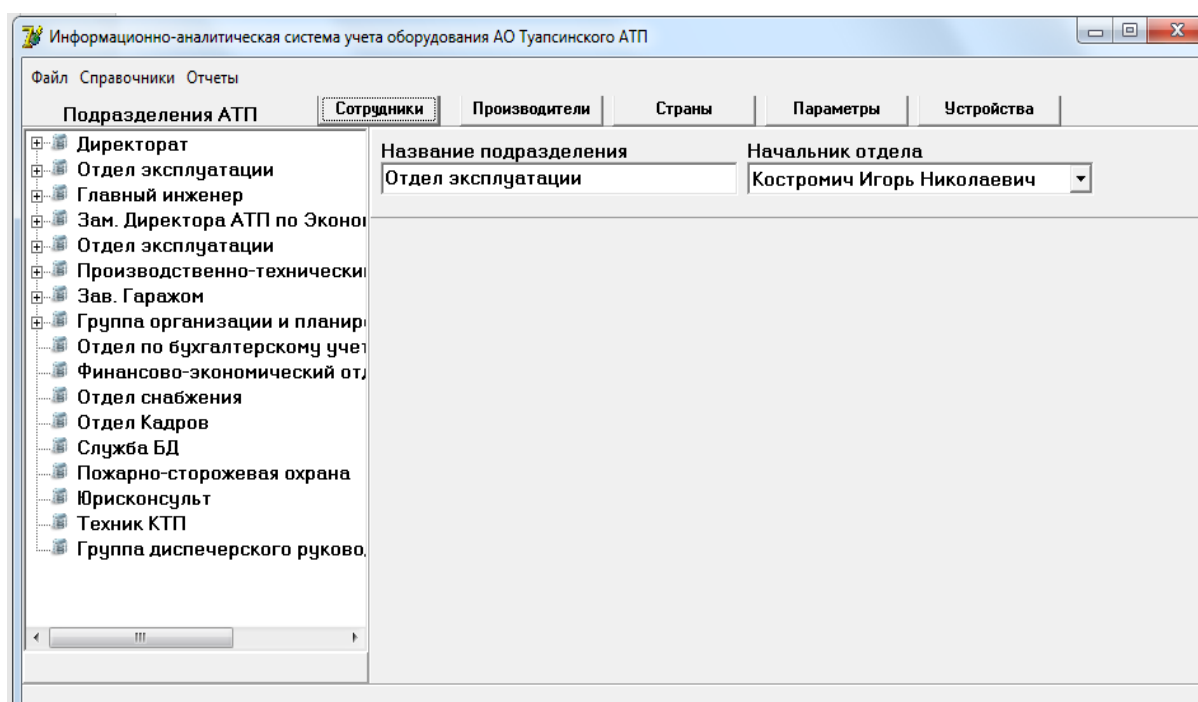


Рисунок 2.9 – Главное окно программы

При запуске программы автоматически загружается та база, с которой была работа в предыдущий раз.

В левом верхнем углу основного меню программы есть возможность выбора подразделения предприятия, а также компьютерного оборудования,

находящегося в данном отделе (рисунок 2.10).

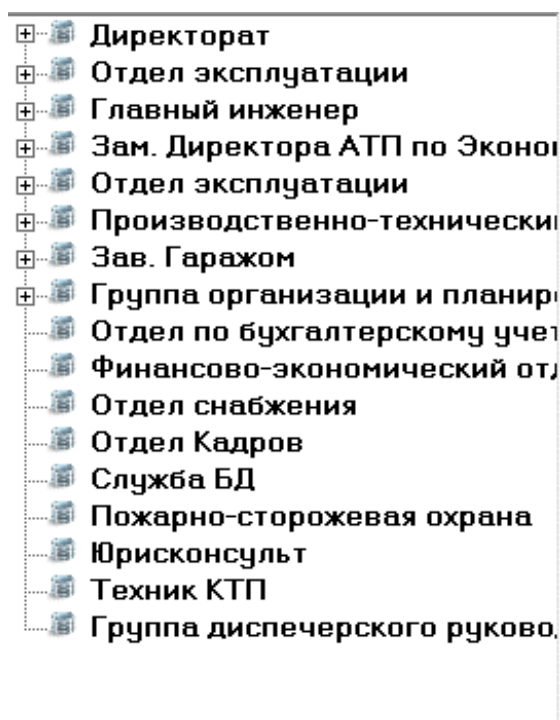


Рисунок 2.10 – Выбор подразделения

При выборе компьютера, можно увидеть полную информацию по производителю, принятию к учету, инвентарному номеру, программному обеспечению и расходным материалам, выделяемым на рабочее место (рисунок 2.11).

Название подразделения	Начальник отдела
Директорат	Манучарян Аветик Артушович
Общая информация Устройства Программное обеспечение Расход	
Название компьютера	Сетевое имя компьютера
Компьютер Евсеевой Л.И.	iv-42344
ФИО ответственного сотрудника	Инвентарный номер
Манучарян Аветик Артушович	12000450
IP адрес компьютера	<input checked="" type="checkbox"/> Карточка АРМ
Примечание	
<input type="text"/>	
<input checked="" type="button" value="Сохранить"/>	<input checked="" type="button" value="Удалить"/>

Рисунок 2.11 – Данные по оборудованию

Основными вкладками программы являются: файл, справочник, отчеты.

Программа поставляется вместе со справочной системой. Справка представляет собой набор html файлов, находящихся в поддиректории Help. При необходимости из программы запускается один из этих файлов для отображения с помощью браузера, зарегистрированного в ОС по умолчанию.

Основными вкладками программы являются: подразделения филиала, сотрудники, производители, страны, параметры, устройства.

После того как информационная система учета оборудования и расходных материалов в АО ТАТП спроектирована и реализована, необходимо провести контроль качества. Он заключается в проверке однозначности и предсказуемости действий системы при испытании ее функций, а также в соответствии этих действий ранее заявленным функциональным спецификациям. В нижеследующей Таблице 2.14 представлены тестируемые функции системы и описание сценариев, по которым должно развиваться взаимодействие пользователя и системы при их испытании.

Таблица 2.14 - Тестируемые функции системы

Тестируемая функция	Последовательность действий пользователя / реакция системы
1. Создание БД	1. Пользователь выбирает пункт меню Файл/Создать пустую базу 2. Система выводит диалоговое окно ввода имени файла для сохранения 3. Пользователь вводит имя нового файла. Диалоговое окно закрывается. Создается новый файл БД и открывается в программе 4. Пользователь нажимает отмену. Диалоговое окно закрывается.
2. Ввод данных по оборудованию	1. Пользователь выбирает пункт меню Справочники/Производители 2. Система выводит окно с таблицей поставщиков 3. Пользователь добавляет новые записи. Данные автоматически сохраняются и становятся видны в данном окне. 4. Пользователь закрывает окно. Окно закрывается.
3. Распределение оборудования по подразделениям	1. Пользователь выбирает пункт меню Справочники/Справочник устройств 2. Система выводит окно «Поступление на склад» 3. Пользователь выбирает поставщика и вводит наименование и количество поставляемых товаров. 4. После каждого изменения пересчитываются итоговые суммы и доступный объем на складе. 5. Пользователь нажимает кнопку «Карточка АРМ». Открывается файл табличного редактора с открытой карточкой АРМ

Окончание таблицы 2.14

4. Ввод сотрудников	<ol style="list-style-type: none"> 1. Пользователь выбирает пункт меню Справочники/сотрудники 2. Система выводит окно с таблицей сотрудников 3. Пользователь добавляет новые записи. Данные автоматически сохраняются и становятся видны в данном окне. 4. Пользователь закрывает окно. Окно закрывается.
5. Просмотр расходных материалов	<ol style="list-style-type: none"> 1. Пользователь выбирает пункт меню «Расходные материалы» 2. Система выводит окно с таблицей использованных расходных материалов 3. Пользователь меняет тип фильтрации. В окне отображаются данные соответствующего типа. 4. Пользователь закрывает окно. Окно закрывается.
6. Отчеты	<ol style="list-style-type: none"> 1. Пользователь выбирает один из пунктов меню Отчеты 2. Появляется окно с соответствующим отчетом. 3. Пользователь закрывает окно. Окно закрывается.

В соответствии с данным пунктом тестирование и отладка производились непосредственно во время разработки проекта. После завершения работы над программой было еще раз произведено полное тестирование.

Все найденные ошибки были успешно устранены.

Получившийся в результате программный продукт удовлетворяет всем предъявленным требованиям, полностью работоспособен, не требует принципиальных доработок, и готов к эксплуатации.

3 Обоснование экономической эффективности проекта

3.1 Эффективность применения информационных технологий

Экономический эффект заключается, безусловно, в значительном уменьшении затрат времени на выполнение работ сотрудником за счет соответствующего внедрения программного продукта «Автоматизация оборудования и расходных материалов». При расчёте экономии необходимо учитывать следующие виды работ:

1. ведение учёта по оборудованию и расходным материалам;
2. внесение и редактирование данных;
3. формирование запросов;
4. составление отчётных форм.

Для того чтобы определить необходимость внедрения данной программы, несомненно, нужно произвести подсчёт годового экономического эффекта от внедрения этой программы.

Таблица 3.1 - Распределение трудоемкости по стадиям разработки проекта

Этап разработки	Перечень работ	Трудоём . час.
Определение задач	Предпроектное обследование объекта. Разработка, согласование и утверждение технико-экономического обоснования проекта.	10
Выполнение технического проекта	Уточнение структуры и формы представления входных и выходных данных. Разработка алгоритма решения поставленной задачи. Разработка структурных блоков программного продукта. Разработка плана мероприятий по разработке и внедрению. Разработка пояснительной записки к программе. Согласование и утверждение технического проекта.	45
Разработка рабочего проекта	Реализация программного продукта на выбранном языке Первоначальная отладка и тестирование программы. Разработка, согласование и утверждение порядка и методики испытаний программного продукта. Необходимая корректировка программы.	75
Выполнение документационного обеспечения	Разработка руководства пользователя и программной документации. Подготовка к итоговой сдаче программы и программной документации	50
ИТОГО:		180

Коэффициент сложности задачи c – характеризует сложность программы по отношению к так именуемой типовой задаче, сложность которой принята равной единице (величина c лежит в пределах от 0,5 до 1,5). Для выбранного программного продукта сложность задачи берем 1,1; так как данный программный продукт не сложный.

$$c = 1,1 \text{ – (коэффициент сложности программы)}$$

Коэффициент коррекции программы p – повышение объема работ за счет ввода изменений в алгоритм или программу по результатам уточнения постановок (величина p лежит в пределах от 0,1 до 0,5). Наиболее соответствующим в этом проекте является среднее значение коэффициента коррекции, соответствующие $p = 0,4$, что можно объяснить неоднократными доработками и корректировками программы, связанные с неправильным описанием программы со стороны заказчика.

$$p = 0,4 \text{ - коэффициент коррекции программы в ходе разработки.}$$

Коэффициент повышения затрат труда b , вследствие не полного описания задачи, в зависимости от сложности задачи принимается от 1 до 1,5. Так как в данной задаче, потребовались уточнения и больших доработок, примем $b = 1,4$.

Коэффициент квалификации разработчика k устанавливается в зависимости от стажа работы и составляет:

- для работающих до двух лет - 0,8;
- от трех лет до пяти лет - 1,0;
- от пяти до восьми - 1,2 ;
- более восьми лет - 1,5.

Так как стажа работы по специальности у меня нет, возьмем $k = 0,8$.

На основании того, что оклад программиста фирмы равен 10000 рублей (З/П). С учётом использования коэффициентов заработной платы основная заработная плата разработчика программного продукта составит:

$$ЗП_{\text{осн}} = \text{Окл} * c * (p + 1) * b * k = 10000 * 1.1 * (1 + 0.4) * 1.4 * 0.8 = 17248 \text{ руб./мес} \tag{1}$$

Стоит отметить, что за время, потраченное на разработку программного продукта, дополнительная заработная плата не выплачивалась.

Страховые взносы берутся в размере 30% от основной и дополнительной заработной платы:

$$\Phi O = ЗП_{\text{осн}} * 30\% = 17248 * 30\% = 5864,32 \frac{\text{руб.}}{\text{мес}} \quad (2)$$

Полная заработная плата сотрудника за весь срок разработки программного продукта (180-10 = 170 ч.) будет составлять:

$$ЗП_{\text{общ.}} = ЗП_{\text{осн}} + СВ * \frac{170}{176} = 17248 + 5864,32 * \frac{170}{176} = 22324,4 \text{ руб.} \quad (3)$$

где, СВ – страховые взносы

Показатель величины оплаты труда сотрудника составит:

$$ЗП_{\text{сум.}} = ЗП_{\text{осн.}} + СВ = 17248 + 5864,32 = 23112,32 \text{ руб.} \quad (4)$$

Затраты, связанные с содержанием и эксплуатацией вычислительной техники составят:

$$C_{\text{вт}} = C_{\text{м-ч}} * ЧО \quad (5)$$

где, С_{м-ч} – стоимость машино-часа работы, руб.;

ЧО – число часов отладки

Для вычисления показателя часов отладки необходимо просуммировать время трех последних пунктов таблицы 3.1: технический проект, рабочий проект, документация и внедрение.

$$ЧО = 45 + 75 + 50 = 170 \text{ час.}$$

Значение показателя стоимости машино-часа будем рассчитывать, как

сумма отдельных составляющих:

$$C_{\text{м-ч}} = (\text{Электроэнергия} + \text{Амортизация} + \text{Ремонт}) / \Phi_{\text{вт}} \quad (6)$$

где, $\Phi_{\text{вт}}$ - фонд времени работы вычислительной техники.

Примем для расчетов стоимость 1 КВТ/час электроэнергии как 4,5 руб.

Одна ЭВМ будет потреблять, в среднем, около 300 Вт в час.

В течении года оплата за электроэнергию, потребляемую одним компьютером будет, составлять:

$$8\text{час.} * 22\text{дн.} * 12\text{мес.} * 0,3 \frac{\text{КВТ}}{\text{ч}} * 4,5\text{руб.} = 2851,2 \text{ руб.}$$

Обратим внимание на тот факт, что программный продукт разрабатывался при непосредственной помощи компьютера принадлежащего Туапсинскому филиалу РГГМУ. Срок полезного использования ПК составляет четыре года, так как данный компьютер проработал более четырех лет по назначению, конечно, расчет амортизации производится, не будет.

Учтем, что покупка персонального компьютера была осуществлена в декабре месяце 2012 года по стоимости 22500 руб.

Стоимость ремонта одного персонального компьютера составит:

$$\text{Рем} = 22500 * 5\% = 1125 \text{ руб.}$$

В течении одного календарного года ЭВМ будет работать в среднем 2 112 часов, на его ремонт уйдет примерно 5% времени, следовательно, окончательное полное время функционирования ЭВМ составит 2 006 часов за один год. Стоимость машинного часа равна:

$$C_{\text{м-ч}} = \frac{2851,2 + 1125}{2006} = 1,98 \text{ руб.}$$

Сумма, затраченная на содержание и эксплуатация вычислительного комплекса за время разработки программного продукта будет составлять:

$$C_{\text{с. и эк.}} = 170 * 1,98 = 3366 \text{ руб.}$$

Затраты на программное обеспечение по созданию базы, разработки продукта и написанию соответствующей документации отсутствуют, так как

лицензионная версия MS Office уже предварительно была установлена на рабочем персональном компьютере.

Итоговые расходы по разработке программного продукта представлены в таблице 3.2.

Таблица 3.2 – Итоговые расходы по разработке программного продукта

№	Наименование затрат	Затраты, руб.
1	Суммарная заработная плата разработчиков	22324,4
2	Страховые отчисления	5864,32.
3	Эксплуатация, содержание и ремонт эксплуатируемого вычислительного комплекса.	3366
4	Покупка лицензированного программного обеспечения	0
Итого:		31554,72

3.2 Расчет экономической эффективности разработки информационной системы

Далее приступим к расчету экономического эффекта, который должен быть получен за счет снижения расходов рабочего времени сотрудником на введение нужных данных после установки программы.

Также для ведения расчетов будут введены следующие обозначения:

T_1 - время, которое требуется сотруднику на поиск и составление информации до установки данного программного продукта;

T_2 - время, которое требуется сотруднику на поиск и составление информации после установки данного программного продукта.

Прежде чем была внедрена программа, сотрудник затрачивал за один день примерно в районе тридцати минут на то, чтобы найти необходимую информацию о предъявителе заявки и составить запись на прием:

$$T_1 = 40 \text{ мин.}$$

Изначально, сотрудник отдела информационных технологий выполнял десять аналоговых операций. Безусловно, после того как будет установлена программа закономерно уменьшатся и временные затраты на поиск необходимой

ему в работе информации. Таким образом, суммарные временные затраты уменьшатся, за счет автоматизации, до тридцати минут:

$$T_2 = 30 \text{ мин.}$$

Необходимо уточнить какое именно количество времени будет экономить сотрудник отдела информационных технологий акционерного общества Туапсинское АТП при использовании разработанной данной программы.

$$T = T_1 - T_2 = 10 \text{ мин.} = 0,166 \text{ час.} \quad (7)$$

Далее нужно провести расчет стоимости трудовых затрат сотрудника автоматизируемого рабочего места. За один календарный месяц (22 рабочих дня) сотрудник будет работать следующее количество часов: $8 * 22 = 176$ час.

Размер окладной части сотрудника отдела информационных технологий составляет 10 тыс. рублей.

Стоимость 1 часа (P), рассчитывается следующим образом:

$$P = \frac{10000}{176} = 56,81 \text{ руб.}$$

Следует учитывать страховые взносы, которые берем в размере $30\% = 0,30$

$$P_{\text{фо}} = \frac{10000 * 0,30}{176} = 19,32 \text{ руб.}$$

$$P_{\text{сумм}} = P + P_{\text{фо}} = 56,81 + 19,32 = 76,13 \text{ руб./час.} \quad (8)$$

Данную сумму сотрудник отдела будет зарабатывать в 1 час своей работы.

Следующим этапом будет являться расчет годового экономического эффекта. Для этого перемножим выигранное время, количество операций, количество рабочих дней в месяце и на количество рабочих месяцев

$$\Delta T_{\text{год}} = (0,166 * 10) * 22 * 12 = 438,24 \text{ час./год}$$

Поведем расчет показателя годовой экономии при внедрении разрабатываемого программного продукта:

$$\text{Год. экономия} = \Delta T_{\text{год}} * P_{\text{сумм}} = 438,24 * 76,13 = 33363,21 \text{ руб./год} \quad (9)$$

Стоит констатировать, что полученный экономический эффект будет составлять величину, равную годовой экономии средств за минусом затрат.

$$\text{Экон. эффект} = 33363,21 - 31554,72 = 1808,49 \text{ руб./год}$$

Заключительным действием будет расчет срока окупаемости разрабатываемого программного продукта:

$$T = \frac{\text{затраты}}{\text{год.экономия}} = \frac{31554,72}{33363,21} = 0,946 \text{ года} = 11,5 \text{ месяцев} \quad (10)$$

На основании проведенных расчетов, можно сделать вывод о том, что окупаемость разработанного программного продукта наступит немногим меньше, чем через год.

Заключение

В данной выпускной квалификационной работе была реализована задача автоматизации учета оборудования и расходных материалов Акционерном обществе Туапсинское Автотранспортное предприятие.

По окончанию работы были получены результаты:

- проведен анализ теоретических вопросов по теме на основе научной, учебной, производственной, справочной и др. литературы, дополнительно была использована информация, полученная с помощью глобальных сетей;
- проведен анализ состояния системы управления складским хозяйством, учетом оборудования и расходных материалов;
- разработаны модели бизнес – процессов и определены направления совершенствования системы.

Безусловно, компания становится наиболее конкурентоспособной, так как, используя информационные технологии, значительно повышаются управляемость и приспособленность к изменениям рыночных условий. Используя основу единого банка данных, при помощи подобной автоматизации, руководство и специалисты обеспечены предельно подробной, оперативной и верной информацией, что, несомненно, приводит к значительному повышению эффективности управления.

Безусловно, данная разработка дает возможность минимизировать расходы на ведение дел за счет автоматизации процессов обработки информации, регламентации и конечно упрощения доступа сотрудников компании к нужной информации, а также, кстати, освободить сотрудников от выполнения рутинной работы, что, несомненно, предоставит возможность сконцентрироваться на профессионально существенных обязанностях. Поможет обеспечить надежность учета и контроля поступлений расходных материалов и оборудования на всех уровнях управления. Также позволяет руководству нижнего и среднего звеньев анализировать работу подразделений, после чего оперативно подготовить аналитические и сводные отчеты для руководства и

смежных отделов.

Безусловно, достоинства, которые можно назвать соответственно данного разработанного программного продукта это, несомненно, удобство и, конечно, простота в непосредственной эксплуатации, кроме этого небольшой объем занимаемой памяти, что может быть немаловажно.

Несомненно, разработанное программное обеспечение, позволяет непосредственно осуществлять: автоматизированный учёт оборудования и соответственно расходных материалов; несомненно, контроль движения оборудования в определённый период времени, кроме этого выводить на непосредственную печать, безусловно, необходимую для работы на складе информацию.

Несомненно, обратим внимание на тот факт, что в данной дипломной работе была непосредственно разработана конфигурация на платформе «Borland Delphi 7.0» и «MS Access», которая, безусловно, автоматизирует процесс учета оборудования и соответственно расходных материалов. Прделанная работа по написанию непосредственно программы была выполнена, конечно, вполне успешно, так как информационная программа, несомненно, полностью соответствует поставленной задаче.

Программный продукт, несомненно, отлично позволяет выполнять поиск информации по разнообразным критериям, изменять и соответственно вводить данные, производить расчеты по складу, кроме этого, конечно, выводить на печать необходимую документацию.

Наконец, была совершена экономическая оценка эффективности от непосредственного внедрения программного продукта. Дающая нам основания полагать, что разработанный программный продукт окупит себя менее, чем за 11 месяцев, что, кстати, является приемлемым в качестве срока окупаемости затрат по автоматизации рабочего процесса.

Список использованной литературы

1. Андрончик, А. Н. Защита информации в компьютерных сетях. Практический курс: учеб. пособие / под ред. Н. И. Синадского. - Екатеринбург: УГТУ - УПИ, 2017. - 248 с.
2. Архангельский, А. Я. Программирование Delphi 7 - М.: Бином-Пресс, 2018. — 1152 с.
3. Бабенко, Г. В. Анализ современных угроз безопасности информации, возникающих при сетевом взаимодействии // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. 2017. - №2 -С.149-152.
4. Биячуев, Т. А. Безопасность корпоративных сетей / под ред. Л. Г. Осовецкого. - СПб.: СПбГУ ИТМО, 2016. - 161 с.
5. Благодатских, В. А. и др. Стандартизация разработки программных средств: учеб. пособие / В. А. Благодатских, В. А. Волнин, К. Ф. Посакалов; Под ред. О. С. Разумова. М.: Финансы и статистика, 2015. -511 с.
6. Благодатских, В. А. и др. Экономика, разработка и использование программного обеспечения ЭВМ учеб. пособие / В.А. Благодатских, М.А. Енгибарян, Е.В. Ковалевская и др.; М.: Финансы и статистика, 2015. -288 с.
7. Бобровский, С. И. Delphi 7. учеб. курс / С. И. Бобровский. — СПб.: Питер, 2017. — 736 С.:
8. Бондаренко, М.Ф. Системологическая технология моделирования информационных и организационных систем: [Текст] учеб. пособие / М.Ф. Бондаренко, Е.А. Соловьева, С.И. Маторин, Д.Б. Ельчанинов. – Х.: ХНУРЭ, 2015. – 136 с.
9. Борщёва, Н.Л. Информационные системы в экономике: учеб. пособие. / Н.Л. Борщёва. – Издательство Томского политехнического университета. –Томск: 2016. – 115 с.
10. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М.: Финансы и статистика, 2017. – 367с.

11. Гаджинский, А.М. Современный склад. Организация, технологии, управление и логистика: учеб.-практ. пособие. – М.: Проспект, 2018. – 176 с.
12. Голицина, О. Л Базы данных: учеб. пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - 2-е изд., испр. и доп. - М.: Форум: ИНФРА-М, 2017. - 400 с.
13. Демченко, А.И. Коммерческая логистика: учеб. пособие для практ. занятий/А. И. Демченко; Юж.-Урал. гос. ун-т, Фак. коммерции. -Челябинск: Изд-во ЮУрГУ, 2018. – 317с.
14. Ехлаков, Ю.П. Теоретические основы автоматизированного управления. – Томск: Изд-во Том. гос. ун-та систем управления и радиоэлектроники, 2017. – 337 с.
15. Ехлаков, Ю.П. Функциональные модели управления складским хозяйством// Доклады ТУСУРа, № 2 (18), часть 2, декабрь 2017. – М.: изд. ТУСУРа, 2018. – с. 135-139
16. Заглянов, П. Обнаружение телекоммуникационных атак: теория и практика // Системный администратор. - 2017. - № 10. - С. 48-67.
17. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы: (Сборник): ГОСТ 34.003 - 90, РД 50 - 680 - 88, РД 50 - 682 - ГОСТ 34. 201 - 89 - ГОСТ 34.602.89 – М.: Изд-во стандартов, 2019. – 213с.
18. Информационные системы в экономике: учеб. для студентов ВУЗов, обучающихся по специальностям «Финансы и кредит», «Бухгалтерский учет, анализ и аудит» / Под. ред. Г.А. Титоренко. – 2-е изд., перераб. и доп. – М.: ЮНИТИ-ДАНА, 2017. – 463 с
19. Кузнецов, С.Д. Проектирование и разработка корпоративных информационных систем. Центр информационных технологий. М.: МГУ, 2017. – 412с.
20. Логистика: учеб. пособие / Под ред. Б.А. Аникина. – М.: ИНФРА-М, 2017. – 327 с.
21. Океанова, З. К. Основы экономической теории: учеб. пособие / З.К.

Океанова. - 3-е изд., испр. и доп. - М.: ФОРУМ: ИНФРА-М, 2016. - 288 с..

22. Орлов, С. А. Технологии разработки программного обеспечения, 2-е изд. – СПб.: «Питер», 2018. – 387с.

23. Рагулин, П.Г. Информационные технологии: учеб. — Владивосток: ТИДОТ Дальневост. ун-та, 2017. - 208 с.

24. Рудаков, А. В. Технология разработки программных продуктов. Практикум: учеб. пособие для студ. учреждений сред. проф. образования / А. В. Рудаков, Г. Н. Федорова. — 6-е изд., стер. — М.: Издательский центр «Академия»; 2016. — 192 с.

25. Семакин, И.Г., Шестаков, А.П. Основы программирования: учеб. — М.: Мастерство, 2018. - 432 с.

26. Силич, В.А. Содержательные модели систем и их использование при проектировании АСУ. — Томск: Изд-во ТГУ, 2018. – 451с.

27. Фаронов, В. В. Программирование баз данных в Delphi 7, СПб. : Питер, 2018. – 374с.

28. Шаньгин, В. Ф. Защита компьютерной информации. Эффективные методы и средства. - М.: ДМК Пресс, 2017. - 544 с.

29. Якубайтис, Э. А. Информационные сети и системы - М.: Финансы и статистика, 2018. – 420с.

30. Microsoft Office Access. Шаг за шагом + приложение (пер. с англ.). С. Ламберт, М.Доу : Из-во ЭКОМ, 2018. – 412с.

31. Балтер, Э. Microsoft Office Access. Профессиональное программирование / Э. Балтер. - М.: Вильямс, 2017. – 1296 с.

Исходный текст программы

Программный модуль UComp.dpr

```

program UComp;

uses
  Forms,
  UnMain in 'UnMain.pas' {FmMain},
  UnDM in 'UnDM.pas' {DM: TDataModule},
  UnTree in 'UnTree.pas',
  UnDep in 'UnDep.pas' {FmDep},
  UnMan in 'UnMan.pas' {FmMan},
  UnParam in 'UnParam.pas' {FmParam},
  UnCountry in 'UnCountry.pas' {FmCountry},
  UnManuf in 'UnManuf.pas' {FmManuf},
  UnDevice in 'UnDevice.pas' {FmDevice},
  UnHistory in 'UnHistory.pas' {FmHistory},
  UnUpdate in 'UnUpdate.pas' {FmUpdate},
  UnGDate in 'UnGDate.pas' {FmGDate},
  UnInt in 'UnInt.pas' {FmInt};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TFmMain, FmMain);
  Application.CreateForm(TDM, DM);
  Application.CreateForm(TFmDep, FmDep);
  Application.CreateForm(TFmMan, FmMan);
  Application.CreateForm(TFmParam, FmParam);
  Application.CreateForm(TFmCountry, FmCountry);
  Application.CreateForm(TFmManuf, FmManuf);
  Application.CreateForm(TFmDevice, FmDevice);
  Application.CreateForm(TFmHistory, FmHistory);
  Application.CreateForm(TFmUpdate, FmUpdate);
  Application.CreateForm(TFmGDate, FmGDate);
  Application.CreateForm(TFmInt, FmInt);
  Application.Run;
end.

```

Программный модуль UnCountry.pas

```

unit UnCountry;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, DBCtrls, Grids, DBGrids, ExtCtrls;

type
  TFmCountry = class(TForm)
    Panel1: TPanel;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Panel2: TPanel;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FmCountry: TFmCountry;

implementation
uses UnDM;
{$R *.dfm}

procedure TFmCountry.BitBtn1Click(Sender: TObject);
begin

```

```
DM.TbCountry.Close;
DM.TbCountry.Open;
end;

end.
```

Программный модуль UnDep.pas

```
unit UnDep;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Mask, DBCtrls, Buttons;

type
  TFmDep = class(TForm)
    DBEdit1: TDBEdit;
    DBLookupComboBox1: TDBLookupComboBox;
    DBEdit2: TDBEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FmDep: TFmDep;

implementation
uses UnDM, UnTree, UnMain;
{$R *.dfm}

procedure TFmDep.BitBtn1Click(Sender: TObject);
begin
  if DBEdit1.Text = '' then
    begin
      ShowMessage('Вы не ввели название отдела');
      exit;
    end;

  if DBLookupComboBox1.Text = '' then
    begin
      ShowMessage('Вы не выбрали начальника отдела');
      exit;
    end;

  if DM.TbDep.Modified then DM.TbDep.Post;
  BuildCompTree;
  if FmMain.TreeView1.Items.Count <> 0 then
    FmMain.TreeView1.Items.Item[0].Selected := true;
  close;
end;

procedure TFmDep.BitBtn2Click(Sender: TObject);
begin
  close;
end;

end.
```

Программный модуль UnDevice.pas

```
unit UnDevice;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Buttons, Grids, DBGrids, DBCtrls;
```

```

type
TFmDevice = class(TForm)
  Panel1: TPanel;
  Panel2: TPanel;
  Panel3: TPanel;
  Panel4: TPanel;
  Panel5: TPanel;
  Panel6: TPanel;
  Panel7: TPanel;
  Panel8: TPanel;
  BitBtn1: TBitBtn;
  DBNavigator1: TDBNavigator;
  DBGrid1: TDBGrid;
  DBNavigator2: TDBNavigator;
  DBGrid2: TDBGrid;
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FmDevice: TFmDevice;

implementation
uses UnDM;
{$R *.dfm}

end.

```

Программный модуль UnDM.pas

```

unit UnDM;

interface

uses
  SysUtils, Classes, DB, ADODB;

type
TDM = class(TDataModule)
  ADOC: TADOConnection;
  TbDep: TADOTable;
  DSDep: TDataSource;
  TbARM: TADOTable;
  DSARM: TDataSource;
  TbMan: TADOTable;
  DSMan: TDataSource;
  TbDepid: TAutoIncField;
  TbDepDepName: TWideStringField;
  TbDepchief: TIntegerField;
  TbDepNote: TWideStringField;
  TbARMid: TAutoIncField;
  TbARMid_dep: TIntegerField;
  TbARMcomp_name: TWideStringField;
  TbARMnet_name: TWideStringField;
  TbARMinv: TWideStringField;
  TbARMip: TWideStringField;
  TbARMchief: TIntegerField;
  TbManid: TAutoIncField;
  TbManFIO: TWideStringField;
  TbManBDay: TDateTimeField;
  TbManTabNum: TWideStringField;
  TbManPassport: TWideStringField;
  TbManNote: TWideStringField;
  TbARMNote: TWideStringField;
  TbCountry: TADOTable;
  DSCountry: TDataSource;
  TbManuf: TADOTable;
  DSManuf: TDataSource;
  TbSoft: TADOTable;
  DSSoft: TDataSource;
  TbSet: TADOTable;
  DSSet: TDataSource;
  TbDevice: TADOTable;
  DSDevice: TDataSource;
  TbParam: TADOTable;
  DSParam: TDataSource;
  TbNote: TADOTable;

```


DSNote: TDataSource;
 TbCountryid: TAutoIncField;
 TbCountryCountry: TWideStringField;
 TbManufid: TAutoIncField;
 TbManufManuf: TWideStringField;
 TbManufcountry: TIntegerField;
 TbManufCoun: TStringField;
 TbSoftid: TAutoIncField;
 TbSoftid_arm: TIntegerField;
 TbSoftSoft: TWideStringField;
 TbSoftmanuf: TIntegerField;
 TbSoftLicence: TWideStringField;
 TbSetid: TAutoIncField;
 TbSetid_arm: TIntegerField;
 TbSetdevice: TIntegerField;
 TbSetin_box: TBooleanField;
 TbSetsdate: TDateTimeField;
 TbSetgdate: TDateTimeField;
 TbParamid: TAutoIncField;
 TbParamParamName: TWideStringField;
 TbDeviceid: TAutoIncField;
 TbDevedevice: TWideStringField;
 TbDevicemanuf: TIntegerField;
 TbDeviceNote: TWideStringField;
 TbNoteid: TAutoIncField;
 TbNoteid_dev: TIntegerField;
 TbNotenpar: TIntegerField;
 TbNotenValue: TWideStringField;
 TbNoteparametr: TStringField;
 TbNotenNote: TWideStringField;
 TbDevedManuf: TStringField;
 TbHistory: TADOTable;
 TbUpdate: TADOTable;
 TbUpdateid: TAutoIncField;
 TbUpdateid_soft: TIntegerField;
 TbUpdateUNum: TWideStringField;
 TbUpdateNote: TWideStringField;
 DSHistory: TDataSource;
 DSUpdate: TDataSource;
 TbHistoryid: TAutoIncField;
 TbHistoryid_set: TIntegerField;
 TbHistoryMan: TIntegerField;
 TbHistoryNote: TWideStringField;
 TbHistoryHMan: TStringField;
 TbSoftSManuf: TStringField;
 TbSetCDev: TStringField;
 TbHistoryHDate: TDateTimeField;
 TbUpdateUDate: TDateTimeField;
 QIp: TADOQuery;
 QIpcomp_name: TWideStringField;
 QIpnet_name: TWideStringField;
 QIpinv: TWideStringField;
 QIpip: TWideStringField;
 QIpFIO: TWideStringField;
 QIpDepName: TWideStringField;
 TbDepDChief: TStringField;
 TbARMachief: TStringField;
 QGar: TADOQuery;
 QUp: TADOQuery;
 QGarcomp_name: TWideStringField;
 QGarnet_name: TWideStringField;
 QGarFIO: TWideStringField;
 QGarsdate: TDateTimeField;
 QGargdate: TDateTimeField;
 QGarDepName: TWideStringField;
 QGarinv: TWideStringField;
 QGardevice: TWideStringField;
 QUpcomp_name: TWideStringField;
 QUpnet_name: TWideStringField;
 QUpFIO: TWideStringField;
 QUpDepName: TWideStringField;
 QUpinv: TWideStringField;
 QUpSoft: TWideStringField;
 QUpUDate: TDateTimeField;
 QUpUNum: TWideStringField;
 QUpLicence: TWideStringField;
 TbMat: TADOTable;
 TbMatid: TAutoIncField;
 TbMatid_arm: TIntegerField;

```

TbMatmat: TWideStringField;
TbMatmdate: TDateTimeField;
TbMatmnote: TWideStringField;
DSMat: TDataSource;
QMat: TADOQuery;
QMatcomp_name: TWideStringField;
QMatinv: TWideStringField;
QMatFIO: TWideStringField;
QMatmat: TWideStringField;
QMatmdate: TDateTimeField;
QMatDepName: TWideStringField;
QMatnet_name: TWideStringField;
private
  { Private declarations }
public
  { Public declarations }
end;

var
  DM: TDM;

implementation

{$R *.dfm}

end.

```

Программный модуль UnGDate.pas

```

unit UnGDate;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ComCtrls, ComObj;

type
  TFmGDate = class(TForm)
    DateTimePicker1: TDateTimePicker;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FmGDate: TFmGDate;

implementation

uses UnDM;
{$R *.dfm}

procedure TFmGDate.BitBtn1Click(Sender: TObject);
var Excel: Variant;
    WorkbookName: string;
    i,k,PosInc : integer;
begin
  if FmGDate.Caption = 'Список устройств, находящихся на гарантии на дату' then
  begin
    DM.QGar.Active := false;
    DM.QGar.Parameters[0].Value := DateTimePicker1.Date;
    DM.QGar.Open;
    DM.QGar.First;

    k := 8;
    PosInc := k;
    //Информация по Гарантии

    WorkbookName := GetCurrentDir + '\Гарантийные устройства.xlt';
    Excel := CreateOleObject('Excel.Application');

    try
      Excel.Workbooks.Open(WorkbookName);
      //Экспорт данных из таблиц БД:
      Excel.WorkSheets[1].Select;
      Excel.Cells(4, 7) := DateToStr (DateTimePicker1.Date);
    finally
      Excel.Quit;
    end;
  end;
end;

```

```

for i := 0 to DM.QGar.RecordCount-1 do

begin

if i <> DM.QGar.RecordCount-1 then
begin

Excel.Rows.Item[PosInc+1].Select;
Excel.Selection.Insert;
Excel.Rows.Item[PosInc].Select;
Excel.Selection.Copy;
Excel.Rows.Item[PosInc+1].Select;
Excel.ActiveSheet.Paste;
Excel.Application.CutCopyMode := False;
end;

Excel.Cells(k, 1) := IntToStr (i+1);
Excel.Cells(k, 2) := DM.QGar.FieldByName('device').AsString;
Excel.Cells(k, 3) := DM.QGar.FieldByName('sdate').AsString;
Excel.Cells(k, 4) := DM.QGar.FieldByName('gdate').AsString;
Excel.Cells(k, 5) := DM.QGar.FieldByName('comp_name').AsString;
Excel.Cells(k, 6) := DM.QGar.FieldByName('net_name').AsString;
Excel.Cells(k, 7) := DM.QGar.FieldByName('inv').AsString;
Excel.Cells(k, 8) := DM.QGar.FieldByName('FIO').AsString;
Excel.Cells(k, 9) := DM.QGar.FieldByName('Depname').AsString;
k := k + 1;
DM.QGar.Next;

end;
Excel.WorkSheets[1].Select;
Excel.Visible := True;
except
Excel.Quit;
end;
end
else
begin
DM.QUp.Active := false;
DM.QUp.Parameters[0].Value := DateTimePicker1.Date;
DM.QUp.Open;
DM.QUp.First;

k := 7;
PosInc := k;
//Информация по Обновлениям

WorkbookName := GetCurrentDir + '\Обновления.xlt';
Excel := CreateOleObject('Excel.Application');

try
Excel.Workbooks.Open(WorkbookName);
//Экспорт данных из таблиц БД:
Excel.WorkSheets[1].Select;
Excel.Cells(4, 7) := DateToStr (DateTimePicker1.Date);
for i := 0 to DM.QUp.RecordCount-1 do

begin

if i <> DM.QUp.RecordCount-1 then
begin

Excel.Rows.Item[PosInc+1].Select;
Excel.Selection.Insert;
Excel.Rows.Item[PosInc].Select;
Excel.Selection.Copy;
Excel.Rows.Item[PosInc+1].Select;
Excel.ActiveSheet.Paste;
Excel.Application.CutCopyMode := False;
end;

Excel.Cells(k, 1) := IntToStr (i+1);
Excel.Cells(k, 2) := DM.QUp.FieldByName('soft').AsString;
Excel.Cells(k, 3) := DM.QUp.FieldByName('Licence').AsString;
Excel.Cells(k, 4) := DM.QUp.FieldByName('UNum').AsString + '/' + DM.QUp.FieldByName('UDate').AsString;
Excel.Cells(k, 5) := DM.QUp.FieldByName('comp_name').AsString;
Excel.Cells(k, 6) := DM.QUp.FieldByName('net_name').AsString;
Excel.Cells(k, 7) := DM.QUp.FieldByName('inv').AsString;

```

```

Excel.Cells(k, 8) := DM.QUp.FieldByName('FIO').AsString;
Excel.Cells(k, 9) := DM.QUp.FieldByName('Depname').AsString;
k := k + 1;
DM.QUp.Next;

end;
Excel.WorkSheets[1].Select;
Excel.Visible := True;
except
Excel.Quit;
end;
end;
end;

end.

```

Программный модуль UnHistory.pas

```

unit UnHistory;

interface

uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Mask, DBCtrls, Buttons, Grids, DBGrids, ExtCtrls,
ComCtrls, ComObj;

type
TFmHistory = class(TForm)
Panel1: TPanel;
DBGrid1: TDBGrid;
DBNavigator1: TDBNavigator;
Panel2: TPanel;
BitBtn1: TBitBtn;
Panel3: TPanel;
Label1: TLabel;
DBEdit1: TDBEdit;
DBEdit2: TDBEdit;
DBEdit3: TDBEdit;
Label2: TLabel;
Label3: TLabel;
DateTimePicker1: TDateTimePicker;
procedure DBGrid1ColExit(Sender: TObject);
procedure DBGrid1KeyPress(Sender: TObject; var Key: Char);
procedure DateTimePicker1Change(Sender: TObject);
procedure DateTimePicker1DropDown(Sender: TObject);
procedure DBGrid1DrawColumnCell(Sender: TObject; const Rect: TRect;
DataCol: Integer; Column: TColumn; State: TGridDrawState);
private
{ Private declarations }
public
{ Public declarations }
end;

var
FmHistory: TFmHistory;

implementation
uses UnDM, DB;
{$R *.dfm}

procedure TFmHistory.DBGrid1ColExit(Sender: TObject);
begin
if DBGrid1.SelectedField.FieldName = 'HDate' then
DateTimePicker1.Visible := False;
end;

procedure TFmHistory.DBGrid1KeyPress(Sender: TObject; var Key: Char);
begin
if (key = Chr(9)) then Exit; // код табуляции
if (DBGrid1.SelectedField.FieldName = 'HDate') then
begin
DateTimePicker1.Visible := true;
DateTimePicker1.SetFocus;
SendMessage(DateTimePicker1.Handle, WM_Char, word(Key), 0);
end;
end;
end;

```

```

procedure TFmHistory.DateTimePicker1Change(Sender: TObject);
begin
  if DBGrid1.DataSource.State in [dsEdit, dsInsert] then
    DM.TbHistory.FieldByName('HDate').Value := DateTimePicker1.Date;
end;

procedure TFmHistory.DateTimePicker1DropDown(Sender: TObject);
begin
  DBGrid1.DataSource.Edit;
end;

procedure TFmHistory.DBGrid1DrawColumnCell(Sender: TObject);
const Rect: TRect; DataCol: Integer; Column: TColumn;
State: TGridDrawState);
begin
  if (gdFocused in State) then
  begin
    if (Column.Field.FieldName = 'HDate') then
      with DateTimePicker1 do
      begin
        if DM.TbHistory.FieldByName('HDate').AsString <> '' then DateTimePicker1.Date:=DM.TbHistory.FieldByName('HDate').Value;
        Left := Rect.Left + DBGrid1.Left + 1;
        Top := Rect.Top + DBGrid1.Top + 1;
        Width := Rect.Right - Rect.Left + 2;
        Width := Rect.Right - Rect.Left + 2;
        Height := Rect.Bottom - Rect.Top + 2;
        Visible := True;
      end;
    end;
  end;
end;

end.

```

Программный модуль UnInt.pas

```

unit UnInt;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ComCtrls, ComObj;

type
  TFmInt = class(TForm)
    DateTimePicker1: TDateTimePicker;
    Label1: TLabel;
    Label2: TLabel;
    DateTimePicker2: TDateTimePicker;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FmInt: TFmInt;

implementation
uses UnDM;
{$R *.dfm}

procedure TFmInt.BitBtn1Click(Sender: TObject);
var Excel: Variant;
    WorkbookName: string;
    i,k,PosInc : integer;
begin
  DM.QMat.Active := false;
  DM.QMat.Parameters[0].Value := DateTimePicker1.Date;
  DM.QMat.Parameters[1].Value := DateTimePicker2.Date;
  DM.QMat.Open;
  DM.QMat.First;

  k := 7;
  PosInc := k;
  //Информация по Гарантии

```

```

WorkbookName := GetCurrentDir + '\Материалы.xlt';
Excel := CreateOleObject('Excel.Application');

try
Excel.Workbooks.Open(WorkbookName);
//Экспорт данных из таблиц БД:
Excel.WorkSheets[1].Select;
Excel.Cells(4, 5) := DateToStr (DateTimePicker1.Date);
Excel.Cells(4, 7) := DateToStr (DateTimePicker2.Date);
for i := 0 to DM.QMat.RecordCount-1 do

begin

if i <> DM.QMat.RecordCount-1 then
begin

Excel.Rows.Item[PosInc+1].Select;
Excel.Selection.Insert;
Excel.Rows.Item[PosInc].Select;
Excel.Selection.Copy;
Excel.Rows.Item[PosInc+1].Select;
Excel.ActiveSheet.Paste;
Excel.Application.CutCopyMode := False;
end;

Excel.Cells(k, 1) := IntToStr (i+1);
Excel.Cells(k, 2) := DM.QMat.FieldName('mat').AsString;
Excel.Cells(k, 3) := DateToStr(DM.QMat.FieldName('mdate').AsDateTime);
Excel.Cells(k, 4) := DM.QMat.FieldName('comp_name').AsString;
Excel.Cells(k, 5) := DM.QMat.FieldName('net_name').AsString;
Excel.Cells(k, 6) := DM.QMat.FieldName('inv').AsString;
Excel.Cells(k, 7) := DM.QMat.FieldName('FIO').AsString;
Excel.Cells(k, 8) := DM.QMat.FieldName('Depname').AsString;
k := k + 1;
DM.QMat.Next;

end;
Excel.WorkSheets[1].Select;
Excel.Visible := True;
except
Excel.Quit;
end;
end;
end.

```

Программный модуль UnMan.pas

```

unit UnMan;

interface

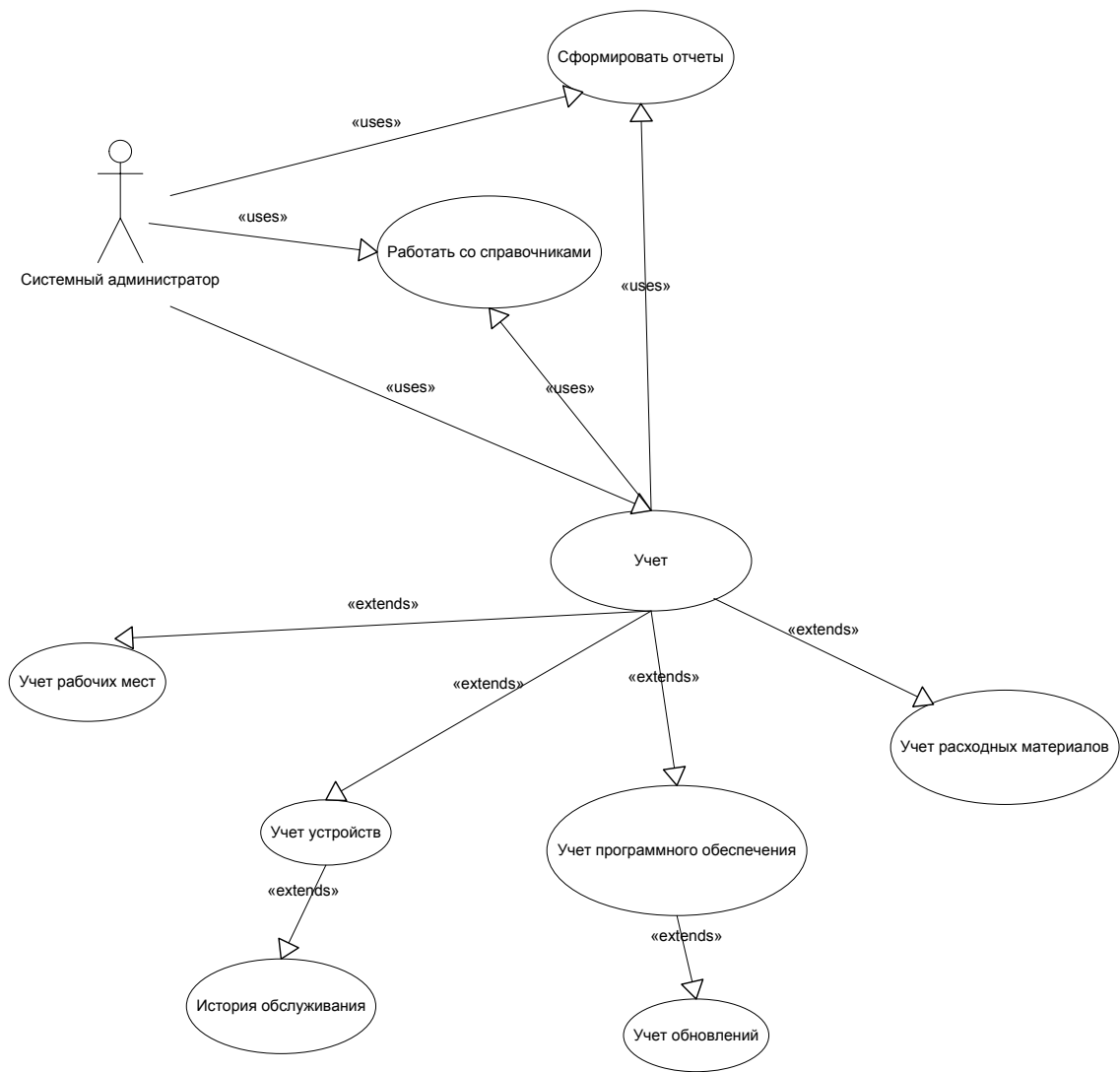
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons, DBCtrls, Grids, DBGrids, ExtCtrls;

type
TFmMan = class(TForm)
Panel1: TPanel;
Panel2: TPanel;
DBGrid1: TDBGrid;
DBNavigator1: TDBNavigator;
BitBtn1: TBitBtn;
private
{ Private declarations }
public
{ Public declarations }
end;
var
FmMan: TFmMan;
implementation
uses UnDM;
{$R *.dfm}

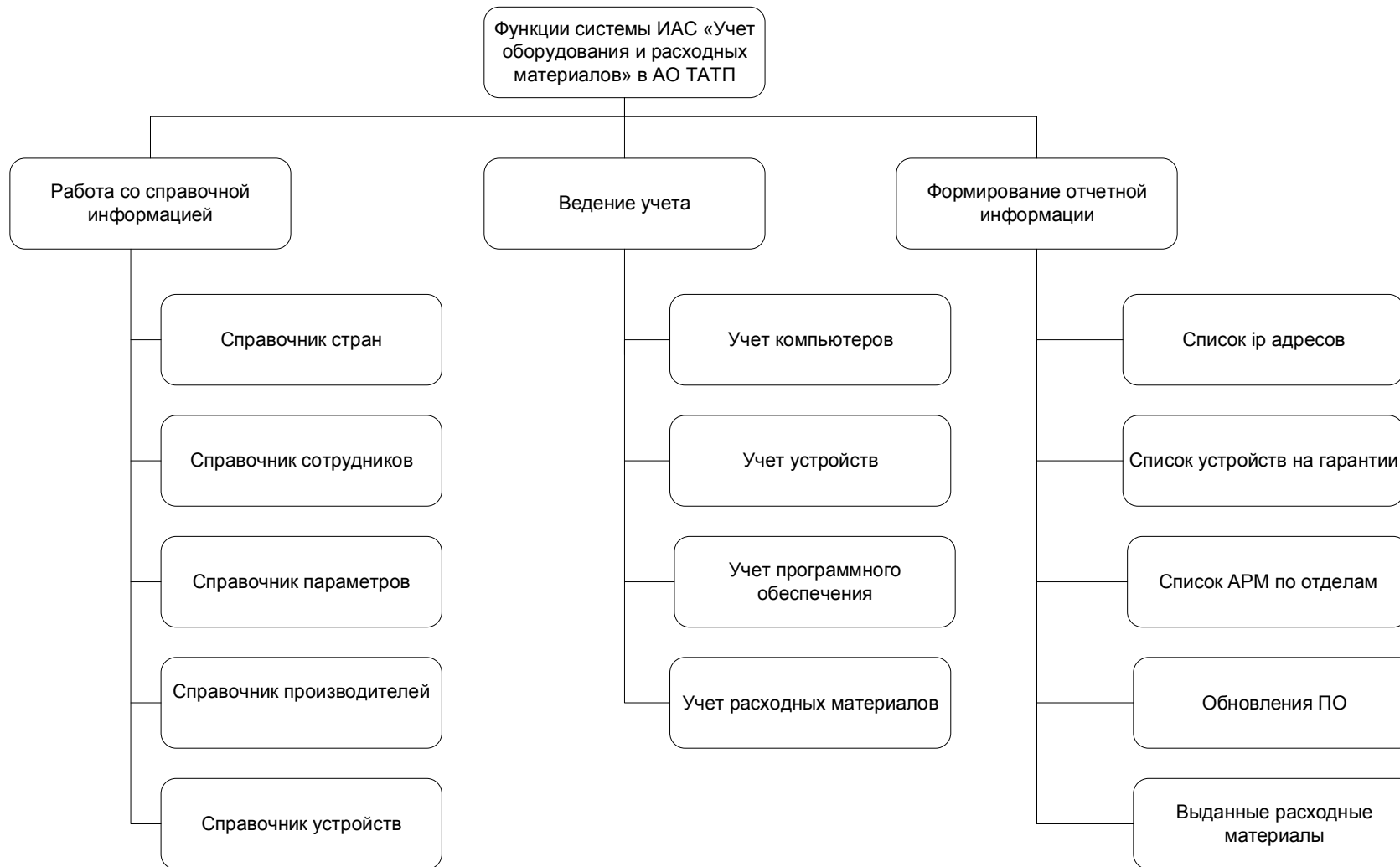
end.

```

Диаграмма прецедентов



Дерево функций



Структура меню программы

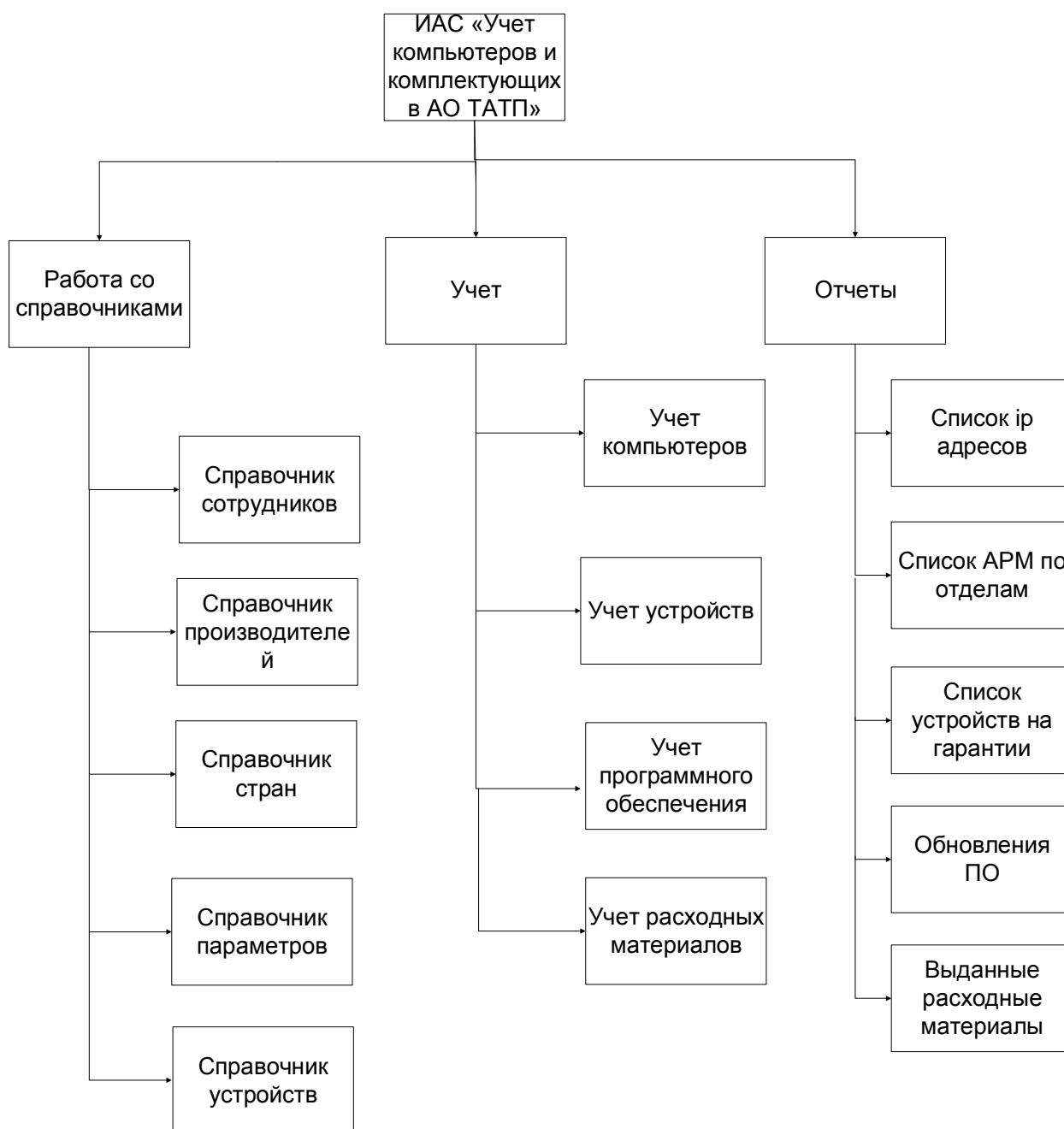
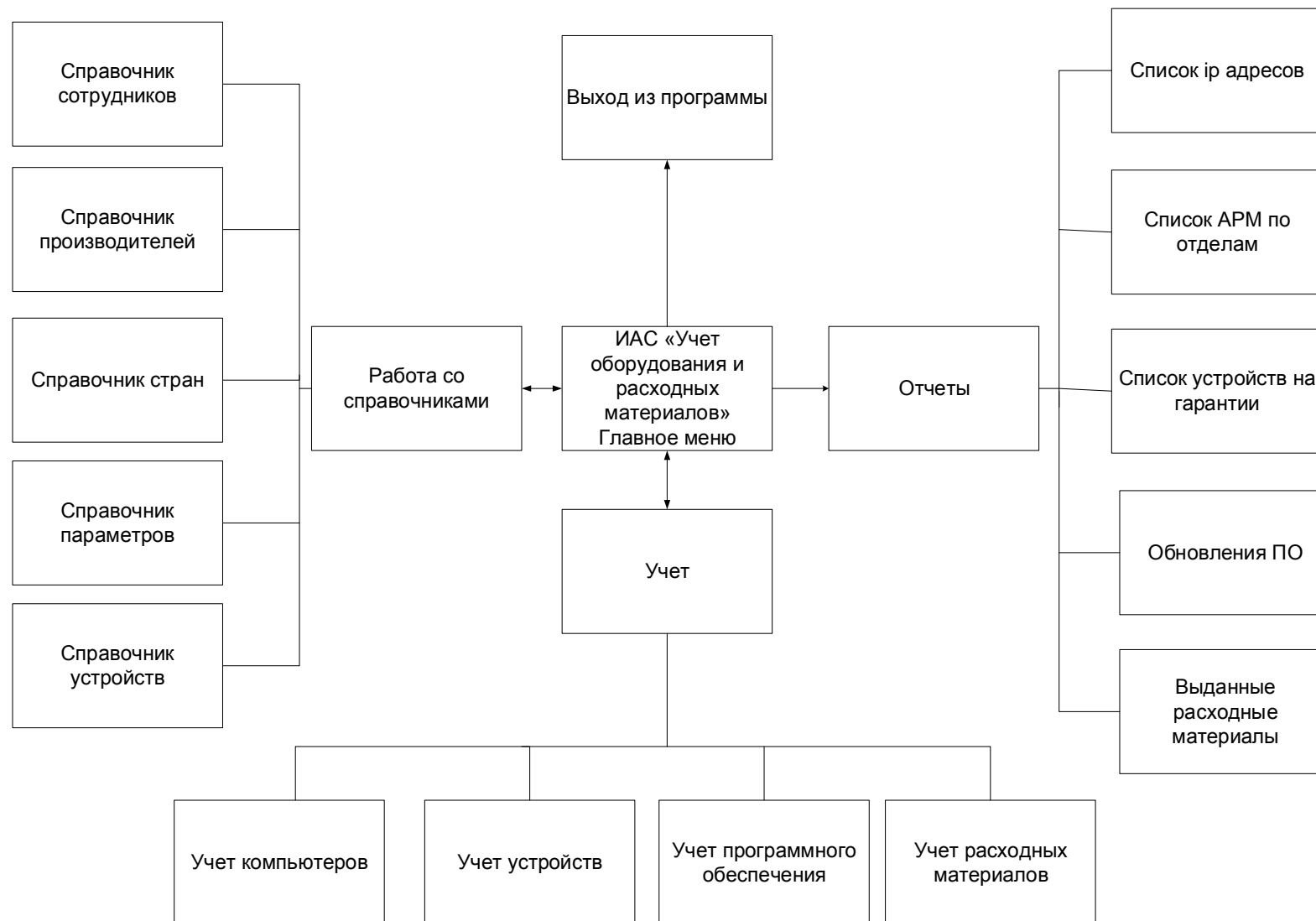


Схема модулей



Сценарии диалогов программы



Контекстная диаграмма системы

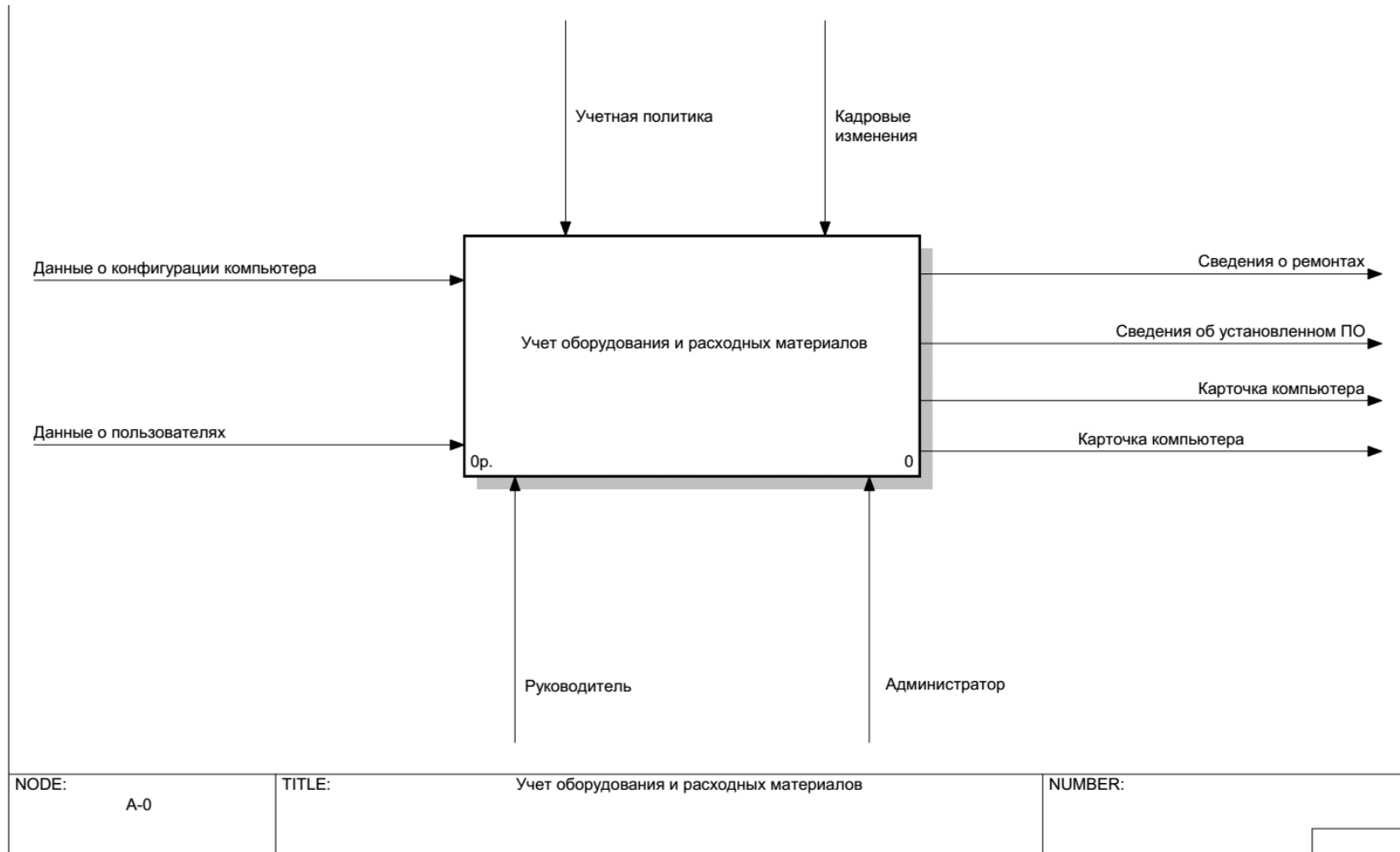
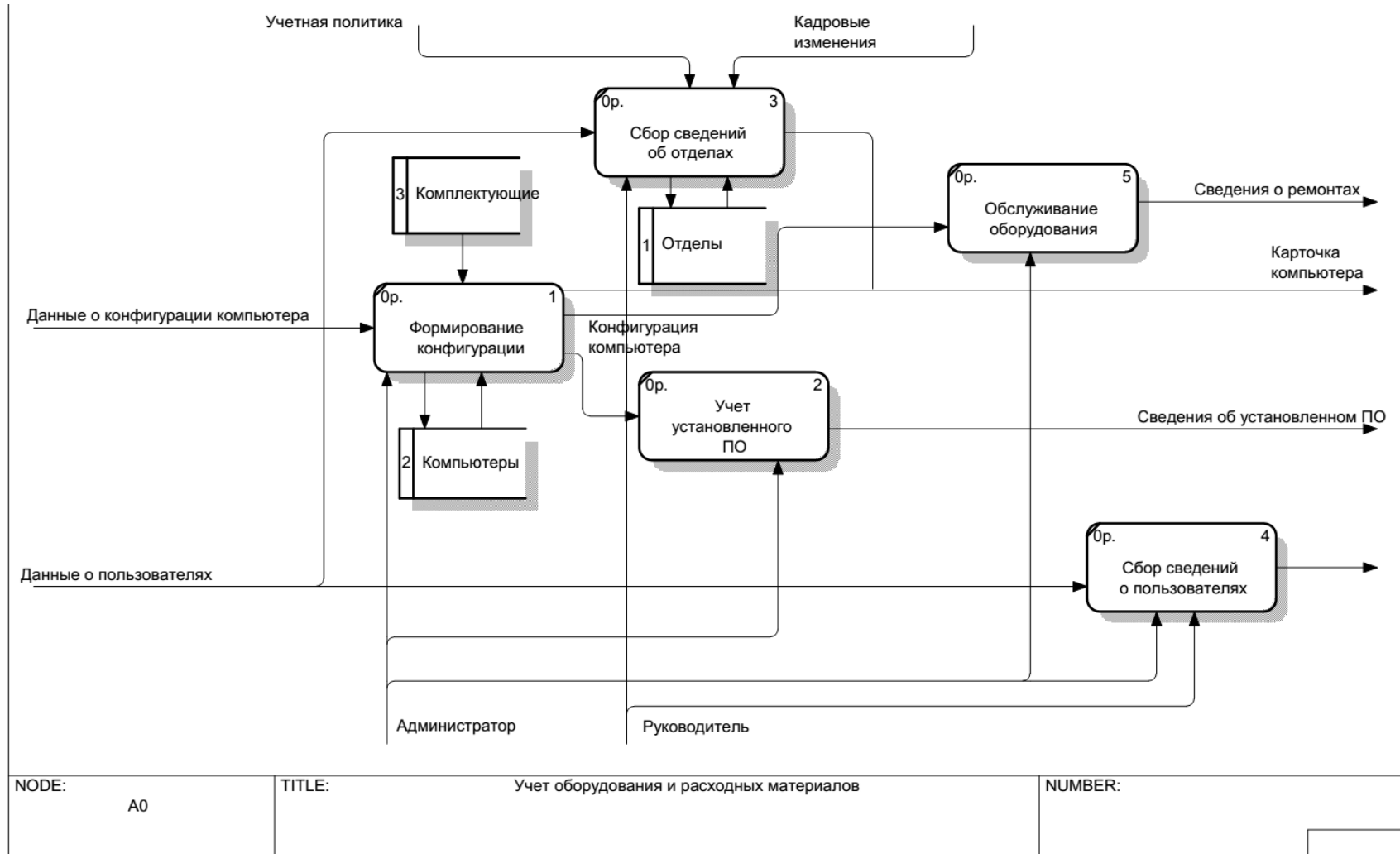


Диаграмма декомпозиции системы



Внешний вид программы

