

**Образец содержит дипломную работу,
презентацию и доклад (речь на
защиту)**

**РАЗРАБОТКА ИС УЧЕТА И КОНТРОЛЯ ВЫПОЛНЕНИЯ
ЗАЯВОК НА ТЕХНИЧЕСКОЕ И ПРОГРАММНОЕ ОБСЛУЖИВАНИЕ
КОМПЬЮТЕРНОЙ ТЕХНИКИ**

ОГЛАВЛЕНИЕ

Введение	3
Глава 1. Аналитическая часть процесса разработки автоматизированной системы «Автоматизация бизнес-процессов Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск»».....	6
Глава 2. Разработка автоматизированной системы «Автоматизация бизнес-процессов Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск»».....	20
Глава 3 Экономическое обоснование рентабельности разработки автоматизированной системы «Автоматизация бизнес-процессов Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск»».....	42
Заключение.....	48
Литература.....	49
Приложение 1. Скрипт для создания базы данных.....	51
Приложение 2. Листинг программы.....	68
Приложение 3. Техническое задание.....	120

ВВЕДЕНИЕ

Не последнее место в автоматизации предприятия занимает организации подразделения поддержки и ремонта технологического оборудования и агрегатов, задачей которого является поддержание, обслуживание и ремонт различного оборудования [6].

Одной из основных задач такого подразделения является обработка заявок пользователей, связанных с выходом из строя и/или возникновением неисправностей в станках и другом оборудовании цехов. Выполнение обязанностей по устранению неисправностей возложено на инженерно-технический персонал подразделения, а управление данным процессом (обработка заявок и назначение исполнителя), на диспетчера [7]. На сегодняшний момент, данный бизнес-процесс не автоматизирован, в связи, с чем наблюдается недостаточная эффективность и низкая оперативность выполнения означенных задач. Таким образом, можно сделать вывод, что необходима разработка усовершенствованного бизнес-процесса управления инженерно-техническим персоналом и распределения ресурсов на выполнение заявок пользователей. По итогам проведённого анализа рынка полностью удовлетворяющих разработке сторонних производителей обнаружено не было.

Целью данной работы является разработка системы по контролю выполнения заявок на техническое и программное обслуживание компьютерной техники.

Для достижения цели необходимо решить следующие задачи. Изучить предметную область Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск» [10]. Изучить бизнес-процесс приёма заявок, выявить его слабые стороны. Выделить задачи, требующие разработки бизнес-процессов, поскольку они имеют слабые стороны, которые были выявлены во время изучения бизнес-процессов приёма заявок Муниципального ресурсного центра по работе с одаренными детьми, является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск». Обосновать необходимость

использования вычислительной техники в автоматизации бизнес-процесов, поскольку они имеют слабые стороны, которые были выявлены во время изучения бизнес-процессов приёма заявок Муниципального ресурсного центра по работе с одаренными детьми, является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск». Разработать методику автоматизации бизнес-процессов, имеющих слабые стороны, которые были выявлены во время изучения бизнес-процессов приёма заявок Муниципального ресурсного центра по работе с одаренными детьми, является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск». Выбрать средства решения задачи автоматизации бизнес-процессов, имеющих слабые стороны, которые были выявлены во время изучения бизнес-процессов приёма заявок Муниципального ресурсного центра по работе с одаренными детьми, является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск». Разработать техническое задание на создание автоматизированной системы «Автоматизация бизнес-процессов Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск»». Разработать базу данных автоматизированной системы «Автоматизация бизнес-процессов Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск»». Разработать программный модуль автоматизированной системы «Автоматизация бизнес-процессов Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск»». Разработать интерфейс программного модуля автоматизированной системы «Автоматизация бизнес-процессов Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск»». Разработать техническую документацию по работе с программным продуктом «Автоматизация

бизнес-процессов Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск»».

В качестве субъекта решения задачи будет выступать среда разработки бизнес-процессов BP Win Process modeler, интегрированная среда разработки Delphi 7, СУБД MS Access [1].

В качестве объекта задачи будет выступать. Поток заявок от пользователей. Существующая традиционная система журнального учёта.

Гипотеза состоит в том, что с помощью разработки бизнес-процессов управления и современных информационно-технических удастся сократить трудовые затраты и вместе с тем повысить надёжность, скорость и оперативность работы инженерно-технической службы.

ГЛАВА 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ ПРОЦЕССА РАЗРАБОТКИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ «АВТОМАТИЗАЦИЯ БИЗНЕС- ПРОЦЕССОВ МУНИЦИПАЛЬНОГО РЕСУРСНОГО ЦЕНТРА ПО РАБОТЕ С ОДАРЕННЫМИ ДЕТЬМИ ЯВЛЯЕТСЯ СОВЕРШЕНСТВОВАНИЕ РАБОТЫ ПО ВЫЯВЛЕНИЮ, ПОДДЕРЖКЕ И РАЗВИТИЮ ОДАРЕННЫХ И ТАЛАНТЛИВЫХ ДЕТЕЙ НА ТЕРРИТОРИИ МО ГО «УСИНСК»».

1.1 Вступление

Моя практика проходила в Муниципальном ресурсном центре по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск». Основной целью деятельности Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск» [9][10][12].

Режим работы МАУДО "ЦДОД" г. Усинска: Понедельник-Воскресенье
08:00-20:00

Телефоны. Директор, приёмная: 28-5-82. Вахта: 2-87-37.

Почтовый адрес: 169710, Республика Коми, г. Усинск, ул. Мира, 11-а.

Адрес электронной почты: cdod-usinsk@mail.ru.

Официальный сайт: www.cdod-usinsk.ucoz.ru .

Учредитель МАУДО «ЦДОД» г. Усинска: Управление образования администрации муниципального образования городского округа «Усинск» [9];

1.2 Анализ предметной области и выделение задачи автоматизации

Работа службы технической поддержки сводится к приёму заявок и сообщений от производственного персонала о возникновении проблем с станочным парком и последующее их решение.

На текущий момент модель обеспечения процесса распределения заявок состоит из выполнения следующих процессов. При возникновении неисправности сотрудник звонит в службу и сообщает диспетчеру (руководителю, менеджеру) место, оборудование, и признаки неисправности. Диспетчер принимает заявку и передаёт её на исполнение соответствующему техническому специалисту. Технический специалист устраняет неисправность. Сотрудник сообщает руководителю, что неисправность устранена. Диспетчер ставит отметку у себя в журнале об устранении неисправности.

В рамках данной работы руководством центра было поручено изучить систему поддержки пользователей, определить, возможно, ли её усовершенствование и целесообразность решения данной задачи, и, в случае положительного ответа, произвести необходимые мероприятия.

Поддержка пользователей сводится к приёму заявок и сообщений от пользователей о возникновении проблем с компьютерной техникой и последующее их решение [5].

На текущий момент модель обеспечения процесса распределения заявок состоит из выполнения следующих процессов. При возникновении неисправности пользователь звонит в службу и сообщает оператору место, оборудование, и признаки неисправности; оператор принимает заявку и передаёт её на исполнение соответствующему технику; техник устраняет неисправность; пользователь сообщает оператору, что неисправность устранена; оператор ставит отметку у себя в журнале об устранении неисправности.

Схема выполнения данной задачи представлена на рисунке (1.2, 1.3) (Схема выполнена согласно стандарту IDF0) [1][8]:

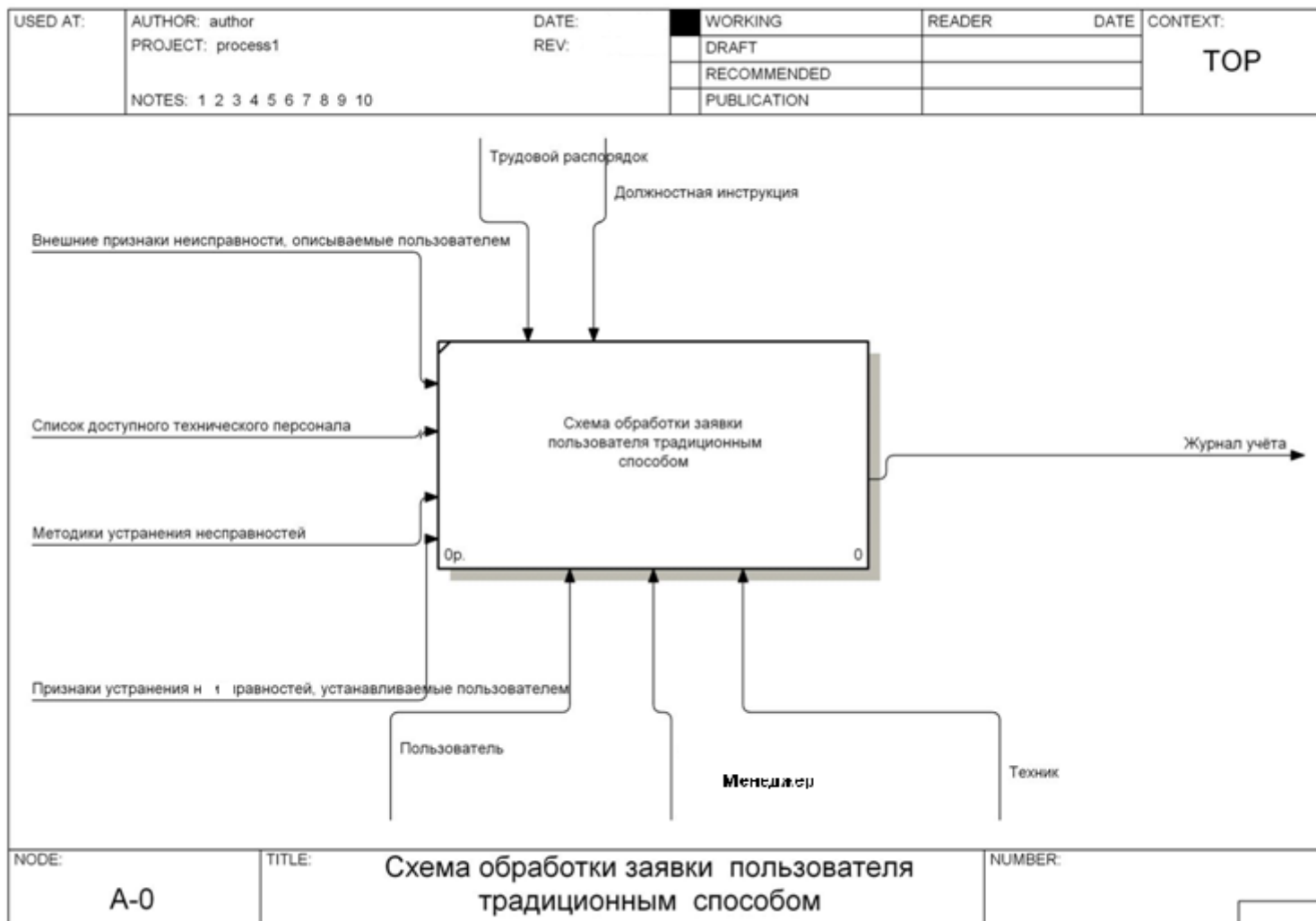


Рисунок 1.2 - Схема обработки заявки традиционным способом

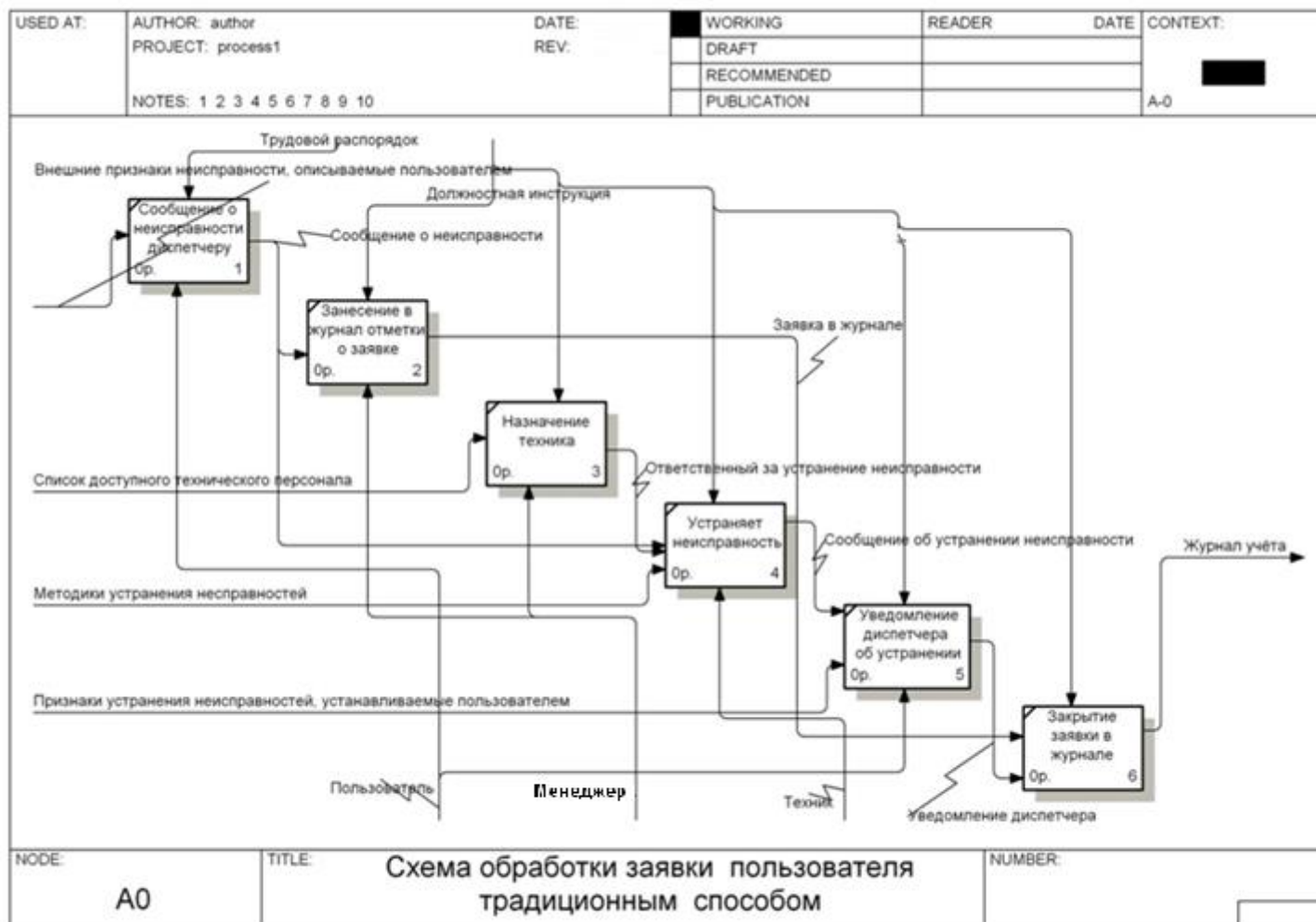


Рисунок 1.3 - Схема обработки заявки традиционным способом (декомпозиция)

Пояснения к схеме. Первый уровень «Сообщение»: Пользователь по телефону сообщает оператору о возникших неисправностях с указанием времени, места возникновения, и внешних проявлений неисправности. Второй уровень «Занесение в журнал»: Оператор заносит в журнал полученную информацию в соответствии и правилами заполнения журнала и собственной должностной инструкцией. Третий уровень «Назначение». Исходя из наличия свободного персонала оператор, руководствуясь должностной инструкцией, назначает ответственного за устранение неисправности. Четвертый уровень «Устранение». Назначенный сотрудник устраняет неисправность. Пятый уровень «Сообщение». Руководствуясь сообщением мастера и наблюдая отсутствие признака неисправности, пользователь должен сообщить оператору об устранении неисправности. Шестой уровень «Занесение в журнал»: Оператор заносит в журнал полученную информацию в соответствии и правилами заполнения журнала и собственной должностной инструкцией.

Вывод: на рисунке схема обработки заявки традиционным способом (декомпозиция).

Данная схема имеет следующие недостатки. Нежелательное многократное преобразование информации. Наличие ненадёжных источников передачи информации (сотрудник). Большие временные затраты на выполнение операции. Большие затраты человеческого труда на выполнение операции. Часто возникающие ошибки, вследствие человеческого фактора. Отсутствие автоматического контроля за выполнение операции. Непрозрачность работы службы, которая не позволяет диспетчеру достаточно эффективно контролировать его работы. Невозможность составления отчётности по работе отдела

Рассмотрев данную задачу, можно сделать вывод, что данная область нуждается в автоматизации, которая должна состоять в уменьшении количества промежуточных пунктов, сообщением сотрудников и устранением неисправности, уменьшении количества преобразований информации, ускорении процесса, уменьшении трудозатрат и исключении человеческого фактора, как главного источника ошибок [7].

Бизнес-процессы, выполняемые персоналом ССР, в процессе осуществления трудовой деятельности, требуют усовершенствования.

Усовершенствование обработки данного комплекса задач можно свести к следующим подзадачам. Усовершенствовать способ заполнения документов, с целью исключить, либо уменьшить возможность совершения ошибки оператором. Усовершенствовать способ ведения журнала. Разработать меры по избеганию необходимости дублирования информации при заполнении документов. Либо, сделать создание дублей, если это необходимо, более простым и удобным. Разработать и реализовать методику поиска необходимой информации в журнале. Разработать и реализовать методики быстрого составления отчётов. Данную проблему можно эффективно решить при помощи информационно-вычислительной техники и соответствующего программного обеспечения [2].

Использование современно вычислительной техники для автоматизации данной задачи прежде всего должно сводиться к сокращению временных затрат на устранение неисправностей, и тем самым сокращение времени простоя, а также к снижению нагрузки на обслуживающий персонал вычислительного центра [17][18].

Автоматизация решения данной задачи при помощи современных компьютерных технологий позволила бы решить следующие проблемы. Малая скорость обработки данных человеком. Недостаточное качество ведения записей в журнале. Большая загруженность человека, заполняющего журнал. Низкая оперативность обработки заявок.

Автоматизация решения данной задачи при помощи современных компьютерных технологий позволила бы достичь следующих положительных моментов. Уменьшить присутствие человеческого фактора. Увеличить скорость и надёжность создания заявок. Повысить скорость обработки. Повысить качество и оперативность работы персонала. Снизить нагрузку на персонал предприятия. Дать возможность для формирования отчётов [20]. Рассмотрим возникающие процессы с точки зрения возникающих ошибок (см. таблицу 1.1):

Таблица 1.1

Возможность возникновения ошибок

№ п/п	Автоматизируемый процесс	Традиционный способ	Способ с применением информационных технологий
1	Заполнение заявки на ремонт	Ошибки относительно редки, основной источник ошибок, человеческий фактор	Ошибки будут несколько реже, поскольку программное обеспечение осуществляет контроль ошибок, основной источник ошибок, человеческий фактор
2.	Создание записи в журнале.	Ошибки редки, но, тем не менее, возникают, основной источник ошибок, человеческий фактор	Ошибки возникать не могут

Продолжение таблицы 2.1

3.	Назначение ответственного за ремонт	Ошибки редки, основной источник ошибок, человеческий фактор	Ошибки возникать не могут
4.	Заполнение отметки о ремонте	Ошибки редки, основной источник ошибок, человеческий фактор	Ошибки не возникают
5.	Ведение журнала необработанных заявок	Ошибки редки, основной источник ошибок, человеческий фактор. С увеличением нагрузки пропорционально увеличивается количество ошибок.	Ошибок практически нет. С увеличением нагрузки количество ошибок возрастает незначительно.

Оценивая данную таблицу, можно сделать вывод, что автоматизированный способ превосходит традиционный по этому параметру.

Оценка качества ввода, хранения, и предоставления информации приведена в следующей таблице (см. табл. 1.2).

Таблица 1.2

Оценка качества ввода, хранения, и предоставления информации

№ п/п	Автоматизируемый процесс	Традиционный способ	Способ с применением информационных технологий
1	Заполнение заявки на ремонт	Качество ввода информации можно считать недостаточным, поскольку ввод сильно зависит от возможностей человека (например, подчерк).	Качество ввода информации высокого качества, поскольку ввод контролируется информационной системой.
2.	Создание записи в журнале.	Качество ввода информации можно считать недостаточным, поскольку ввод сильно зависит от возможностей человека	Отличное качество информации и форма его предоставления.
3.	Назначение ответственного за ремонт	Качество ввода информации не влияет на процесс.	Качество ввода информации высокое.

4.	Заполнение отметки о ремонте	Качество ввода информации можно считать недостаточным, поскольку ввод сильно зависит от возможностей человека (например, подчерк).	Качество ввода информации высокого качества, поскольку ввод контролируется информационной системой.
5.	Создание записи в журнале.	Качество ввода информации можно считать недостаточным, поскольку ввод сильно зависит от возможностей человека (например, подчерк).	Качество ввода информации высокого качества, поскольку ввод контролируется информационной системой.

Из рассмотрения данной таблицы можно сделать вывод, что качество информации при использовании информационной системы существенно выше, чем при использовании традиционного способа. В целом по разделу можно отметить, что автоматизация задачи позволит повысить эффективность работы, вместе с тем уменьшив количество возникающих ошибок и увеличив качество хранимой информации, снизить простои, возникшие вследствие неисправностей техники.

1.3 Выбор и обоснование средств разработки

Технической заданием на разработку автоматизированной системы предполагается при разработке системы использовать язык программирования и систему для хранения данных [11].

В качестве языка программирования необходимо выбрать язык, позволяющий реализовать систему в кратчайшие сроки и с должной эффективностью. Следовательно, необходимо выбрать среду разработки, имеющую в комплектации средства для работы с исходными данными. Среди таких средств автором были рассмотрены Delphi 7, C++ PHP.

Поскольку, по условиям задачи, требуется обеспечить простую переносимость программного обеспечения, язык PHP не подходит, поскольку требует наличия вебсервера и интерпретатора.

Delphi и C++ имеют примерно равные функциональные возможности в рамках решения данной задачи, поэтому автором было отдано предпочтение среде разработки Delphi 7, как наиболее знакомой. Ниже будет дана её краткая характеристика [7].

Delphi 7.0 одна из самых мощных систем визуального объектно-ориентированного программирования, позволяющая решать множество задач. создавать законченные приложения для Windows самой различной направленности, от чисто вычислительных и логических, до графических и мультимедиа; быстро создавать (даже начинающим программистам) профессионально выглядящий оконный интерфейс для любых приложений, написанных на любом языке; интерфейс удовлетворяет всем требованиям Windows и автоматически настраивается на ту систему, которая установлена на компьютере пользователя, поскольку использует многие функции, процедуры, библиотеки Windows; создавать мощные системы работы с локальными и удаленными базами данных любых типов; при этом имеются средства автономной отладки приложений с последующим выходом в сеть; создавать многозвенные распределенные приложения, основанные на различных технологиях; создавать приложения, которые управляют другими приложениями, в частности, такими программами Microsoft Office, как Word, Excel и др.; создавать приложения различных классов

для работы в Internet; создавать профессиональные программы установки для приложений, учитывающие всю специфику и все требования Windows; и многое, многое другое, включая создание отчетов, справочных систем, библиотек DLL, компонентов ActiveX и т.п.

Delphi базируется на языке Object Pascal (начиная с Delphi 7 разработчики назвали его языком Delphi). Компиляторы с языков семейства Паскаль фирмы Borland (начиная с Turbo Pascal 1.0), были одними из самых быстрых компиляторов. В настоящее время Object Pascal – это объектно-ориентированный язык в виде хорошего компилятора.

Однако ориентация Delphi на Object Pascal несколько не сужает возможностей разработчика.

В целом, Delphi – великолепный инструмент, как для начинающих программистов, так и для профессионалов программирования

Кроме программного модуля, система должна обладать возможностью хранить преобразованные данные. Наиболее современный и эффективный способ хранения больших объемов структурированных данных – база данных. Для управления базой данных необходимо использование системы управления базами данных (СУБД) Разрабатываемая система должна обладать возможностью хранить преобразованные данные. Наиболее современный и эффективный способ хранения больших объемов структурированных данных – база данных. Для управления базой данных необходимо использование системы управления базами данных (СУБД)

Среди наиболее ярких представителей систем управления базами данных можно отметить: Microsoft Access, Borlandd Base, Borland Paradox, Microsoft Visual Fox Pro, Microsoft SQL Server, Oracle, MySQL, Postge SQL, используемые в приложениях, построенных по технологии «клиент-сервер» [7]. Сравним представленные СУБД по основным критериям (см. таблицу 1), предъявленным разработчиком (см. таблицу 1.2). В каждой графе будет проставляться «+», если система полностью удовлетворяет требованию, «+/-» если система частично удовлетворяет требуемому критерию, «-» - если требование не удовлетворяется.

Сравнение характеристик СУБД

	MicrosoftAccess	BorlanddBase	BorlandParadox	MicrosoftVisualFoxPr	MicrosoftSQLServer	Oracle	MySQL	PostgreSQL
Транспортабельность	+	+	+	-	-	-	-	-
Низкая требовательность к системным ресурсам	+	+	+	-	-	-	+	+
Наличие интегрированных визуальных средств проектирования и обслуживания	+	+/-	+/-	+	+	+	-	-
Наличие встроенных систем контроля целостности данных, триггеров, систем безопасности	+/-	+/-	+/-	+	+	+	-	+
Наличие интегрированного языка программирования и манипулирования данными	+	-	-	+	+	+	-	-

Продолжение таблицы 1.3

Низкая стоимость системы	+/-	+/-	+/-	-	-	-	+	+
Наличие документации	+	+	+	+	+	+	+	+
Поддержка и обновления	+	-	-	+	+	+	+	+
Итоговая оценка	7	4,5	4,5	5	5	5	4	5

Как видно из таблицы, наиболее удовлетворяет требуемым критериям СУБД MSACCES, поэтому она и будет использоваться при разработке информационной системы автоматизации.

Microsoft Office Access или просто Microsoft Access — реляционная СУБД корпорации Microsoft. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA, в самом Access можно писать приложения, работающие с базами данных [13].

Основные данные. Тип: РСУБД; Разработчик: Microsoft; Операционная система: Microsoft Windows; последняя версия: 2013 (15.0.4420.1017) (2 октября 2012); Лицензия: Проприетарное программное обеспечение EULA;

Вывод.

В данном разделе было приведено описание предметной области и её бизнес-процессов. Выполнена оценка эффективности выполнения бизнес-процессов, выявлены их недостатки, предложены решения по их усовершенствованию. В конце главы сделан выбор средств реализации программного обеспечения.

**ГЛАВА 2. РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ
«АВТОМАТИЗАЦИЯ БИЗНЕС-ПРОЦЕССОВ МУНИЦИПАЛЬНОГО
РЕСУРСНОГО ЦЕНТРА ПО РАБОТЕ С ОДАРЕННЫМИ ДЕТЬМИ
ЯВЛЯЕТСЯ СОВЕРШЕНСТВОВАНИЕ РАБОТЫ ПО ВЫЯВЛЕНИЮ,
ПОДДЕРЖКЕ И РАЗВИТИЮ ОДАРЕННЫХ И ТАЛАНТЛИВЫХ ДЕТЕЙ НА
ТЕРРИТОРИИ МО ГО «УСИНСК»».**

**2.1 Разработка базы данных ИС «Автоматизация бизнес-процессов
Муниципального ресурсного центра по работе с одаренными детьми является
совершенствование работы по выявлению, поддержке и развитию одаренных
и талантливых детей на территории МО ГО «Усинск»»**

Для проектирования данной программы была выбрана СУБД MS Access, которая уже имеется на предприятии и используется для хранения данных учётных системы предприятия. Структура базы – реляционная. Была спроектирована база данных, состоящая из таблиц, структура которых приведена в таблицах 2.1 – 2.7.

Таблица 2.1

Структура таблицы «Заявки»

Наименование поля	Формат	Пояснение
ID	Счётчик	Ключ
Дата_подачи	Дата	Дата подачи заявления
Подразделение	Числовой	Ссылка на подразделение
Тип_неисправности	Числовой	Ссылка на неисправность
Оборудование	Числовой	Ссылка на оборудование
Заявитель	Числовой	Ссылка на заявителя

Продолжение таблицы 2.1

Статус	Числовой	Ссылка на статус
Исполнитель	Числовой	Ссылка на исполнителя
Срочность	Текст	Срочность
Комментарий	Мемо	Комментарий
Работы	Числовой	Ссылка на работы

Вывод. Таблица состоит из нескольких полей различного типа.

Таблица 2.2

Структура таблицы «Сотрудники»

Наименование поля	Формат	Пояснение
ID	Счётчик	Ключ
Фамилия	Текст	Фамилия
Имя	Текст	Имя
Отчество	Текст	Отчество
Признак	Числовой	Признак учётной записи
Логин	Текст	Логин
Пароль	Текст	Пароль

Вывод. Таблица состоит из нескольких полей различного типа.

Таблица 2.3

Структура таблицы «Справочник неисправностей»

Наименование поля	Формат	Пояснение
ID	Счётчик	Ключ
Наименование	Текст	Наименование
Оборудование	Числовой	Ссылка на оборудование

Вывод. Таблица состоит из нескольких полей различного типа.

Таблица 2.4

Структура таблицы «Справочник оборудования»

Наименование поля	Формат	Пояснение
ID	Счётчик	Ключ
Наименование	Текст	Наименование

Вывод. Таблица состоит из нескольких полей различного типа.

Таблица 2.5

Структура таблицы «Справочник подразделений»

Наименование поля	Формат	Пояснение
ID	Счётчик	Ключ
Наименование	Текст	Наименование

Вывод. Таблица состоит из нескольких полей различного типа.

Таблица 2.6

Структура таблицы «Справочник работ»

Наименование поля	Формат	Пояснение
ID	Счётчик	Ключ
Наименование	Текст	Наименование

Вывод. Таблица состоит из нескольких полей различного типа.

Таблица 2.7

Структура таблицы «Справочник статусов»

Наименование поля	Формат	Пояснение
ID	Счётчик	Ключ
Наименование	Текст	Наименование

Вывод. Таблица состоит из нескольких полей различного типа.

Для связи таблиц и реализации была разработана схема данных, представленная на рисунке 2.1.

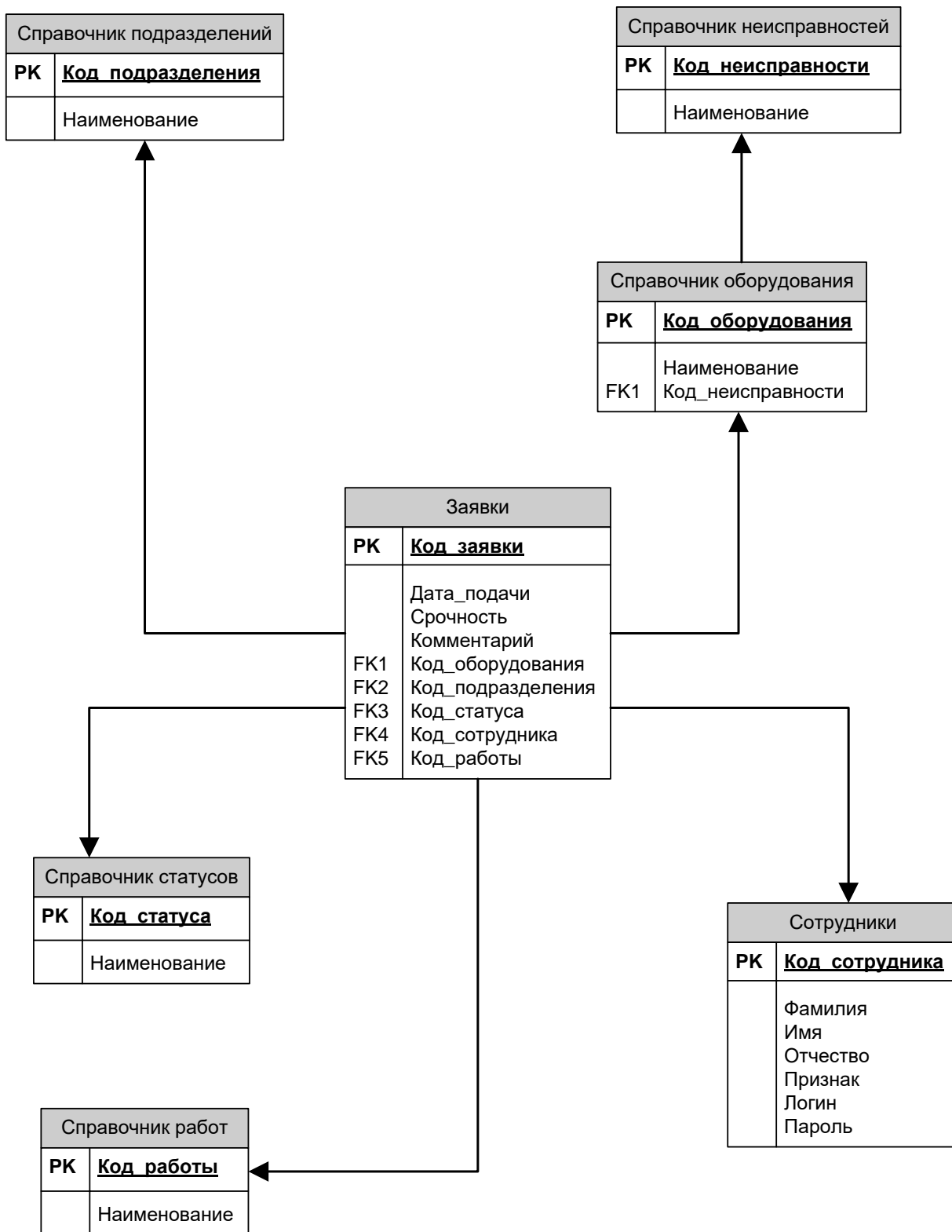


Рисунок 2.1 - Структура БД.

Вывод. Схема данных состоит из нескольких таблиц.

Физическая реализация изображена на рис. 2.2.

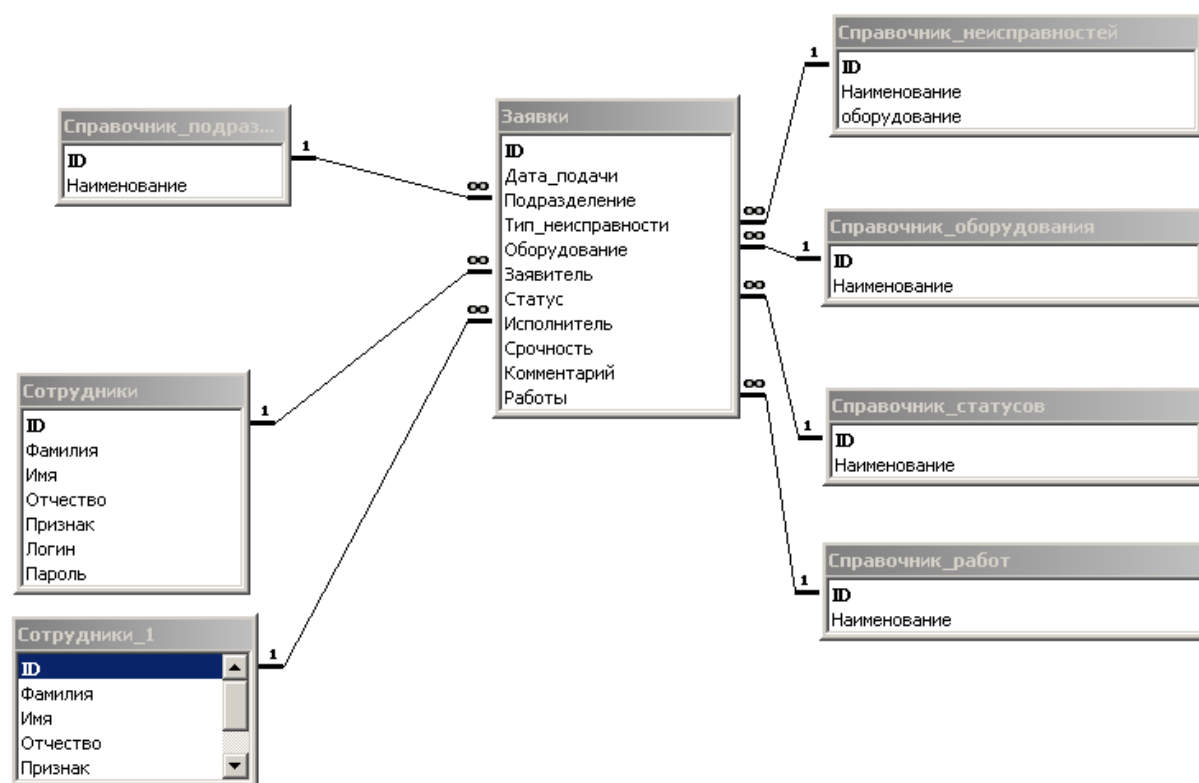


Рисунок 1.2 – Физическая реализация БД

Вывод. Физическая реализация соответствует схеме.

2.2 Разработка программного модуля ИС «Автоматизация бизнес-процессов Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск»»

Программный продукт состоит из программного модуля и базы данных. Доступ к данным осуществляется через технологию ADO. Программа построена по трёхзвенной архитектуре с тонким клиентом [20]. Структура программного продукта изображена на рис. 2.3.



Рисунок 2.3 - Структура программного продукта.

Вывод. Структура состоит из трёх частей.

Схема пакета программных модулей построена по иерархическому принципу. Структура иерархии приведена на рисунке 2.4.

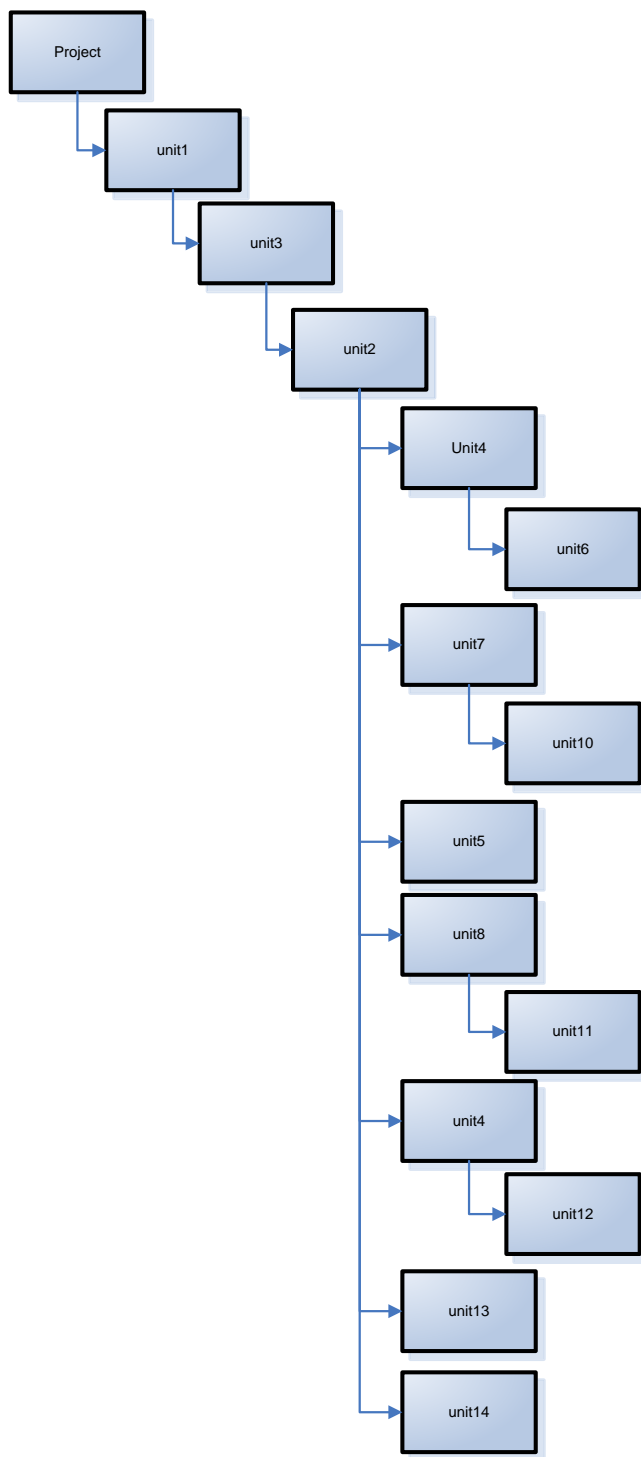


Рисунок 2.4 - Структурная схема пакета.

Вывод. Структура состоит из нескольких частей.

Схема работы программного продукта приведена на рисунке 2.5.

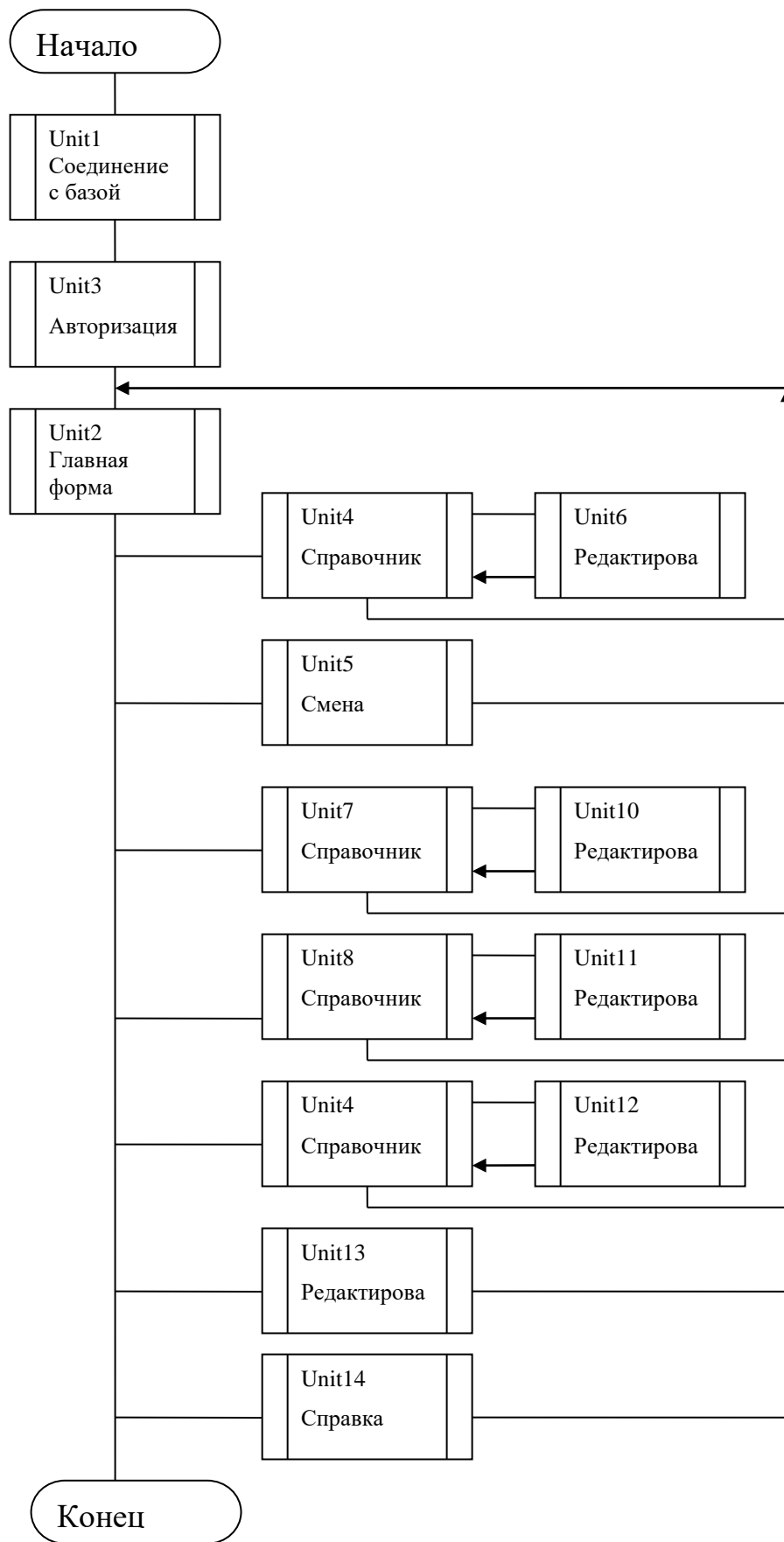


Рисунок 2.5 - Схема работы программного продукта.

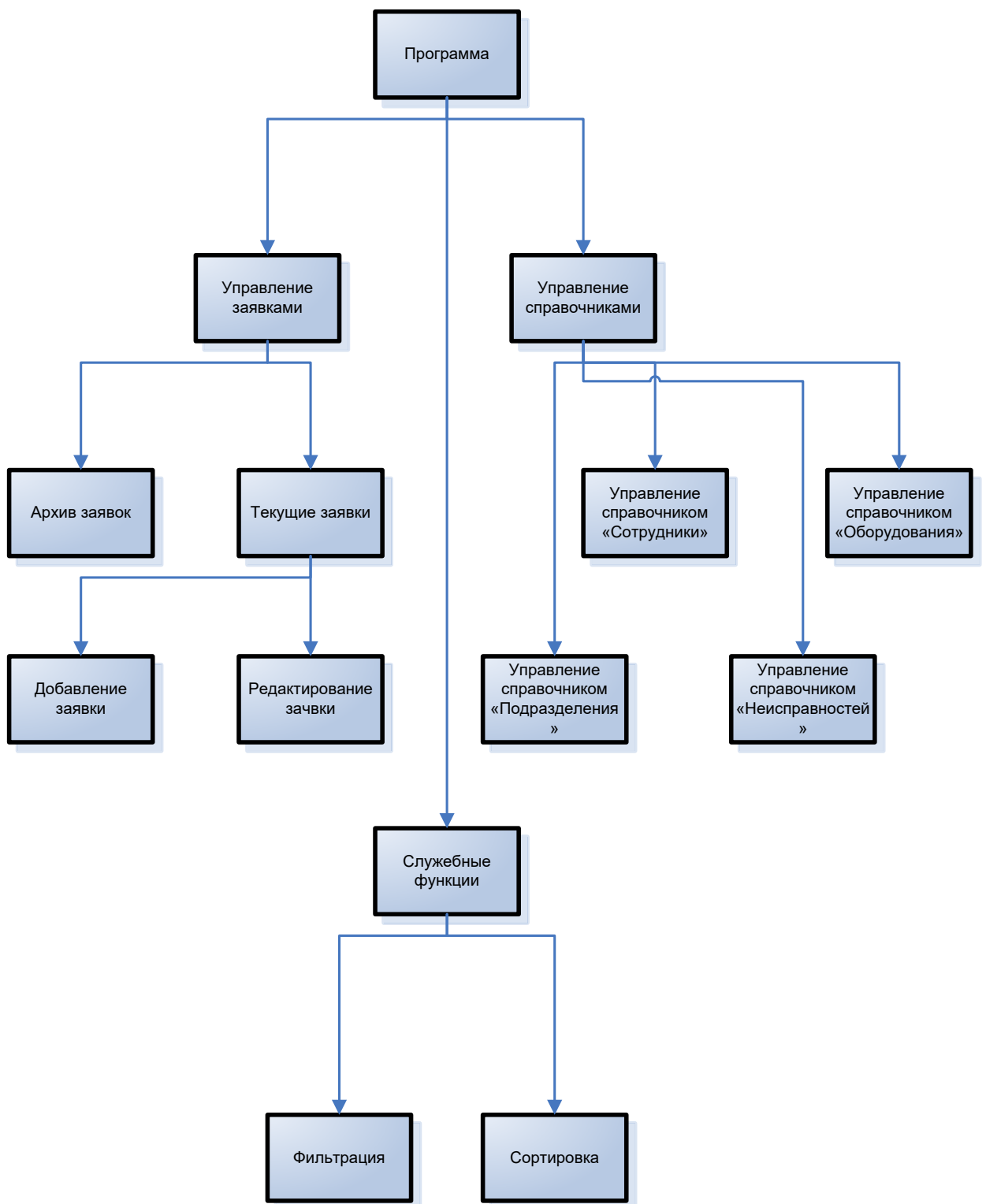
Список и назначение модулей программы.

Unit1 – Обеспечивает соединение с БД. Unit2 – Главная форма программы. Unit3 – Модуль авторизации. Unit4 – Модуль справочника пользователей. Unit5 – Смена пароля. Unit6 – Модуль редактирования справочника пользователей. Unit7 – Модуль справочника подразделений. Unit8 – Модуль справочника оборудования. Unit8 – Модуль справочника неисправностей. Unit10 – Модуль редактирования справочника подразделений. Unit11 – Модуль редактирования справочника оборудования. Unit12 – Модуль редактирования справочника неисправностей. Unit13– Модуль редактирования заявок. Unit14– Справка.

2.3 Разработка интерфейса ИС «Автоматизация бизнес-процессов Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск»»

Интерфейс приложения представляет собой стандартный многооконный интерфейс Windows-приложения. При проектировании интерфейса преимущество отдавалось обеспечению удобства и интуитивной понятности при работе с приложением. Все функции приложения, собранные в иерархическое дерево.

Дерево функций автоматизированной системы изображено на рис.2.6.



Вывод: Дерево функций имеет разветвлённую структуру.

Рисунок 2.6. Дерево функций автоматизированной системы.

2.4 Инструкция пользователя ИС «Автоматизация бизнес-процессов Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск»»

После запуска приложения и показа заставки пользователю программного продукта будет предложено пройти авторизацию. При этом он должен указать тип учётной записи (пользователь, техник, или диспетчер). В случае, если пользователь не верно указал логин, пароль или тип доступа, то ему будет отказано в авторизации (см. рис. 2.7)

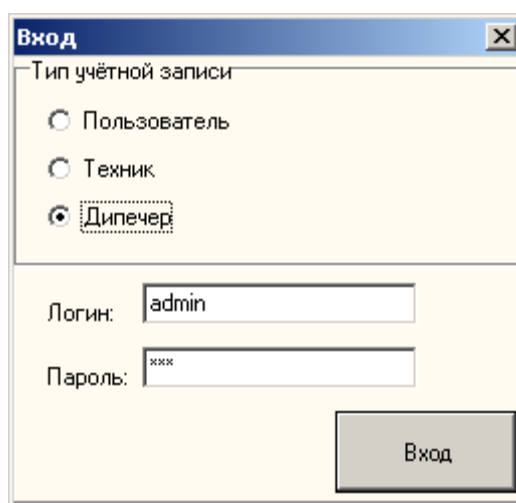


Рисунок 2.7. Авторизация.

Вывод: В случае, если пользователь не верно указал логин, пароль или тип доступа, то ему будет отказано в авторизации.

В случае удачного входа будет открыто главное окно программного продукта (см. рис. 2.8).

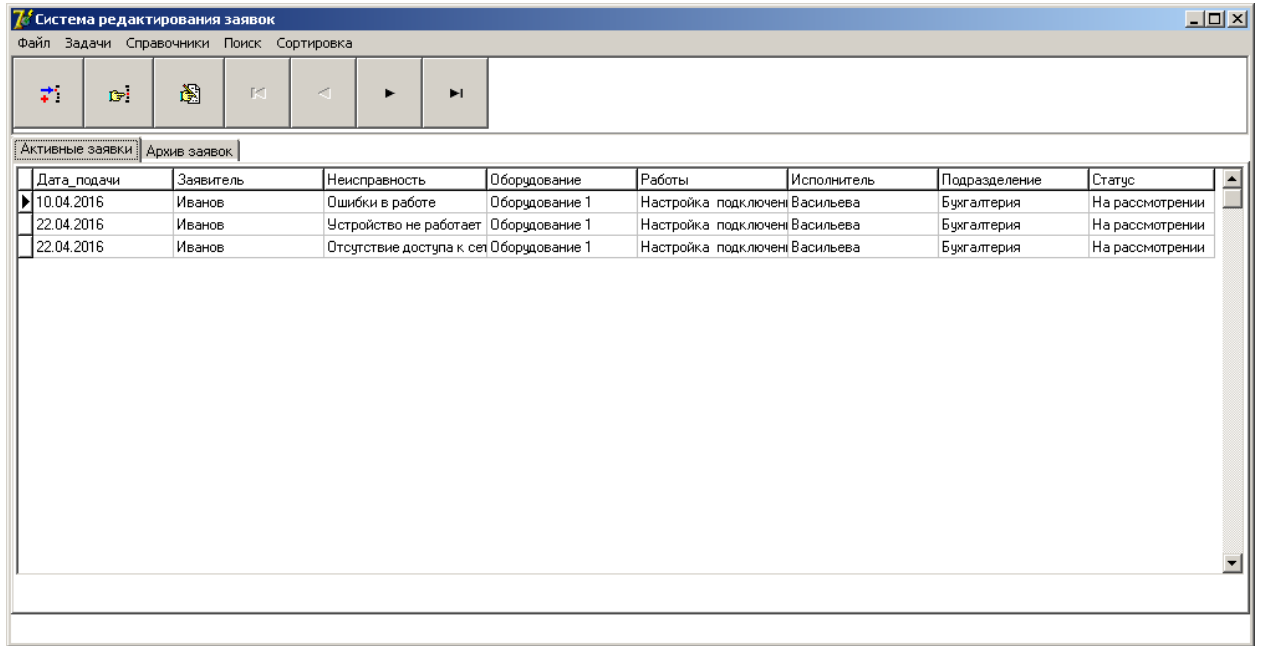


Рисунок 2.8 - Главное окно программного продукта.

Вывод: В случае удачного входа будет открыто главное окно программного продукта.

В данном окне отображаться активные заявки пользователей (не имеющие статус «закрыто»). После установки статуса «закрыто» заявки перемещаться в архив (см. рис. 2.9).

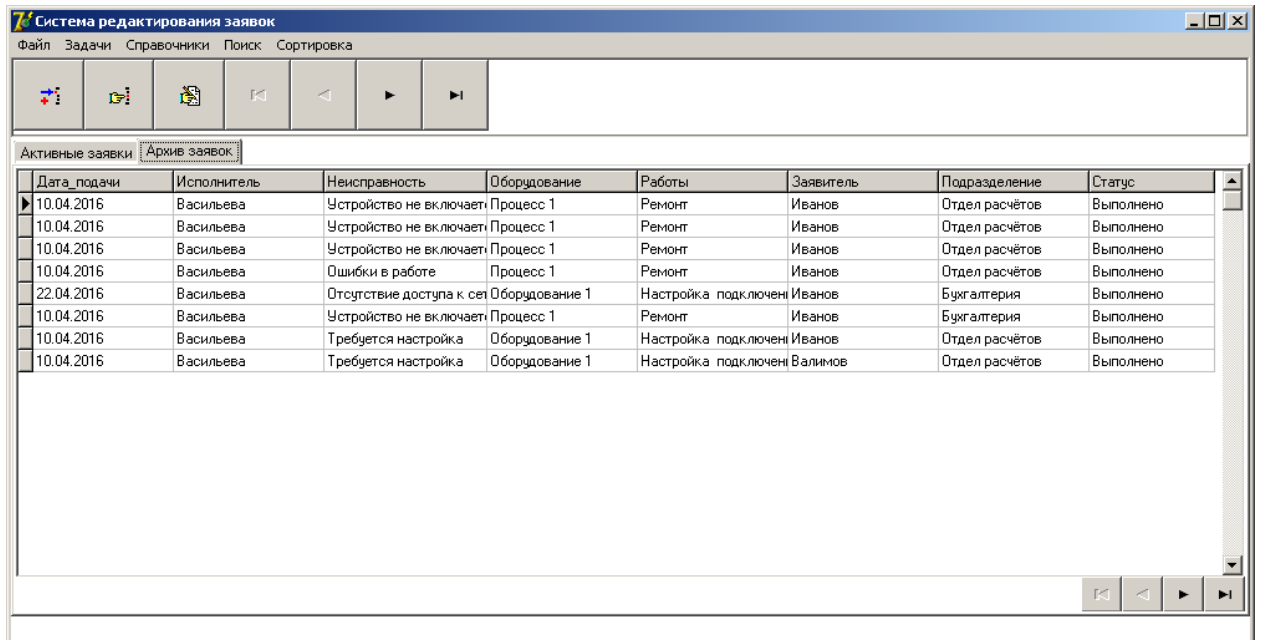


Рисунок 2.9 - Архив заявок.

Вывод: после установки статуса «закрыто» заявки перемещаться в архив.

Для ввода и редактирования нормативно справочной информации в программе предусмотрены справочники сотрудников, подразделений, оборудования, неисправностей (см. рис. 2.10-2.13). Журнал и каждый справочник позволяет. Перемещаться по записям в БД. Добавлять новые данные. Редактировать данные. Удалять данные. Осуществлять поиск по любому столбцу. Устанавливать сортировку.

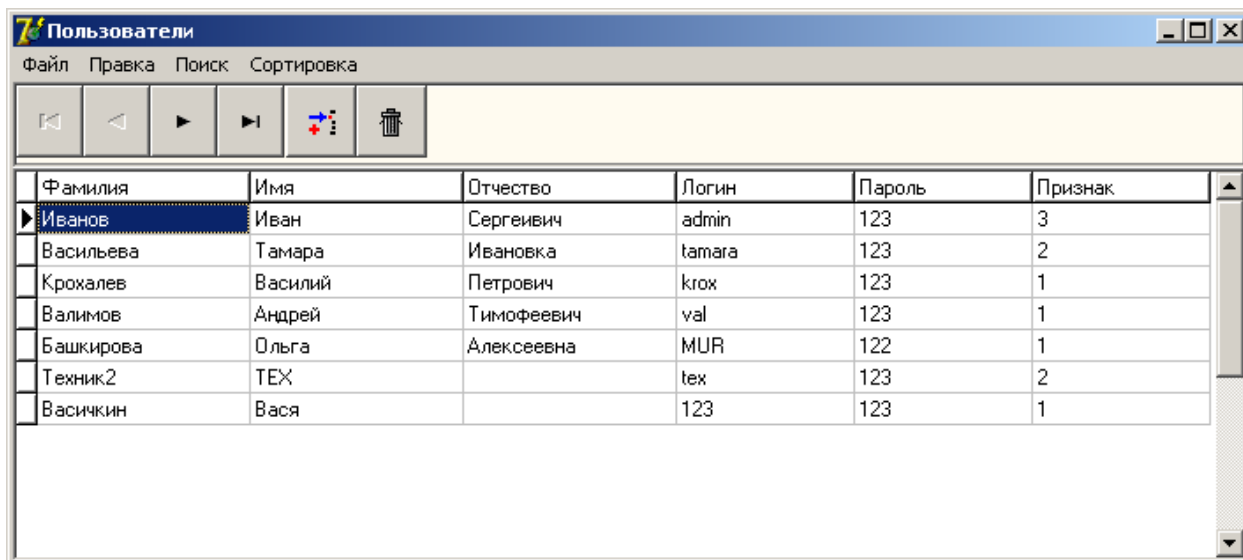


Рисунок 2.10 - Справочник «Пользователи».

Вывод: Форма позволяет работать со справочником.

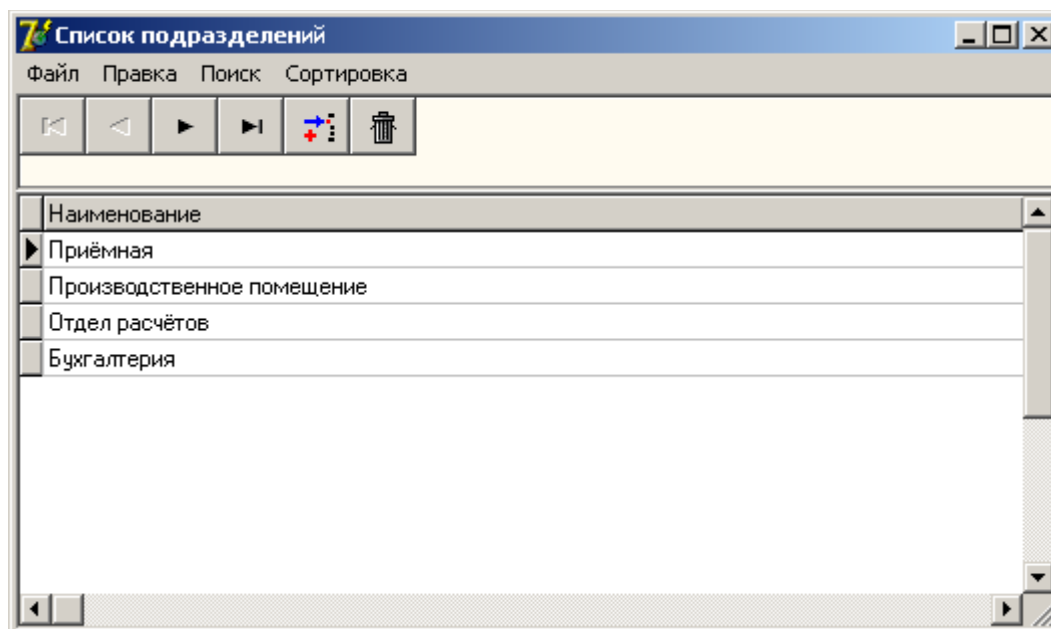


Рисунок 2.11 - Справочник «Подразделений».

Вывод: Форма позволяет работать со справочником.

Вывод: Форма позволяет работать со справочником.

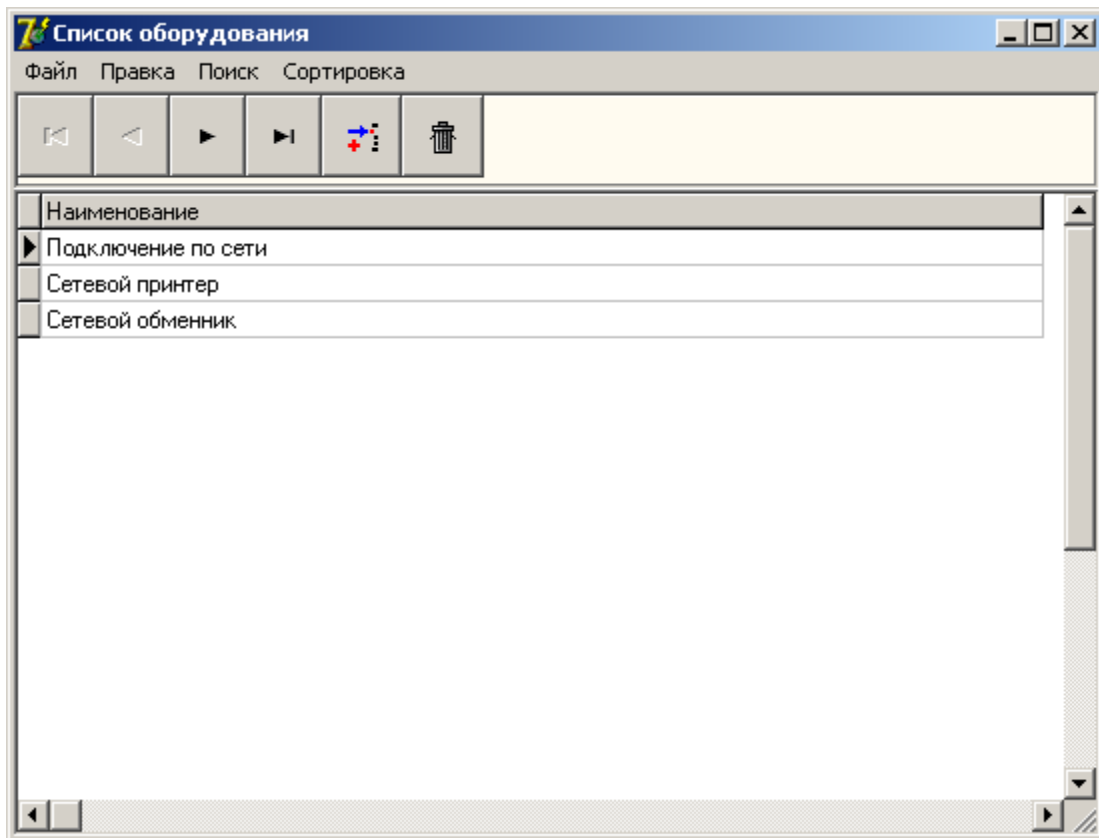


Рисунок 2.12 - Справочник «Список оборудования».

Вывод: Форма позволяет работать со справочником.

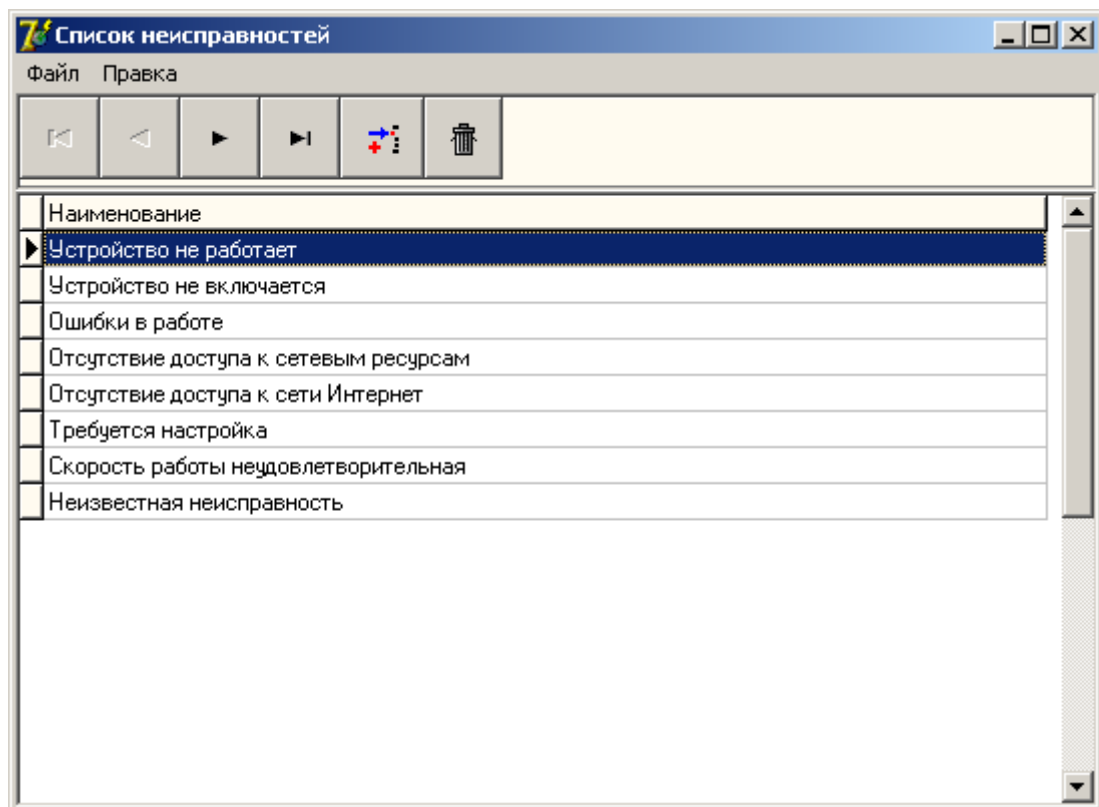


Рисунок 2.13 - Справочник «Список неисправностей».

Перемещение по справочнику.

Для перемещения по записям базы данных используются кнопки. Вперед на одну запись. В конец. Назад на одну запись. В начало. Добавление новой записи.

Для добавления новой записи необходимо кликнуть на кнопку «Добавить новую запись». При этом будет открыто окно для создания записи. Каждый справочник имеет собственное окно (см. рис. 2.14-2.17).

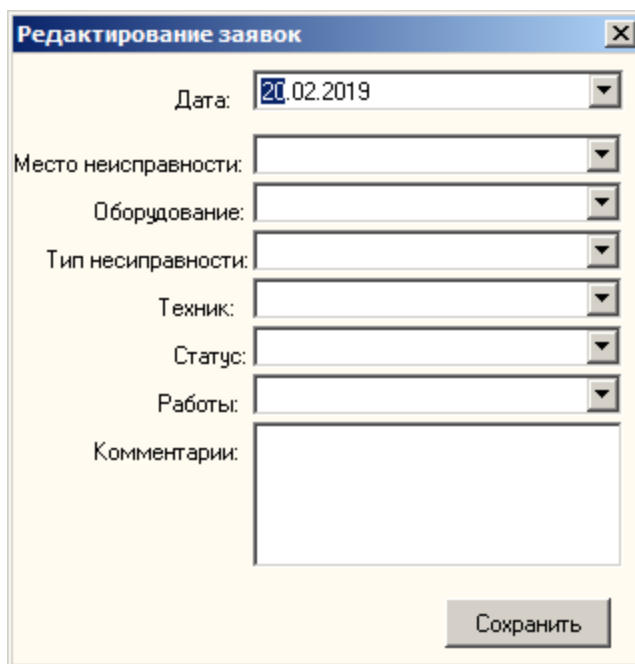


Рисунок 2.14 - Добавление заявки.

Вывод: Форма позволяет работать со справочником.

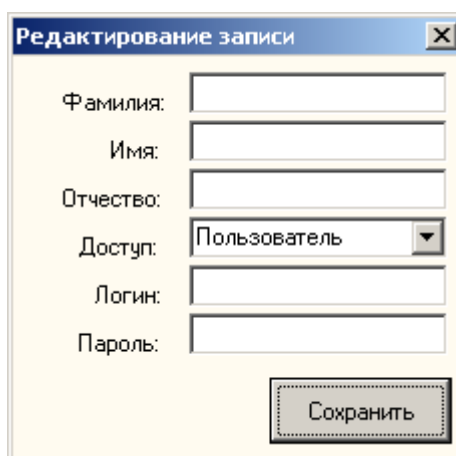


Рисунок 2.15 - Добавление записи в справочник «Сотрудники».

Вывод: Форма позволяет работать со справочником.

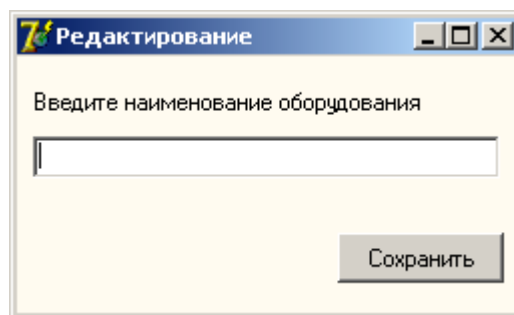


Рисунок 2.16 - Добавление записи в справочник «Оборудование».

Вывод: Форма позволяет работать со справочником.

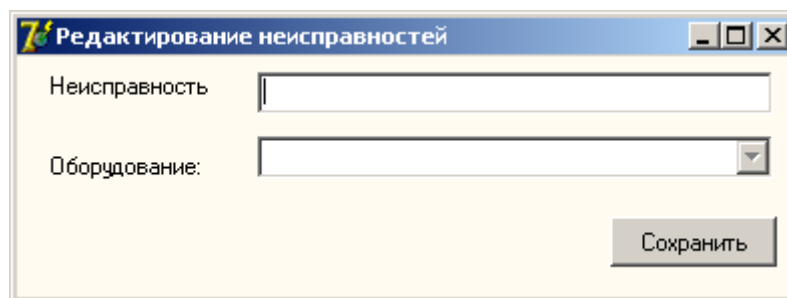


Рисунок 2.17 - Добавление записи в справочник «Неисправности».

Вывод: Форма позволяет работать со справочником.

Редактирование записи.

Для редактирования новой записи необходимо установить курсор кликнуть на кнопку «Редактировать новую запись». При этом будет открыто окно аналогичное окну создания новой записи с заполненными данными для редактирования.

Удаление записи.

Для удавления записи необходимо установить курсор на запись и нажать кнопку удалить. На запрос программы нужно подтвердить удаление.

Поиск записи.

Для поиска записи необходимо установить курсор на столбец, по которому будет осуществляться поиск, и выбрать в главном меню «Поиск - Найти запись», в открытом окне нужно указать параметры поиска и нажать «Ок» (см. рис. 2.18). Для сброса результатов поиска выполните команду «Поиск – отменить поиск».

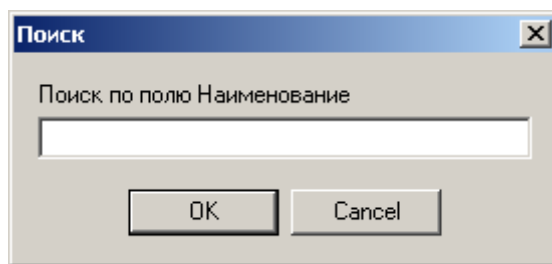


Рисунок 2.18 - Окно поиска.

Вывод: Форма позволяет работать со справочником.

Сортировка.

Для сортировки по столбцу установите курсор в требуемый столбец и выберите «Сортировка – по возрастанию/убыванию».

Сообщения пользователю представлены в таблице

Таблица 2.10

Сообщения пользователю

№ п/п	Сообщение	Описание
1	Доступ закрыт	Уровень доступа пользователя не позволяет воспользоваться данным интерфейсом
2.	Поиск по полю «Наименование поля»	Пользователю необходимо указать критерия поиска по полю.
3.	Неверно введён пароль	Пользователь неверно указал логики и/или пароль
4.	Вы действительно хотите удалить запись?	Запрос на подтверждение удаления записи.
5.	Старый пароль соответствует новому паролю.	Сообщение, возникающее при смене пароль в том случае, если старый пароль соответствует новому паролю.
6.	Укажите строку соединения	Программа не смогла обнаружить базу данных по предполагаемому адресу и предлагает пользователю самому казать месторасположение базы данных.

2.5 Описание технологии работы с информационной системой

Под технологическим обеспечением понимается совокупность методов обработки и обмена информацией в информационной системе.

Порядок обработки информации в программном продукте приведён на рис. 2.19, 2.20, 2.21 [16][14].

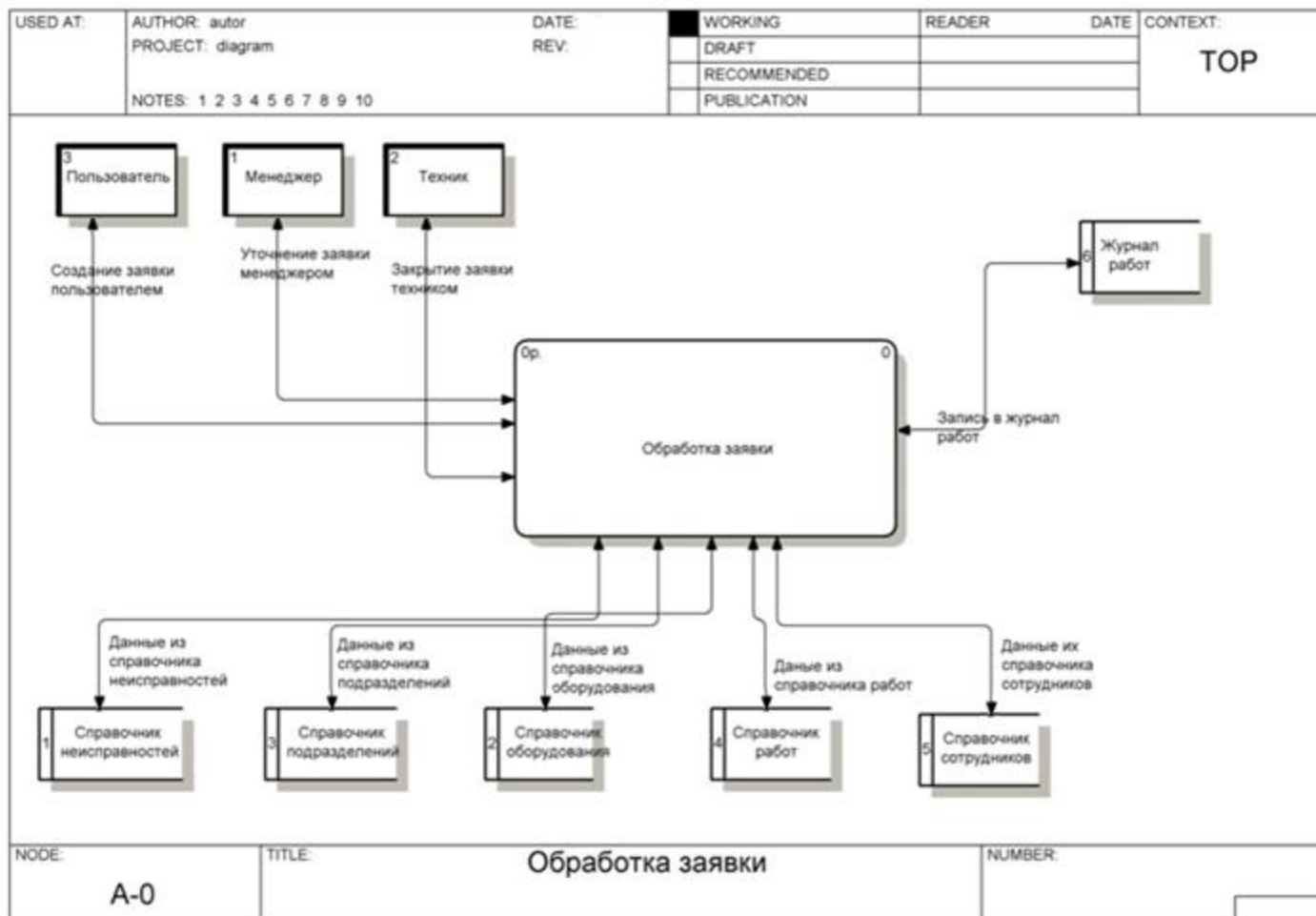


Рисунок 2.19. Схема работы в формате DFD

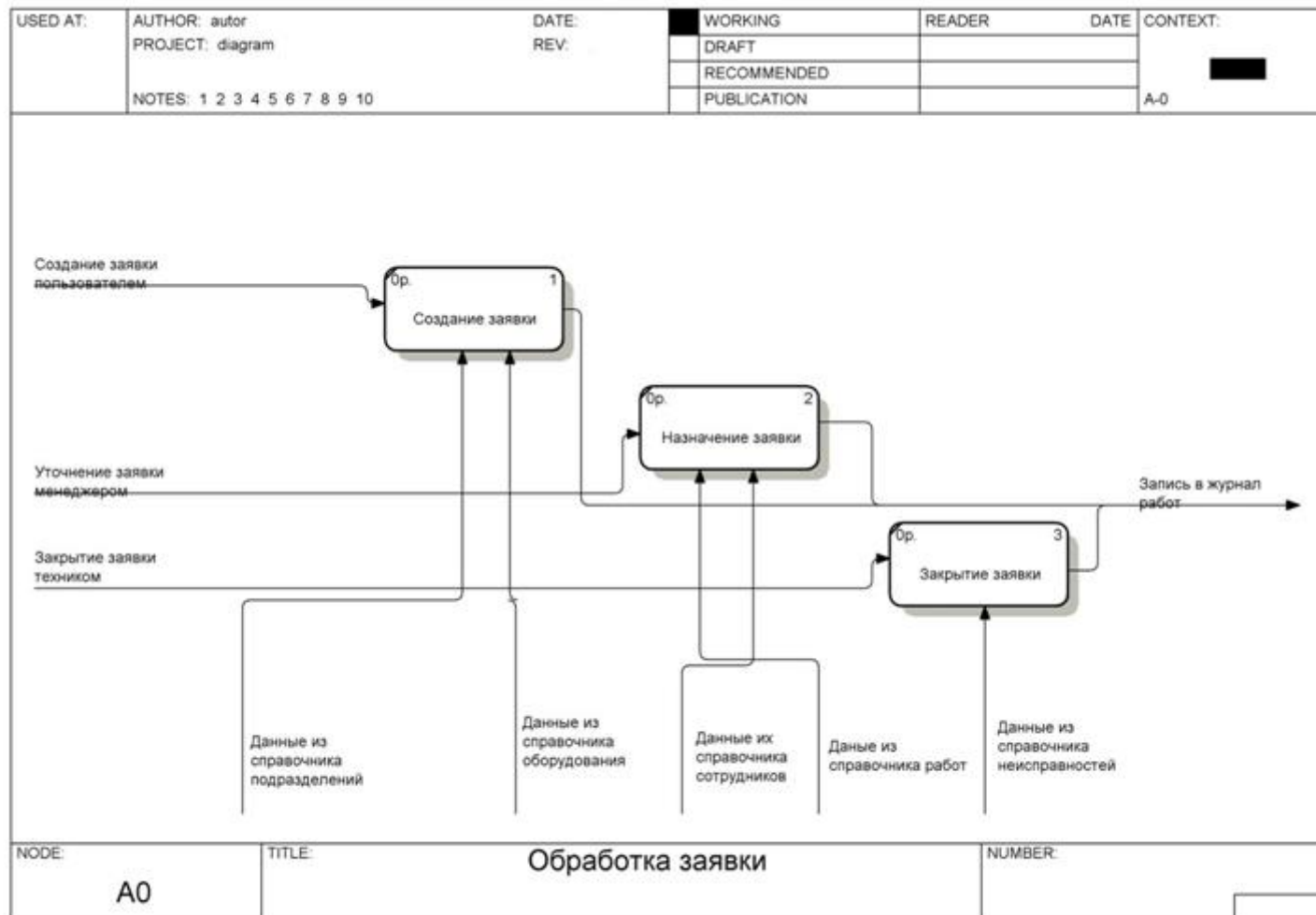


Рисунок 2.20. Схема работы в формате DFD

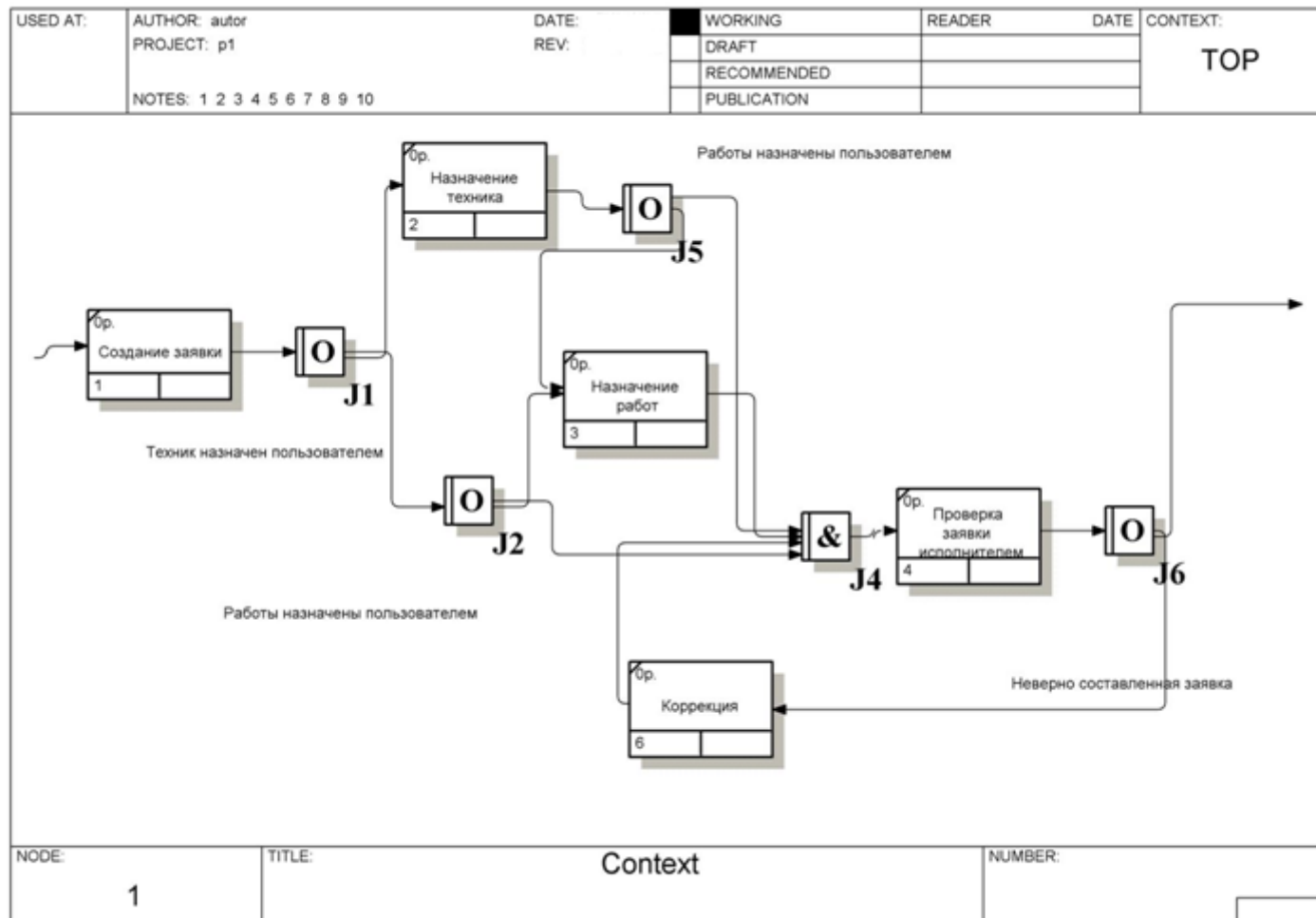


Рисунок 2.21. Схема работы с системой в рамках бизнес-процесса.

Вывод: Технология выполнения бизнес-процесса изменилась.

Вывод. Данная глава имеет практическое направление по посвящена
детальному описанию разработки информационной системы. Она включает в себя
техническое задание, разработку архитектуры, базы данных, программного модуля,
интерфейса и технологии работы с информационной системой.

**ГЛАВА 3 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РЕНТАБЕЛЬНОСТИ
РАЗРАБОТКИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ
«АВТОМАТИЗАЦИЯ БИЗНЕС-ПРОЦЕССОВ МУНИЦИПАЛЬНОГО
РЕСУРСНОГО ЦЕНТРА ПО РАБОТЕ С ОДАРЕННЫМИ ДЕТЬМИ
ЯВЛЯЕТСЯ СОВЕРШЕНСТВОМ РАБОТЫ ПО ВЫЯВЛЕНИЮ,
ПОДДЕРЖКЕ И РАЗВИТИЮ ОДАРЕННЫХ И ТАЛАНТЛИВЫХ ДЕТЕЙ НА
ТЕРРИТОРИИ МО ГО «УСИНСК»».**

3.1 Оценка затрат

Смета затрат на проведение работы включает следующие статьи:

1. материалы;
2. командировочные расходы;
3. контрагентские расходы;
4. покупные изделия;
5. расходы, связанные с оплатой труда;
6. накладные расходы.

3.1.1 Материалы

Расходы на материалы представлены в таблице 3.1.

Таблица 3.1

Материалы

Ном ер	Наименование	Ед. измери я	Цена, руб.	Количес тво	Сумма, руб.
1	Бумага	пачка	100,00	1	100,00
2	Тонер	упаковка	150,00	0,5	75,00
3	Дискета	шт.	15,00	5	75,00
4	CD-RW	шт.	32,00	3	96,00
5	Авторучка	шт.	7,00	4	28,00

Продолжение таблицы 3.1

6	Карандаш	шт.	2,00	4	8,00
7	Степлер	шт.	30,00	1	30,00
8	Скрепки для степлера	упаковка	15,00	2	30,00
	ИТОГО				442,00

Вывод: в таблице представлена смета расходов на материалы.

3.1.2 Командировочные расходы

Командировочные расходы представлены в таблице 3.2.

Таблица 3.2

Командировочные расходы

Номер	Наименование	Цена, руб.	Количество поездок	Сумма, руб.
1	Билет гор. транспорта	15,00	150	2 550,00
2	Билет маршрутного такси	18,00	30	540,00
	ИТОГО			2790,00

Вывод: в таблице представлена смета командировочных расходов.

3.1.3 Контрагентские расходы

К контрагентским расходам относятся все расходы по оплате услуг сторонним организациям, в том числе машинное время (см. таблицу 3.3).

Таблица 3.3

Контрагентские расходы

Номер	Наименование	Ед. измерения	Цена, руб.	Количество	Сумма, руб.
1	Аренда машинного времени	маш. час	20,00	420	8 475,00
	ИТОГО				8475,00

Вывод: в таблице представлена смета командировочных расходов.

3.1.4 Покупные изделия

Затраты на покупные изделия отображены в таблице 3.4.

Таблица 3.4

Покупные изделия и специальное оборудование

Номер	Наименование	Ед. измерения	Цена, руб.	Количество	Сумма, руб.
1	Литература	Шт.	490,00	1	490,00
2	Упаковка дисков		80,00	1	80,00
3	Бумага для печатающих устройств		250,00	1	250,00
	ИТОГО				820,00

Вывод: в таблице представлена смета покупных изделий.

3.1.5 Расходы, связанные с оплатой труда

Рассчитаем заработную плату сотрудников в зависимости от занимаемой должности и количества отработанных дней, для чего найдем стоимость одного дня, равную отношению оклада работника к количеству дней в месяце (примем равным 22 дням).

Таблица 3.5

Заработная плата

Номер	Должность	Оклад, руб.	Количество отработанных дней	Стоимость одного дня, руб.	Сумма, руб.
1	Руководитель	30000	15	1363,63	20454,55
2	Инженер	20000	123	909,09	111818,18
	ИТОГО				132272,73

Вывод: в таблице представлена смета расходов на оплату труда.

Рассчитаем социальные начисления на заработную плату. Они составляют 28% от фонда оплаты труда, т.е. в нашем случае $132272,73 \cdot 0,28 = 37036,36$ руб.

3.1.6 Накладные расходы

Накладные расходы включаю в себя все расходы на обслуживание, и составляют 10% от суммы, полученной в пунктах с 1 по 5 (18122,17 руб.). Таким образом, после нахождения накладных расходов легко можем найти итоговую сумму – себестоимость проведенной работы (см. табл. 3.8).

Таблица 3.8

Смета затрат

Номер	Наименование статьи расходов	Сумма, руб.
1	Материалы	442,00
2	Командировочные расходы	2790,00
3	Контрагентские расходы	8475,00
4	Покупные изделия	820,00
5	Расходы, связанные с оплатой труда	169309,09
6	Накладные расходы	18183,17
	ИТОГО	200019,26

Вывод: в таблице представлена смета затрат.

3.2 Описание организационных эффектов

В результате внедрения бизнес-процесса обработки заявок пользователей получилось добиться следующих организационных эффектов:

- Увеличение скорости обработки заявки.
- Повешение качества работы информационно-технического персонала.
- Централизованный контроль выполнения заявок.
- Выполнение заявок в соответствии с приоритетами.
- Накопление знаний.

Полученные организационные эффекты существенно повысили эффективность работы отдела СПВЦ.

3.3 Оценка экономических эффектов

Внедрение информационной системы позволило освободить ставку одного инженера и перевести его на другие работы. Расходы по обслуживанию места инженера составляли:

1. ФОТ и отчисления – 31 680 т.р.
2. Эксплуатационные расходы на рабочее место (в.т.ч. амортизационные, накладные, транспортные и т.д.) 15 000.

Таким образом, экономия в год ставит:

$$(31680+15000) *12 = 560 160 \text{ руб. в год.}$$

Кроме экономии от упрощения технологии выполнения бизнес-процесса, так же будет непрямая экономия, связанная с общим улучшением работы электронно-вычислительной техники в организации. Показатели, характеризующие данную прибыль, могут быть рассчитаны только эмпирически. Предположительно данная экономия составит год не менее 300 000руб. Для расчёта будет использовать данное минимальное значение.

$$\text{Общая экономия составит } 560 160+300 000=860 160\text{руб.}$$

3.4 Оценка рисков

Предположительно возможно возникновение следующих рисков:

Неготовность персонала. Возникновение данного риска характерно для внедрения большинства прогрессивных технологий, с которыми должны работать пользователи. С точки зрения внедрения данной системы автоматизации этот риск невелик, поскольку большая часть системы будет работать в автоматическом режиме. Роль персонала компании, в работе данной системы заключается в чисто административных функциях.

Проблемы безопасности при передаче данных. Стоимость решения проблем безопасности данных должна оцениваться как отношение стоимости похищения информации со стоимостью системы защиты. В данном случае, финансовые потери

от разглашения данной информации невелики, поэтому защита информации при передаче данных от сервера к приложению клиента будет решена стандартными способами обеспечения целостности сеанса.

Наличие угрозы разглашения коммерческих данных, которые не могут быть общедоступными. Угроза разглашения коммерческих данных, используемых в виде выходных системой автоматизации невелика, поскольку ценность таких данных невысока.

Угроза потери или порчи данных. В отличие от потерь при разглашении данных, потери от порчи или разглашении данных довольно велики, и решение данной проблемы требует пристального внимания. Избежать данной угрозы можно путём проведения правильной административной политики безопасности на уровне операционной системы, а также на уровне аккаунта доступа к СУБД.

Неготовность пользователей. Неготовность конечных пользователей к использованию системы автоматизации так же является большой угрозой при её внедрении, поскольку оказывает негативное влияние на достижение положительного результата в вопросах экономии времени и человеческих ресурсов в решении данной задачи. Следует так же отметить, что статистическая возможность возникновения данной ситуации уменьшается с каждым годом. Это связано с возрастающей компьютерной грамотностью и тотальной информатизацией общества. Для минимизации влияния этого риска возможно использовать различные наглядные обучающие фото и видеоматериалы, демонстрирующие основные принципы работы с системой.

Вывод

Проведя общую оценку внедрения бизнес-процесса управления пользователями на основе современных информационных технологий, можно сделать вывод, что данное внедрение целесообразно, что доказывается расчётом экономической эффективности.

ЗАКЛЮЧЕНИЕ

В процессе работы над данным дипломным проектом была достигнута начальная цель: разработано автоматизированное рабочее место руководителя или диспетчера отдела, позволяющее существенно повесить скорость, оперативность и эффективность работы отдела по обслуживанию информационной вычислительной техники. В процессе разработки программы были решены следующие задачи. Проанализированы бизнес-процессы распределения заявок, выявленные его слабые стороны, разработаны методы по его усовершенствованию. Выбраны средства разработки, которые будут использоваться при разработки автоматизированной системы. Разработано техническое задание. Разработан интерфейс пользователей для автоматизации создания заявок - требований по устранению неисправностей аппаратно – станочного комплекса или его настройки. Разработан инструмент диспетчера для распределения заявок по исполнителям из инженерно-технического обслуживающего персонала предприятия. Разработан интерфейс мастера (инженера) для получения заявок и в работу. Разработан интерфейс для обслуживания разрабатываемой системы её настройки и ввода всех необходимых для правильного функционирования данных. Разработана структура БД системы автоматизации. Разработана техническая документация по работе информационной системой.

Все поставленные цели были выполнены. Программное обеспечение было отлажено и запущено в тестовую эксплуатацию.

Эффективность внедрения данного программного продукта была доказана в экономической части работы.

ЛИТЕРАТУРА

1. Intuit. Лекция: Моделирование бизнес-процессов средствами BРwin [Электронный ресурс] – Режим доступа: <http://www.intuit.ru/department/se/devis/7/>
2. Автоматизированное проектирование систем управления / под ред. М. Джамшиди. – М.: Машиностроение, 2015. – 212с.
3. Бахтизин В.В., Глухова Л.А. Стандартизация и сертификация программного обеспечения: Учеб. пособие - Мн.: БГУИР, 2016. - 200с.
4. Джонс Дж. К. Методы проектирования / Дж. К. Джонс. – М.: Мир, 2011. – 320с.
5. Елиферов В.Г., Репин В.В. Бизнес-процессы: регламентация и управление - М.: ИНФРА-М, 2014. – 250с.
6. Инновационные системы совершенствования бизнес-процессов. [Электронный ресурс] – Режим доступа: <http://bedredinov.ru/>
7. Калянов Г.Н. Теория и практика реорганизации бизнес-процессов. - М.: Инфра-М, 2017. – 235с.
8. Маклаков С.В. Моделирование бизнес-процессов с AllFusion Process Modeler – М.: ДИАЛОГ-МИФИ, 2014.- 235с.
9. МАУДО "ЦДОД" г.Усинска - http://cdod-usinsk.ucoz.ru/index/shkola_olimpiadnogo_rezerva_shor/0-247
10. Муниципальный ресурсный центр по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск». Приказ о создании [Электронный ресурс] - Режим доступа: http://усинск-обр.рф/sites/default/files/page/files/pr_uo_no_1128_o_sozdanii_mrc_po_rabote_s_odarennymi_detmi.pdf
11. Норенков И. П. Введение в автоматизированное проектирование технических устройств и систем: учеб. пособие для втузов / И. П. Норенков. – М.: Высш. шк., 2013.

12. О системе выявления и поддержки одаренных детей и талантливой молодежи в Республике Коми [Электронный ресурс] - Режим доступа: <https://docplayer.ru/59229505-O-sisteme-vyyavleniya-i-podderzhki-odarenyh-detey-i-talantlivoy-molodezhi-v-respublike-komi.html>
13. Определение потребностей в CASE-средствах [Электронный ресурс] - Режим доступа: http://studopedia.ru/2_102408_opredelenie-potrebnostey-v-CASE-sredstvah.html
14. Официальный дистрибьютор CA technologies в России [Электронный ресурс] – Режим доступа: <http://www.interprocom.ru>
15. Официальный сайт CA ERwin® Process Modeler [Электронный ресурс] – Режим доступа: <http://www.ca.com>
16. Примеры комплексов CASE-средств [Электронный ресурс] - Режим доступа: <http://www.ref.by/refs/34/7739/1.html>
17. Рождественский Д. А. Автоматизация проектирования систем и средств управления: учеб. пособие Ч 1 / Д. А. Рождественский. – Томск: Том. межвуз. центр дистанционного образования, 2014. – 67с.
18. Рождественский Д. А. Автоматизация проектирования систем и средств управления: учеб. пособие Ч. 2 / Д. А. Рождественский. – Томск: Том. межвуз. центр дистанционного образования, 2014. – 131с.
19. Советов Б.Я., Яковлев С.А. Моделирование систем. Учебник для вузов. М.: Высшая школа, 2013. – 124с.
20. Современные методы и средства проектирования информационных систем [Электронный ресурс] - Режим доступа: <http://vernikov.ru/component/k2/item/340.html#32>

ПРИЛОЖЕНИЕ 1. СКРИПТ ДЛЯ СОЗДАНИЯ БАЗЫ ДАННЫХ

```
Dim ERwinWorkspace As Workspace
Dim ERwinDatabase As Database
Dim ERwinTableDef As TableDef
Dim ERwinQueryDef As QueryDef
Dim ERwinIndex As Index
Dim ERwinField As Field
Dim ERwinRelation As Relation
Set ERwinWorkspace = DBEngine.WorkSpaces(0)

Set ERwinDatabase = ERwinWorkspace.OpenDatabase("db.mdb")

' CREATE TABLE Заявки
Set ERwinTableDef = ERwinDatabase.CreateTableDef("Заявки")
Set ERwinField = ERwinTableDef.CreateField("ID", DB_LONG)
ERwinField.Attributes = ERwinField.Attributes + DB_AUTOINCRFIELD
ERwinField.Required = True
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Дата_подачи",
DB_DATETIME)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Подразделение", DB_LONG)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Тип_неисправности",
DB_LONG)
ERwinField.DefaultValue = "0"
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Оборудование", DB_LONG)
```

```

ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Заявитель", DB_LONG)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Статус", DB_LONG)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Исполнитель", DB_LONG)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Срочность", DB_TEXT, 255)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Комментарий", DB_MEMO)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Работы", DB_LONG)
ERwinTableDef.Fields. Append ERwinField
ERwinDatabase.TableDefs. Append ERwinTableDef
Set ERwinField = ERwinTableDef.Fields("ID")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "ID:")
Set ERwinField = ERwinTableDef.Fields("Дата_подачи")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Дата_подачи:")
Set ERwinField = ERwinTableDef.Fields("Подразделение")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Подразделение:")
Set ERwinField = ERwinTableDef.Fields("Тип_неисправности")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Тип_неисправности:")
Set ERwinField = ERwinTableDef.Fields("Оборудование")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Оборудование:")
Set ERwinField = ERwinTableDef.Fields("Заявитель")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Заявитель:")
Set ERwinField = ERwinTableDef.Fields("Статус")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Статус:")
Set ERwinField = ERwinTableDef.Fields("Исполнитель")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Исполнитель:")
Set ERwinField = ERwinTableDef.Fields("Срочность")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Срочность:")

```

```
Set ERwinField = ERwinTableDef.Fields("Комментарий")
Call SetFieldProp (ERwinField, "Caption", DB_ТЕХТ, "Комментарий:")
Set ERwinField = ERwinTableDef.Fields("Работы")
Call SetFieldProp (ERwinField, "Caption", DB_ТЕХТ, "Работы:")
```

```
' CREATE INDEX PrimaryKey
```

```
Set ERwinTableDef = ERwinDatabase.TableDefs("Заявки")
Set ERwinIndex = ERwinTableDef.CreateIndex("PrimaryKey")
Set ERwinField = ERwinIndex.CreateField("ID")
ERwinIndex.Fields. Append ERwinField
ERwinIndex.Primary = True
ERwinTableDef.Indexes. Append ERwinIndex
```

```
' CREATE INDEX Дата_подачи
```

```
Set ERwinTableDef = ERwinDatabase.TableDefs("Заявки")
Set ERwinIndex = ERwinTableDef.CreateIndex("Дата_подачи")
Set ERwinField = ERwinIndex.CreateField("Дата_подачи")
ERwinIndex.Fields. Append ERwinField
ERwinTableDef.Indexes. Append ERwinIndex
```

```
' CREATE INDEX Срочность
```

```
Set ERwinTableDef = ERwinDatabase.TableDefs("Заявки")
Set ERwinIndex = ERwinTableDef.CreateIndex("Срочность")
Set ERwinField = ERwinIndex.CreateField("Срочность")
ERwinIndex.Fields. Append ERwinField
ERwinTableDef.Indexes. Append ERwinIndex
```

```
' CREATE TABLE Сотрудники
```

```

Set ERwinTableDef = ERwinDatabase.CreateTableDef("Сотрудники")
Set ERwinField = ERwinTableDef.CreateField("ID", DB_LONG)
ERwinField.Attributes = ERwinField.Attributes + DB_AUTOINCRFIELD
ERwinField.Required = True
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Фамилия", DB_TEXT, 255)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Имя", DB_TEXT, 255)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Отчество", DB_TEXT, 255)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Признак", DB_TEXT, 255)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Логин", DB_TEXT, 255)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Пароль", DB_TEXT, 255)
ERwinTableDef.Fields. Append ERwinField
ERwinDatabase.TableDefs. Append ERwinTableDef
Set ERwinField = ERwinTableDef.Fields("ID")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "ID:")
Set ERwinField = ERwinTableDef.Fields("Фамилия")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Фамилия:")
Set ERwinField = ERwinTableDef.Fields("Имя")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Имя:")
Set ERwinField = ERwinTableDef.Fields("Отчество")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Отчество:")
Set ERwinField = ERwinTableDef.Fields("Признак")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Признак:")
Set ERwinField = ERwinTableDef.Fields("Логин")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Логин:")
Set ERwinField = ERwinTableDef.Fields("Пароль")
Call SetFieldProp (ERwinField, "Caption", DB_TEXT, "Пароль:")

```

```
' CREATE INDEX PrimaryKey
Set ERwinTableDef = ERwinDatabase.TableDefs("Сотрудники")
Set ERwinIndex = ERwinTableDef.CreateIndex("PrimaryKey")
Set ERwinField = ERwinIndex.CreateField("ID")
ERwinIndex.Fields. Append ERwinField
ERwinIndex.Primary = True
ERwinTableDef.Indexes. Append ERwinIndex
```

```
' CREATE INDEX Имя
Set ERwinTableDef = ERwinDatabase.TableDefs("Сотрудники")
Set ERwinIndex = ERwinTableDef.CreateIndex("Имя")
Set ERwinField = ERwinIndex.CreateField("Имя")
ERwinIndex.Fields. Append ERwinField
ERwinTableDef.Indexes. Append ERwinIndex
```

```
' CREATE INDEX Признак
Set ERwinTableDef = ERwinDatabase.TableDefs("Сотрудники")
Set ERwinIndex = ERwinTableDef.CreateIndex("Признак")
Set ERwinField = ERwinIndex.CreateField("Признак")
ERwinIndex.Fields. Append ERwinField
ERwinTableDef.Indexes. Append ERwinIndex
```

```
' CREATE INDEX Фамилия
Set ERwinTableDef = ERwinDatabase.TableDefs("Сотрудники")
Set ERwinIndex = ERwinTableDef.CreateIndex("Фамилия")
Set ERwinField = ERwinIndex.CreateField("Фамилия")
ERwinIndex.Fields. Append ERwinField
```


ERwinTableDef.Indexes. Append ERwinIndex

```
' CREATE TABLE Справочник_неисправностей
Set ERwinTableDef =
ERwinDatabase.CreateTableDef("Справочник_неисправностей")
Set ERwinField = ERwinTableDef.CreateField("ID", DB_LONG)
ERwinField.Attributes = ERwinField.Attributes + DB_AUTOINCRFIELD
ERwinField.Required = True
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Наименование", DB_TEXT,
255)
ERwinTableDef.Fields. Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("оборудование", DB_LONG)
ERwinField.DefaultValue = "0"
ERwinTableDef.Fields.Append ERwinField
ERwinDatabase.TableDefs.Append ERwinTableDef
Set ERwinField = ERwinTableDef.Fields("ID")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "ID:")
Set ERwinField = ERwinTableDef.Fields("Наименование")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Наименование:")
Set ERwinField = ERwinTableDef.Fields("оборудование")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "оборудование:")

' CREATE INDEX PrimaryKey
Set ERwinTableDef =
ERwinDatabase.TableDefs("Справочник_неисправностей")
Set ERwinIndex = ERwinTableDef.CreateIndex("PrimaryKey")
Set ERwinField = ERwinIndex.CreateField("ID")
ERwinIndex.Fields.Append ERwinField
ERwinIndex.Primary = True
```

ERwinTableDef.Indexes.Append ERwinIndex

' CREATE INDEX Наименование

Set ERwinTableDef =

ERwinDatabase.TableDefs("Справочник_неисправностей")

Set ERwinIndex = ERwinTableDef.CreateIndex("Наименование")

Set ERwinField = ERwinIndex.CreateField("Наименование")

ERwinIndex.Fields.Append ERwinField

ERwinTableDef.Indexes.Append ERwinIndex

' CREATE TABLE Справочник_оборудования

Set ERwinTableDef =

ERwinDatabase.CreateTableDef("Справочник_оборудования")

Set ERwinField = ERwinTableDef.CreateField("ID", DB_LONG)

ERwinField.Attributes = ERwinField.Attributes + DB_AUTOINCRFIELD

ERwinField.Required = True

ERwinTableDef.Fields.Append ERwinField

Set ERwinField = ERwinTableDef.CreateField("Наименование", DB_TEXT,

255)

ERwinTableDef.Fields.Append ERwinField

ERwinDatabase.TableDefs.Append ERwinTableDef

Set ERwinField = ERwinTableDef.Fields("ID")

Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "ID:")

Set ERwinField = ERwinTableDef.Fields("Наименование")

Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Наименование:")

' CREATE INDEX ID

Set ERwinTableDef = ERwinDatabase.TableDefs("Справочник_оборудования")

Set ERwinIndex = ERwinTableDef.CreateIndex("ID")

```
Set ERwinField = ERwinIndex.CreateField("ID")
ERwinIndex.Fields.Append ERwinField
ERwinIndex.Primary = True
ERwinTableDef.Indexes.Append ERwinIndex
```

```
' CREATE INDEX Наименование
```

```
Set ERwinTableDef = ERwinDatabase.TableDefs("Справочник_оборудования")
Set ERwinIndex = ERwinTableDef.CreateIndex("Наименование")
Set ERwinField = ERwinIndex.CreateField("Наименование")
ERwinIndex.Fields.Append ERwinField
ERwinTableDef.Indexes.Append ERwinIndex
```

```
' CREATE TABLE Справочник_подразделений
```

```
Set ERwinTableDef = ERwinDatabase.CreateTableDef("Справочник_подразделений")
Set ERwinField = ERwinTableDef.CreateField("ID", DB_LONG)
ERwinField.Attributes = ERwinField.Attributes + DB_AUTOINCRFIELD
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Наименование", DB_TEXT,
255)
ERwinTableDef.Fields.Append ERwinField
ERwinDatabase.TableDefs.Append ERwinTableDef
Set ERwinField = ERwinTableDef.Fields("ID")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "ID:")
Set ERwinField = ERwinTableDef.Fields("Наименование")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Наименование:")
```

```
' CREATE INDEX PrimaryKey
```

```

Set                                     ERwinTableDef                               =
ERwinDatabase.TableDefs("Справочник_подразделений")
Set ERwinIndex = ERwinTableDef.CreateIndex("PrimaryKey")
Set ERwinField = ERwinIndex.CreateField("ID")
ERwinIndex.Fields.Append ERwinField
ERwinIndex.Primary = True
ERwinTableDef.Indexes.Append ERwinIndex

' CREATE INDEX Наименование
Set                                     ERwinTableDef                               =
ERwinDatabase.TableDefs("Справочник_подразделений")
Set ERwinIndex = ERwinTableDef.CreateIndex("Наименование")
Set ERwinField = ERwinIndex.CreateField("Наименование")
ERwinIndex.Fields.Append ERwinField
ERwinTableDef.Indexes.Append ERwinIndex

' CREATE TABLE Справочник_работ
Set ERwinTableDef = ERwinDatabase.CreateTableDef("Справочник_работ")
Set ERwinField = ERwinTableDef.CreateField("ID", DB_LONG)
ERwinField.Attributes = ERwinField.Attributes + DB_AUTOINCRFIELD
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Наименование", DB_TEXT,
255)
ERwinTableDef.Fields.Append ERwinField
ERwinDatabase.TableDefs.Append ERwinTableDef
Set ERwinField = ERwinTableDef.Fields("ID")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "ID:")
Set ERwinField = ERwinTableDef.Fields("Наименование")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Наименование:")

```

```
' CREATE INDEX PrimaryKey
Set ERwinTableDef = ERwinDatabase.TableDefs("Справочник_работ")
Set ERwinIndex = ERwinTableDef.CreateIndex("PrimaryKey")
Set ERwinField = ERwinIndex.CreateField("ID")
ERwinIndex.Fields.Append ERwinField
ERwinIndex.Primary = True
ERwinTableDef.Indexes.Append ERwinIndex
```

```
' CREATE TABLE Справочник_статусов
Set ERwinTableDef = ERwinDatabase.CreateTableDef("Справочник_статусов")
Set ERwinField = ERwinTableDef.CreateField("ID", DB_LONG)
ERwinField.Attributes = ERwinField.Attributes + DB_AUTOINCRFIELD
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Наименование", DB_TEXT,
```

255)

```
ERwinTableDef.Fields.Append ERwinField
ERwinDatabase.TableDefs.Append ERwinTableDef
Set ERwinField = ERwinTableDef.Fields("ID")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "ID:")
Set ERwinField = ERwinTableDef.Fields("Наименование")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Наименование:")
```

```
' CREATE INDEX PrimaryKey
Set ERwinTableDef = ERwinDatabase.TableDefs("Справочник_статусов")
Set ERwinIndex = ERwinTableDef.CreateIndex("PrimaryKey")
Set ERwinField = ERwinIndex.CreateField("ID")
ERwinIndex.Fields.Append ERwinField
```

```
ERwinIndex.Primary = True
ERwinTableDef.Indexes.Append ERwinIndex
```

```
' CREATE INDEX Наименование
Set ERwinTableDef = ERwinDatabase.TableDefs("Справочник_статусов")
Set ERwinIndex = ERwinTableDef.CreateIndex("Наименование")
Set ERwinField = ERwinIndex.CreateField("Наименование")
ERwinIndex.Fields.Append ERwinField
ERwinTableDef.Indexes.Append ERwinIndex
```

```
' CREATE VIEW V_zajav
Set ERwinQueryDef = ERwinDatabase.CreateQueryDef(V_zajav)
ERwinQueryDef.SQL = SELECT Заявки.Дата_подачи, Заявки.Комментарий,
a1.Фамилия AS Техник, a2.Фамилия AS Заявитель,
Справочник_неисправностей.Наименование AS Неисправность,
Справочник_оборудования.Наименование AS Оборудование,
Справочник_подразделений.Наименование AS Подразделение,
Справочник_работ.Наименование AS Работы, Справочник_статусов.Наименование
AS Статус
FROM Справочник_статусов, Справочник_работ,
Справочник_подразделений, Справочник_оборудования,
Справочник_неисправностей, Сотрудники AS a1, Сотрудники AS a2, Заявки
WHERE Заявки.Исполнитель=[a1]. [ID] AND
Заявки.Заявитель=[a2]. [ID] AND
Заявки.Тип_неисправности=[Справочник_неисправностей]. [ID] AND
Заявки.Оборудование=[Справочник_оборудования]. [ID] AND
Заявки.Подразделение=[Справочник_подразделений]. [ID] AND
Заявки.Работы=[Справочник_работ]. [ID] AND
Заявки.Статус=[Справочник_статусов]. [ID]
```

```

' CREATE VIEW v2
Set ERwinQueryDef = ERwinDatabase.CreateQueryDef(v2)
ERwinQueryDef.SQL = SELECT Заявки.Дата_подачи, Заявки.Комментарий,
a1. Фамилия AS Техник, a2. Фамилия AS Заявитель, Справочник_неисправностей.
Наименование AS Неисправность, Справочник_оборудования. Наименование AS
Оборудование, Справочник_подразделений. Наименование AS Подразделение,
Справочник_работ. Наименование AS Работы, Справочник_статусов.
Наименование AS Статус
FROM Справочник_статусов, Справочник_работ,
Справочник_подразделений, Справочник_оборудования,
Справочник_неисправностей, Сотрудники AS a1, Сотрудники AS a2, Заявки
WHERE Заявки.Исполнитель=[a1]. [ID] AND
Заявки.Заявитель=[a2]. [ID] AND
Заявки.Тип_неисправности=[Справочник_неисправностей]. [ID] AND
Заявки.Оборудование=[Справочник_оборудования]. [ID] AND
Заявки.Подразделение=[Справочник_подразделений]. [ID] AND
Заявки.Работы=[Справочник_работ]. [ID] AND
Заявки.Статус=[Справочник_статусов]. [ID]

```

```

' CREATE VIEW v3
Set ERwinQueryDef = ERwinDatabase.CreateQueryDef(v3)
ERwinQueryDef.SQL = SELECT Заявки.Дата_подачи, Сотрудники.Фамилия
AS Исполнитель, Справочник_неисправностей.Наименование AS Неисправность,
Справочник_оборудования.Наименование AS Оборудование,
Справочник_работ.Наименование AS Работы, Сотрудники_1.Фамилия AS
Заявитель, Справочник_подразделений.Наименование AS Подразделение,
Справочник_статусов.Наименование AS Статус, Заявки.Комментарий AS coment,
Заявки.ID, Заявки.Подразделение AS a2, Заявки.Тип_неисправности AS a3,
Заявки.Оборудование AS a4, Заявки.Заявитель AS a5, Заявки.Исполнитель AS a6,
Заявки.Статус AS a7, Заявки.Работы AS a8

```

```

FROM Справочник_статусов INNER JOIN (Справочник_работ INNER JOIN
(Справочник_подразделений INNER JOIN (Справочник_оборудования INNER
JOIN (Справочник_неисправностей INNER JOIN (Сотрудники INNER JOIN
(Сотрудники AS Сотрудники_1 INNER JOIN Заявки ON Сотрудники_1.ID =
Заявки.Заявитель) ON Сотрудники.ID = Заявки.Исполнитель) ON
Справочник_неисправностей.ID = Заявки.Тип_неисправности) ON
Справочник_оборудования.ID = Заявки.Оборудование) ON
Справочник_подразделений.ID = Заявки.Подразделение) ON Справочник_работ.ID
= Заявки.Работы) ON Справочник_статусов.ID = Заявки.Статус

```

```

' CREATE VIEW Запрос1
Set ERwinQueryDef = ERwinDatabase.CreateQueryDef(Запрос1)
ERwinQueryDef.SQL = "SELECT [Заявки].[Дата_подачи] AS
Дата_подачи,[Заявки].[Комментарий] AS Комментарий,[Сотрудники].[Фамилия]
AS Фамилия,[Сотрудники].[Фамилия] AS
a1.Фамилия,[Справочник_неисправностей].[Наименование] AS
a2.Фамилия,[Справочник_оборудования].[Наименование] AS
Справочник_неисправностей.Наименование,[Справочник_подразделений].[Наимен
ование] AS
Справочник_оборудования.Наименование,[Справочник_работ].[Наименование] AS
Справочник_подразделений.Наименование,[Справочник_статусов].[Наименование
] AS Справочник_работ.Наименование FROM
[Справочник_статусов],[Справочник_работ],[Справочник_подразделений],[Справо
чник_оборудования],[Справочник_неисправностей],[Сотрудники],[Сотрудники],[З
аявки] WHERE Заявки.Исполнитель = a1.ID AND Заявки.Заявитель = a2.ID AND
Заявки.Тип_неисправности = Справочник_неисправностей.ID AND
Заявки.Оборудование = Справочник_оборудования.ID AND Заявки.Подразделение
= Справочник_подразделений.ID AND Заявки.Работы = Справочник_работ.ID AND
Заявки.Статус = Справочник_статусов.ID"

```



```
' CREATE VIEW Оборудование_неиспр
Set ERwinQueryDef = ERwinDatabase.CreateQueryDef(Оборудование_неиспр)
ERwinQueryDef.SQL = SELECT Справочник_неисправностей.ID,
Справочник_неисправностей.Наименование,
Справочник_неисправностей.оборудование,
Справочник_оборудования.Наименование AS НОборудование
FROM Справочник_оборудования INNER JOIN Справочник_неисправностей
ON Справочник_оборудования.ID = Справочник_неисправностей.оборудование
```

```
' CREATE VIEW Заявки Запрос
Set ERwinQueryDef = ERwinDatabase.CreateQueryDef(Заявки Запрос)
ERwinQueryDef.SQL = SELECT Заявки.ID, Заявки.Дата_подачи,
Заявки.Подразделение, Заявки.Тип_неисправности, Заявки.Оборудование,
Заявки.Заявитель, Заявки.Статус, Заявки.Исполнитель, Заявки.Срочность,
Заявки.Комментарий, Заявки.Работы
FROM Заявки
```

```
' CREATE RELATIONSHIP СотрудникиЗаявки

Set ERwinRelation = ERwinDatabase.CreateRelation(СотрудникиЗаявки,
Сотрудники, Заявки)

Set ERwinField = ERwinRelation.CreateField(«ID»)
ERwinField.ForeignName = Заявитель
ERwinRelation.Fields.Append ERwinField
ERwinDatabase.Relations.Append ERwinRelation
```

```
' CREATE RELATIONSHIP СотрудникиЗаявки1
```

```
Set ERwinRelation = ERwinDatabase.CreateRelation(СотрудникиЗаявки1,  
Сотрудники, Заявки)
```

```
Set ERwinField = ERwinRelation.CreateField(«ID»)
```

```
ERwinField.ForeignName = Исполнитель
```

```
ERwinRelation.Fields.Append ERwinField
```

```
ERwinDatabase.Relations.Append ERwinRelation
```

```
' CREATE RELATIONSHIP Справочник_неисправностейЗаявки
```

```
Set ERwinRelation =  
ERwinDatabase.CreateRelation(Справочник_неисправностейЗаявки,  
Справочник_неисправностей, Заявки)
```

```
Set ERwinField = ERwinRelation.CreateField(«ID»)
```

```
ERwinField.ForeignName = Тип_неисправности
```

```
ERwinRelation.Fields.Append ERwinField
```

```
ERwinDatabase.Relations.Append ERwinRelation
```

```
' CREATE RELATIONSHIP Справочник_оборудованияЗаявки
```

```
Set ERwinRelation =  
ERwinDatabase.CreateRelation(Справочник_оборудованияЗаявки,  
Справочник_оборудования, Заявки)
```

```
Set ERwinField = ERwinRelation.CreateField(«ID»)
```

```
ERwinField.ForeignName = Оборудование
```

```
ERwinRelation.Fields.Append ERwinField
```

```
ERwinDatabase.Relations.Append ERwinRelation
```

```
' CREATE RELATIONSHIP Справочник_подразделенийЗаявки
```

```

Set                                     ERwinRelation                             =
ERwinDatabase.CreateRelation(Справочник_подразделенийЗаявки,
Справочник_подразделений, Заявки)
Set ERwinField = ERwinRelation.CreateField(«ID»)
ERwinField.ForeignName = Подразделение
ERwinRelation.Fields.Append ERwinField
ERwinDatabase.Relations.Append ERwinRelation

' CREATE RELATIONSHIP Справочник_работЗаявки

Set ERwinRelation = ERwinDatabase.CreateRelation(Справочник_работЗаявки,
Справочник_работ, Заявки)
Set ERwinField = ERwinRelation.CreateField(«ID»)
ERwinField.ForeignName = Работы
ERwinRelation.Fields.Append ERwinField
ERwinDatabase.Relations.Append ERwinRelation

' CREATE RELATIONSHIP Справочник_статусовЗаявки

Set                                     ERwinRelation                             =
ERwinDatabase.CreateRelation(Справочник_статусовЗаявки, Справочник_статусов,
Заявки)
Set ERwinField = ERwinRelation.CreateField(«ID»)
ERwinField.ForeignName = Статус
ERwinRelation.Fields.Append ERwinField
ERwinDatabase.Relations.Append ERwinRelation

ERwinDatabase.Close
ERwinWorkspace.Close

```

' Terminating Access Basic DAO Session...

ПРИЛОЖЕНИЕ 2. ЛИСТИНГ ПРОГРАММЫ

```
unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ComCtrls, DB, ADODB, ExtCtrls, jpeg;

type

TForm1 = class (TForm)
    ADOConnection1: TADOConnection;
    Timer1: TTimer;
    Edit1: TEdit;
    ADOTable1: TADOTable;
    ADOTable2: TADOTable;
    ADOTable1_date: TDateTimeField;
    ADOTable1DSDesigner: TWideStringField;
    ADOTable1DSDesigner2: TWideStringField;
    ADOTable1DSDesigner3: TWideStringField;
    ADOTable1DSDesigner4: TWideStringField;
    ADOTable1DSDesigner5: TWideStringField;
    ADOTable1DSDesigner6: TWideStringField;
    ADOTable1DSDesigner7: TWideStringField;
    ADOTable1ID: TAutoIncField;
    ADOTable1a2_podr: TIntegerField;
    ADOTable1a4_neispr: TIntegerField;
    ADOTable1a4_obor: TIntegerField;
    ADOTable1a5_zajavitel: TIntegerField;
    ADOTable1a6_ispol: TIntegerField;
    ADOTable1a7_stat: TIntegerField;
```

```

ADOTable1a8_rab: TIntegerField;
ADOTable1coment: TMemofield;
Button1: TButton;
Label1: TLabel;
Label2: TLabel;

procedure Timer1Timer (Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button1Click (Sender: TObject);
procedure FormClose (Sender: TObject; var Action: TCloseAction);
private
  {Private declarations}
public
  pr,id:integer;
  login,pass:string;
  filtr1, filtr2: string;
  path1, path:string;

  {Public declarations }
end;

var
  Form1: TForm1;

implementation

uses Unit2, Unit4, Unit7, Unit8, Unit9;

{$R *.dfm}

procedure TForm1.Timer1Timer (Sender: TObject);
begin

```

```

form1.Timer1.Enabled:=false;
form1.close;
end;

procedure TForm1.FormCreate(Sender: TObject);

begin
try
path:=getcurrentdir;
edit1.Text:=path+'\DB.mdb';
form1.ADOConnection1.ConnectionString:='Provider=Microsoft.Jet.
OLEDB.4.0; Password="«; Data Source='+edit1.Text+'; Persist Security Info=True ';
form1.ADOConnection1.Connected:=true;

except on exception do
begin
form1.Timer1.Enabled:=false;
showMessage ('Укажите строку соединения');
form1.Button1.Enabled:=true;
end;
end;
// form1.ADOConnection1.Connected:=false;

end;

procedure TForm1.Button1Click (Sender: TObject);
begin
form1.ADOConnection1.ConnectionString:='Provider=Microsoft.Jet.
OLEDB.4.0; Password="«; Data Source='+edit1.Text+'; Persist Security Info=True ';

```

```

form1.ADOConnection1. Connected: =true;
form1.close;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
form1.ADOTable1. Open;
form1.ADOTable2. Open;

//Сделать для фсех тейблов на всех формах

end;

end.
{$A8, B-, C+, D+, E-, F-, G+, H+, I+, J-, K-, L+, M-, N+, O+, P+, Q-, R-, S-, T-,
U-, V+, W-, X+, Y+, Z1}
{$MINSTACKSIZE $00004000}
{$MAXSTACKSIZE $00100000}
{$IMAGEBASE $00400000}
{$APPTYPE GUI}
{$WARN SYMBOL_DEPRECATED ON}
{$WARN SYMBOL_LIBRARY ON}
{$WARN SYMBOL_PLATFORM ON}
{$WARN UNIT_LIBRARY ON}
{$WARN UNIT_PLATFORM ON}
{$WARN UNIT_DEPRECATED ON}
{$WARN HRESULT_COMPAT ON}
{$WARN HIDING_MEMBER ON}
{$WARN HIDDEN_VIRTUAL ON}
{$WARN GARBAGE ON}
{$WARN BOUNDS_ERROR ON}
{$WARN ZERO_NIL_COMPAT ON}

```


{ \$WARN STRING_CONST_TRUNCED ON }
{ \$WARN FOR_LOOP_VAR_VARPAR ON }
{ \$WARN TYPED_CONST_VARPAR ON }
{ \$WARN ASG_TO_TYPED_CONST ON }
{ \$WARN CASE_LABEL_RANGE ON }
{ \$WARN FOR_VARIABLE ON }
{ \$WARN CONSTRUCTING_ABSTRACT ON }
{ \$WARN COMPARISON_FALSE ON }
{ \$WARN COMPARISON_TRUE ON }
{ \$WARN COMPARING_SIGNED_UNSIGNED ON }
{ \$WARN COMBINING_SIGNED_UNSIGNED ON }
{ \$WARN UNSUPPORTED_CONSTRUCT ON }
{ \$WARN FILE_OPEN ON }
{ \$WARN FILE_OPEN_UNITSRC ON }
{ \$WARN BAD_GLOBAL_SYMBOL ON }
{ \$WARN DUPLICATE_CTOR_DTOR ON }
{ \$WARN INVALID_DIRECTIVE ON }
{ \$WARN PACKAGE_NO_LINK ON }
{ \$WARN PACKAGED_THREADVAR ON }
{ \$WARN IMPLICIT_IMPORT ON }
{ \$WARN HPPEMIT_IGNORED ON }
{ \$WARN NO_RETVAL ON }
{ \$WARN USE_BEFORE_DEF ON }
{ \$WARN FOR_LOOP_VAR_UNDEF ON }
{ \$WARN UNIT_NAME_MISMATCH ON }
{ \$WARN NO_CFG_FILE_FOUND ON }
{ \$WARN MESSAGE_DIRECTIVE ON }
{ \$WARN IMPLICIT_VARIANTS ON }
{ \$WARN UNICODE_TO_LOCALE ON }
{ \$WARN LOCALE_TO_UNICODE ON }
{ \$WARN IMAGEBASE_MULTIPLE ON }
{ \$WARN SUSPICIOUS_TYPECAST ON }

```

{$WARN PRIVATE_PROPACCESSOR ON}
{$WARN UNSAFE_TYPE OFF}
{$WARN UNSAFE_CODE OFF}
{$WARN UNSAFE_CAST OFF}
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ExtCtrls, Grids, DBGrids, DB, ADODB, ComCtrls, Buttons,
  DBCtrls, StdCtrls, RpRender, RpRenderCanvas, RpRenderPreview, RpDefine,
  RpCon, RpConDS;

type
  TForm2 = class (TForm)
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    DataSource1: TDataSource;
    DataSource2: TDataSource;
    N10: TMenuItem;
    Panel1: TPanel;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    DBGrid1: TDBGrid;
  end;

```

DBGrid2: TDBGrid;
SpeedButton1: TSpeedButton;
N14: TMenuItem;
N15: TMenuItem;
N16: TMenuItem;
N17: TMenuItem;
N18: TMenuItem;
N19: TMenuItem;
N20: TMenuItem;
N21: TMenuItem;
N22: TMenuItem;
N23: TMenuItem;
DBNavigator2: TDBNavigator;
N5: TMenuItem;
N9: TMenuItem;
SpeedButton3: TSpeedButton;
N12: TMenuItem;
OpenDialog1: TOpenDialog;
SpeedButton2: TSpeedButton;
DBNavigator1: TDBNavigator;
procedure FormShow (Sender: TObject);
procedure N3Click (Sender: TObject);
procedure N5Click (Sender: TObject);
procedure N7Click (Sender: TObject);
procedure N8Click (Sender: TObject);
procedure N9Click (Sender: TObject);
procedure N6Click (Sender: TObject);
procedure N10Click (Sender: TObject);
procedure SpeedButton1Click (Sender: TObject);
procedure SpeedButton4Click (Sender: TObject);
procedure N15Click (Sender: TObject);
procedure N16Click (Sender: TObject);

```

procedure SpeedButton2Click (Sender: TObject);
procedure SpeedButton3Click (Sender: TObject);
procedure N12Click (Sender: TObject);
private
  {Private declarations}
public
  {Public declarations}
end;

var
  Form2: TForm2;

implementation

uses Unit1, Unit3, Unit4, Unit5, Unit7, Unit8, Unit9, Unit13;

{$R *.dfm}

procedure TForm2.FormShow(Sender: TObject);
begin

  form1.showmodal;
  form3.showmodal;

end;

procedure TForm2.N3Click (Sender: TObject);
begin
  if form1.pr<3 then
    showMessage ('Сюда доступ закрыт')
  else
    form4.showmodal;

```

end;

procedure TForm2.N5Click (Sender: TObject);

begin

form5.showmodal;

end;

procedure TForm2.N7Click (Sender: TObject);

begin

if form1.pr<3 then

 showMessage ('Сюда доступ закрыт')

else

 form7.showmodal;

end;

procedure TForm2.N8Click (Sender: TObject);

begin

if form1.pr<3 then

 showMessage ('Сюда доступ закрыт')

else

 form8.showmodal;

end;

procedure TForm2.N9Click (Sender: TObject);

begin

Form5.ShowModal;

end;

procedure TForm2.N6Click (Sender: TObject);

begin

Form13.fz: =false;

```

form13.showmodal;
end;

procedure TForm2.N10Click (Sender: TObject);
begin
Form13.fz: =true;
Form13.ShowModal;
end;

procedure TForm2.SpeedButton1Click (Sender: TObject);
begin
Form13.fz: =false;
form13.showmodal;
end;

procedure TForm2.SpeedButton4Click (Sender: TObject);
begin
Form13.fz: =true;
Form13.ShowModal;
end;

procedure TForm2.N15Click (Sender: TObject);
var sz: string;
begin
if PageControl1.TabIndex=0
then
begin
sz: =InputBox ('Поиск','Поиск по полю '+DBGrid1.SelectedField.
FieldName,");
if sz<>"
then
begin

```

```

        Form1.ADOTable1. Filtered: =false;
        Form1.ADOTable1. Filter: =DBGrid1.SelectedField. FieldName+' LIKE
''+sz+'%'';
        Form1.ADOTable1. Filtered: =true;
    end;
end;
if PageControl1.TabIndex=1
then
begin
    sz: =InputBox ('Поиск','Поиск по полю '+DBGrid2.SelectedField.
FieldName,");
    if sz<>"
    then
    begin
        Form1.ADOTable2. Filtered: =false;
        Form1.ADOTable2. Filter: =DBGrid2.SelectedField. FieldName+' LIKE
''+sz+'%'';
        Form1.ADOTable2. Filtered: =true;
    end;
end;

end;

procedure TForm2.N16Click (Sender: TObject);
begin
    if PageControl1.TabIndex=0
    then
        Form1.ADOTable1. Filtered: =false
    else
        Form1.ADOTable2. Filtered: =false;
end;

```

```

procedure TForm2.SpeedButton2Click (Sender: TObject);
begin
Form13.fz: =true;
Form13.ShowModal;
end;

procedure TForm2.SpeedButton3Click (Sender: TObject);
var sz: string;
begin
if PageControl1.TabIndex=0
then
begin
sz: =InputBox ('Поиск','Поиск по полю '+DBGrid1.SelectedField.
FieldName,");
if sz<>"
then
begin
Form1.ADOTable1. Filtered: =false;
Form1.ADOTable1. Filter: =DBGrid1.SelectedField. FieldName+' LIKE
"+sz+'%";
Form1.ADOTable1. Filtered: =true;
end;
end;
if PageControl1.TabIndex=1
then
begin
sz: =InputBox ('Поиск','Поиск по полю
'+DBGrid2.SelectedField.FieldName,");
if sz<>"
then
begin
Form1.ADOTable2. Filtered: =false;

```



```

        Form1.ADOTable2. Filter: =DBGrid2.SelectedField. FieldName+' LIKE
"+sz+'%";
        Form1.ADOTable2. Filtered: =true;
    end;
end;
end;

procedure TForm2.N12Click (Sender: TObject);
begin
if form1.pr<3 then
    showMessage ('Сюда доступ закрыт')
else
    form9.showmodal;
end;

end.

unit Unit3;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons, Mask, ExtCtrls, unit1, DB, ADODB, Grids,
DBGrids, DBClient;

type
TForm3 = class (TForm)
    GroupBox1: TGroupBox;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    RadioButton3: TRadioButton;
    Panel1: TPanel;

```

```

Edit1: TEdit;
Label1: TLabel;
Label2: TLabel;
BitBtn1: TBitBtn;
ADOQuery1: TADOQuery;
Edit2: TEdit;
procedure BitBtn1Click (Sender: TObject);
procedure FormClose (Sender: TObject; var Action: TCloseAction);
private
  {Private declarations}
public
  {Public declarations}
end;

var
  Form3: TForm3;

implementation

uses Unit2;

{$R *.dfm}

procedure TForm3.BitBtn1Click (Sender: TObject);

begin
  form3.ADOQuery1. Close;
  form3.ADOQuery1. Parameters.ParamByName('log'). Value: =edit1.Text;
  form3.ADOQuery1. Parameters.ParamByName('pass'). Value: =edit2.Text;
  if form3.RadioButton1. Checked then form1.pr: =1;
  if form3.RadioButton2. Checked then form1.pr: =2;
  if form3.RadioButton3. Checked then form1.pr: =3;

```

```

form3.ADOQuery1. Parameters.ParamByName('pr'). Value: =form1.pr;
form3.ADOQuery1. Open;
if form3.ADOQuery1. Recordset.RecordCount<1 then
begin
  SHowMessage ('Неверно введен пароль');
  form1.pr: =0;
  form1.login: =";
  form1.pass: =";
end
else
begin
  form1.login: =edit1.Text;
  form1.pass: =edit2.Text;
  form1.id: =form3.ADOQuery1. FieldValues['id'];
  if form1.pr=1 then
  begin
    Form2.MainMenu1. Items [2]. Enabled: =false;
    form1.filtr1: =' Статус<>"Выполнено" and a5='+inttostr(form1.id);
    form1.ADOTable1. Filtered: =false;
    form1.ADOTable1. Filter: =form1.filtr1;
    form1.ADOTable1. Filtered: =true;
    form1.filtr2: =' Статус="Выполнено" and a5='+inttostr(form1.id);
    form1.ADOTable2. Filtered: =false;
    form1.ADOTable2. Filter: =form1.filtr2;
    form1.ADOTable2. Filtered: =true;
  end
  else
  if form1.pr=2 then
  begin
    Form1.filtr1: =' Статус<>"Выполнено" and a6='+inttostr(form1.id);
    form1.ADOTable1. Filtered: =false;
    form1.ADOTable1. Filter: =form1.filtr1;

```

```
form1.ADOTable1. Filtered: =true;
form1.filtr2: =' Статус="Выполнено" and a6='+inttostr(form1.id);
form1.ADOTable2. Filtered: =false;
form1.ADOTable2. Filter: =form1.filtr2;
form1.ADOTable2. Filtered: =true;
end;
```

```
form3.Close;
end;
```

```
end;
```

```
procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);
begin
if form1.pr=0 then form2.Close;
end;
```

```
end.
```

```
unit Unit4;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, unit1, DB, ADODB, ExtCtrls, DBCtrls, Grids, DBGrids, Buttons,
StdCtrls, Menus;
```

```
type
```

```
TForm4 = class (TForm)
  ADOTable1: TADOTable;
  DataSource1: TDataSource;
  DBGrid1: TDBGrid;
  Panel1: TPanel;
  MainMenu1: TMainMenu;
  N1: TMenuItem;
  DBNavigator1: TDBNavigator;
  SpeedButton6: TSpeedButton;
  SpeedButton5: TSpeedButton;
  N2: TMenuItem;
  N3: TMenuItem;
  N4: TMenuItem;
  ADOTable1ID: TAutoIncField;
  ADOTable1DSDesigner: TWideStringField;
  ADOTable1DSDesigner2: TWideStringField;
  ADOTable1DSDesigner3: TWideStringField;
  ADOTable1DSDesigner4: TWideStringField;
  ADOTable1DSDesigner5: TWideStringField;
  ADOTable1DSDesigner6: TWideStringField;
  N10: TMenuItem;
  N11: TMenuItem;
  N12: TMenuItem;
  N13: TMenuItem;
  N14: TMenuItem;
  N15: TMenuItem;
  N16: TMenuItem;
  N17: TMenuItem;
  N5: TMenuItem;
  N6: TMenuItem;
  procedure SpeedButton5Click (Sender: TObject);
  procedure SpeedButton6Click (Sender: TObject);
```

```
procedure N16Click (Sender: TObject);
procedure N17Click (Sender: TObject);
procedure N15Click (Sender: TObject);
procedure N6Click (Sender: TObject);
procedure N12Click (Sender: TObject);
procedure N5Click (Sender: TObject);
procedure N13Click (Sender: TObject);
procedure N14Click (Sender: TObject);
procedure BitBtn1Click (Sender: TObject);
procedure FormActivate (Sender: TObject);
private
  {Private declarations}
public
  {Public declarations}
end;

var
  Form4: TForm4;

implementation

uses Unit6;

{$R *.dfm}

procedure TForm4.SpeedButton5Click (Sender: TObject);
begin
  form6.f: =false;
  form6.showmodal;

end;
```

```

procedure TForm4.SpeedButton6Click (Sender: TObject);
begin
    if Application.MessageBox(PChar ('Вы действительно хотите удалить запись
'+ADOTable1.DSDesigner.AsString),'Внимание!',
    MB_OKCANCEL) =id_OK then
        ADOTable1.Delete;
end;

procedure TForm4.N16Click (Sender: TObject);
begin
form4.ADOTable1. Sort: =DBGrid1.SelectedField. FieldName+' DESC';
end;

procedure TForm4.N17Click (Sender: TObject);
begin
form4.ADOTable1. Sort: =DBGrid1.SelectedField. FieldName;

end;

procedure TForm4.N15Click (Sender: TObject);
var s: string;
begin
    s: =InputBox ('Поиск','Поиск по полю '+DBGrid1.SelectedField. FieldName,");
    if s<>" then
        begin
            ADOTable1.Filtered: =false;
            ADOTable1.Filter: =DBGrid1.SelectedField. FieldName+' LIKE "'+s+'%";
            ADOTable1.Filtered: =true;
        end;
end;

procedure TForm4.N6Click (Sender: TObject);

```

```

begin
  ADOTable1.Filtered: =false;
end;

procedure TForm4.N12Click (Sender: TObject);
begin
  form4.Close
end;

procedure TForm4.N5Click (Sender: TObject);
begin
  form6.f: =false;
  form6.showmodal;
end;

procedure TForm4.N13Click (Sender: TObject);
begin
  form6.f: =true;
  form6.showmodal;
end;

procedure TForm4.N14Click (Sender: TObject);
begin
  if Application.MessageBox(PChar ('Вы действительно хотите удалить запись
'+ADOTable1DSDesigner.AsString),'Внимание!',
  MB_OKCANCEL) =id_OK then
    ADOTable1.Delete;
end;

procedure TForm4.BitBtn1Click (Sender: TObject);
begin
  form4.Close

```



```

end;

procedure TForm4.FormActivate(Sender: TObject);
begin
    form4.ADOTable1.Active:=true;
end;

end.

unit Unit5;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, unit1, StdCtrls, Buttons, DB, ADODB;

type
TForm5 = class (TForm)
    Edit1: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    BitBtn1: TBitBtn;
    Edit2: TEdit;
    ADOQuery1: TADOQuery;
    procedure BitBtn1Click (Sender: TObject);
private
    {Private declarations}
public
    {Public declarations}
end;

var

```

Form5: TForm5;

implementation

{ \$R *.dfm }

procedure TForm5.BitBtn1Click (Sender: TObject);

begin

if trim (form1.pass) <>trim (edit1.Text) then

 showMessage ('старый пароль не соответствует новому паролю')

else

 begin

 form5.ADOQuery1. Parameters.ParamByName('pas'). Value: =trim (edit2.Text);

 form5.ADOQuery1. Parameters.ParamByName('log'). Value: =trim

(form1.login);

 form5.ADOQuery1. ExecSQL;

 form5.Close

 end;

end;

end.

unit Unit6;

interface

uses

 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, StdCtrls, Mask, DBCtrls, Buttons, unit4, unit1, DB, ADODB;

type

 TForm6 = class (TForm)

 BitBtn2: TBitBtn;

```
Edit1: TEdit;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Edit2: TEdit;
Edit3: TEdit;
Edit5: TEdit;
Edit6: TEdit;
ComboBox1: TComboBox;
ADOQuery1: TADOQuery;
ADOQuery2: TADOQuery;
procedure FormShow (Sender: TObject);
procedure BitBtn2Click (Sender: TObject);
private
  {Private declarations}
public
  {Public declarations}
  f: boolean;
end;

var
  Form6: TForm6;

implementation

{$R *.dfm}

procedure TForm6.FormShow(Sender: TObject);
begin
```

```

if f then
begin
    form6.Edit1. Text: =form4.ADOTable1DSDesigner.AsString;
    form6.Edit2. Text: =form4.ADOTable1DSDesigner2.AsString;
    form6.Edit3. Text: =form4.ADOTable1DSDesigner3.AsString;
    form6.ComboBox1. ItemIndex: =form4.ADOTable1DSDesigner4.AsInteger-1;
    form6.Edit5. Text: =form4.ADOTable1DSDesigner5.AsString;
    form6.Edit6. Text: =form4.ADOTable1DSDesigner6.AsString;
end
else
begin
    form6.Edit1. Text: =";
    form6.Edit2. Text: =";
    form6.Edit3. Text: =";
    form6.ComboBox1. ItemIndex: =0;
    form6.Edit5. Text: =";
    form6.Edit6. Text: =";
end;

end;

procedure TForm6.BitBtn2Click (Sender: TObject);
begin
if f then
begin
    form6.ADOQuery1. Parameters.ParamByName('a1'). Value: =Edit1.Text;
    form6.ADOQuery1. Parameters.ParamByName('a2'). Value: =Edit2.Text;
    form6.ADOQuery1. Parameters.ParamByName('a3'). Value: =Edit3.Text;
    form6.ADOQuery1.           Parameters.ParamByName('a4').           Value:
=ComboBox1.ItemIndex+1;
    form6.ADOQuery1. Parameters.ParamByName('a5'). Value: =Edit5.Text;
    form6.ADOQuery1. Parameters.ParamByName('a6'). Value: =Edit6.Text;

```

```

        form6.ADOQuery1.      Parameters.ParamByName('ID').      Value:
=form4.ADOTable1ID.Value;
        form6.ADOQuery1. ExecSQL;
end
else
begin
        form6.ADOQuery2. Parameters.ParamByName('a1'). Value: =Edit1.Text;
        form6.ADOQuery2. Parameters.ParamByName('a2'). Value: =Edit2.Text;
        form6.ADOQuery2. Parameters.ParamByName('a3'). Value: =Edit3.Text;
        form6.ADOQuery2.      Parameters.ParamByName('a4').      Value:
=ComboBox1.ItemIndex+1;
        form6.ADOQuery2. Parameters.ParamByName('a5'). Value: =Edit5.Text;
        form6.ADOQuery2. Parameters.ParamByName('a6'). Value: =Edit6.Text;
        form6.ADOQuery2. ExecSQL;
end;
form4.ADOTable1. Active: =false;
form4.ADOTable1. Active: =true;
form6.Close;
end;

end.
unit Unit7;

interface

uses

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

type
TForm7 = class (TForm)

```

```
DBGrid1: TDBGrid;
MainMenu1: TMainMenu;
N1: TMenuItem;
Panel1: TPanel;
DBNavigator1: TDBNavigator;
SpeedButton1: TSpeedButton;
SpeedButton2: TSpeedButton;
N2: TMenuItem;
N3: TMenuItem;
N4: TMenuItem;
N6: TMenuItem;
N7: TMenuItem;
N8: TMenuItem;
N9: TMenuItem;
N10: TMenuItem;
N11: TMenuItem;
N12: TMenuItem;
N13: TMenuItem;
N14: TMenuItem;
ADOTable1: TADOTable;
DataSource1: TDataSource;
ADOTable1ID: TAutoIncField;
ADOTable1DSDesigner: TWideStringField;
N15: TMenuItem;
procedure SpeedButton1Click (Sender: TObject);
procedure N9Click (Sender: TObject);
procedure SpeedButton2Click (Sender: TObject);
procedure N11Click (Sender: TObject);
procedure N10Click (Sender: TObject);
procedure N6Click (Sender: TObject);
procedure N7Click (Sender: TObject);
procedure N8Click (Sender: TObject);
```

```

procedure N15Click (Sender: TObject);
procedure BitBtn1Click (Sender: TObject);
procedure N14Click (Sender: TObject);
procedure FormActivate (Sender: TObject);
private
  {Private declarations}
public
  {Public declarations}
end;

var
  Form7: TForm7;

implementation

uses Unit1, Unit10;

{$R *.dfm}

procedure TForm7.SpeedButton1Click (Sender: TObject);
begin
  form10.f1: =false;
  form10.showmodal;
end;

procedure TForm7.N9Click (Sender: TObject);
begin
  Form10.f1: =true;
  Form10.ShowModal;
end;

procedure TForm7.SpeedButton2Click (Sender: TObject);

```

```

begin
    if Application.MessageBox(PChar ('Вы действительно хотите удалить запись
'+ADOTable1DSDesigner.AsString),'Внимание!',
    MB_OKCANCEL) =id_OK then
        ADOTable1.Delete;
end;

procedure TForm7.N11Click (Sender: TObject);
begin
    Form10.f1: =false;
    Form10.ShowModal;
end;

procedure TForm7.N10Click (Sender: TObject);
begin
    if Application.MessageBox(PChar ('Вы действительно хотите удалить запись
'+ADOTable1DSDesigner.AsString),'Внимание!',
    MB_OKCANCEL) =id_OK then
        ADOTable1.Delete;
end;

procedure TForm7.N6Click (Sender: TObject);
begin
    form7.ADOTable1. Sort: =DBGrid1.SelectedField. FieldName+' DESC';
end;

procedure TForm7.N7Click (Sender: TObject);
begin
    form7.ADOTable1. Sort: =DBGrid1.SelectedField. FieldName;
end;

procedure TForm7.N8Click (Sender: TObject);

```



```

var sp: string;
begin
    sp:  =InputBox ('Поиск','Поиск по полю '+DBGrid1.SelectedField.
FieldName,");
    if sp<>" then
        begin
            ADOTable1.Filtered: =false;
            ADOTable1.Filter: =DBGrid1.SelectedField. FieldName+' LIKE "'+sp+'%";
            ADOTable1.Filtered: =true;
        end;
    end;

procedure TForm7.N15Click (Sender: TObject);
begin
    ADOTable1.Filtered: =false;
end;

procedure TForm7.BitBtn1Click (Sender: TObject);
begin
    form7.Close;
end;

procedure TForm7.N14Click (Sender: TObject);
begin
    form7.Close;
end;

procedure TForm7.FormActivate(Sender: TObject);
begin
    form7.ADOTable1. Active: =true;
end;

```

end.

unit Unit8;

interface

uses

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

type

TForm8 = class (TForm)

DBGrid1: TDBGrid;

MainMenu1: TMainMenu;

N1: TMenuItem;

Panel1: TPanel;

DBNavigator1: TDBNavigator;

SpeedButton1: TSpeedButton;

SpeedButton2: TSpeedButton;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N6: TMenuItem;

N7: TMenuItem;

N8: TMenuItem;

N9: TMenuItem;

N10: TMenuItem;

N11: TMenuItem;

N12: TMenuItem;

N13: TMenuItem;

N14: TMenuItem;

DataSource1: TDataSource;

```

ADOTable1: TADOTable;
ADOTable1ID: TAutoIncField;
ADOTable1DSDesigner: TWideStringField;
N15: TMenuItem;
procedure SpeedButton1Click (Sender: TObject);
procedure N11Click (Sender: TObject);
procedure N9Click (Sender: TObject);
procedure N10Click (Sender: TObject);
procedure SpeedButton2Click (Sender: TObject);
procedure N8Click (Sender: TObject);
procedure N15Click (Sender: TObject);
procedure N6Click (Sender: TObject);
procedure N7Click (Sender: TObject);
procedure SpeedButton4Click (Sender: TObject);
procedure BitBtn1Click (Sender: TObject);
procedure N14Click (Sender: TObject);
procedure FormActivate (Sender: TObject);
private
  {Private declarations}
public
  {Public declarations}
end;

var
  Form8: TForm8;

implementation

uses Unit1, Unit11, Unit9;

{$R *.dfm}

```

```

procedure TForm8.SpeedButton1Click (Sender: TObject);
begin
    Form11.f2: =false;
    Form11.ShowModal;
end;

procedure TForm8.N11Click (Sender: TObject);
begin
    Form11.f2: =false;
    Form11.ShowModal;
end;

procedure TForm8.N9Click (Sender: TObject);
begin
    Form11.f2: =true;
    Form11.ShowModal;
end;

procedure TForm8.N10Click (Sender: TObject);
begin
    if Application.MessageBox(PChar ('Вы действительно хотите удалить запись
'+ADOTable1DSDesigner.AsString),'Внимание!',
    MB_OKCANCEL) =id_OK then
        ADOTable1.Delete;
end;

procedure TForm8.SpeedButton2Click (Sender: TObject);
begin
    if Application.MessageBox(PChar ('Вы действительно хотите удалить запись
'+ADOTable1DSDesigner.AsString),'Внимание!',
    MB_OKCANCEL) =id_OK then
        ADOTable1.Delete;
end;

```

```

end;

procedure TForm8.N8Click (Sender: TObject);
var so: string;
begin
    so:  =InputBox ('Поиск','Поиск по полю '+DBGrid1.SelectedField.
FieldName,");
    if so<>" then
        begin
            ADOTable1.Filtered: =false;
            ADOTable1.Filter: =DBGrid1.SelectedField. FieldName+' LIKE "'+so+'%";
            ADOTable1.Filtered: =true;
        end;
end;

procedure TForm8.N15Click (Sender: TObject);
begin
    ADOTable1.Filtered: =false;
end;

procedure TForm8.N6Click (Sender: TObject);
begin
    form8.ADOTable1. Sort: =DBGrid1.SelectedField. FieldName+' DESC';
end;

procedure TForm8.N7Click (Sender: TObject);
begin
    form8.ADOTable1. Sort: =DBGrid1.SelectedField. FieldName;
end;

procedure TForm8.SpeedButton4Click (Sender: TObject);
begin

```

```

    form9.showmodal;
end;

procedure TForm8.BitBtn1Click (Sender: TObject);
begin
    form8.Close;
end;

procedure TForm8.N14Click (Sender: TObject);
begin
    form8.Close;
end;

procedure TForm8.FormActivate(Sender: TObject);
begin
    form8.ADOTable1.Active := true;
end;

end.
unit Unit9;

interface

uses

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

type
    TForm9 = class (TForm)
        DBGrid1: TDBGrid;
        MainMenu1: TMainMenu;

```

```

N1: TMenuItem;
Panel1: TPanel;
DBNavigator1: TDBNavigator;
SpeedButton1: TSpeedButton;
SpeedButton2: TSpeedButton;
N2: TMenuItem;
N4: TMenuItem;
N5: TMenuItem;
N6: TMenuItem;
N7: TMenuItem;
N8: TMenuItem;
N9: TMenuItem;
ADOTable1: TADOTable;
DataSource1: TDataSource;
ADOTable1ID: TAutoIncField;
ADOTable1nam: TWideStringField;
ADOTable1_obid: TIntegerField;
ADOTable1_nob: TWideStringField;
procedure SpeedButton1Click (Sender: TObject);
procedure N6Click (Sender: TObject);
procedure N4Click (Sender: TObject);
procedure N5Click (Sender: TObject);
procedure SpeedButton2Click (Sender: TObject);
procedure BitBtn1Click (Sender: TObject);
procedure N9Click (Sender: TObject);
procedure SpeedButton3Click (Sender: TObject);
procedure FormActivate (Sender: TObject);
private
  {Private declarations}
public
  {Public declarations}
end;

```

```

var
  Form9: TForm9;

implementation

uses Unit1, Unit12;

{$R *.dfm}

procedure TForm9.SpeedButton1Click (Sender: TObject);
begin
  Form12.f3: =false;
  Form12.ShowModal;
end;

procedure TForm9.N6Click (Sender: TObject);
begin
  Form12.f3: =false;
  Form12.ShowModal;
end;

procedure TForm9.N4Click (Sender: TObject);
begin
  Form12.f3: =true;
  Form12.ShowModal;
end;

procedure TForm9.N5Click (Sender: TObject);
begin
  if Application.MessageBox(PChar ('Вы действительно хотите удалить запись
'+form9.ADOTable1nam.AsString),'Внимание!',

```



```

    MB_OKCANCEL) =id_OK then
    ADOTable1.Delete;
end;

procedure TForm9.SpeedButton2Click (Sender: TObject);
begin
    if Application.MessageBox(PChar ('Вы действительно хотите удалить запись
'+form9.ADOTable1nam.AsString),'Внимание!',
    MB_OKCANCEL) =id_OK then
    ADOTable1.Delete;
end;

procedure TForm9.BitBtn1Click (Sender: TObject);
begin
    form9.Close;
end;

procedure TForm9.N9Click (Sender: TObject);
begin
    form9.Close;
end;

procedure TForm9.SpeedButton3Click (Sender: TObject);
begin
    Form12.f3: =true;
    Form12.ShowModal;
end;

procedure TForm9.FormActivate(Sender: TObject);
begin
    form9.ADOTable1.Active: =true;
end;

```

```

end.
unit Unit10;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, ADODB, StdCtrls, Buttons;

type
TForm10 = class (TForm)
Label1: TLabel;
Edit1: TEdit;
BitBtn1: TBitBtn;
ADOQuery1: TADOQuery;
ADOQuery2: TADOQuery;
procedure FormShow (Sender: TObject);
procedure BitBtn1Click (Sender: TObject);
private
{Private declarations}
public
{Public declarations}
f1: boolean;
end;

var
Form10: TForm10;

implementation

uses Unit1, Unit7;

```

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

```
procedure TForm10.FormShow(Sender: TObject);
```

```
begin
```

```
if f1
```

```
then
```

```
form10.Edit1.Text: =form7.ADOTable1DSDesigner.asstring
```

```
else
```

```
Form10.Edit1.Text: =";
```

```
end;
```

```
procedure TForm10.BitBtn1Click (Sender: TObject);
```

```
begin
```

```
if f1
```

```
then
```

```
begin
```

```
Form10.ADOQuery1.Parameters.ParamByName('ap'). Value: =Edit1.Text;
```

```
Form10.ADOQuery1. Parameters.ParamByName('ID'). Value:
```

```
=form7.ADOTable1ID.Value;
```

```
Form10.ADOQuery1.ExecSQL;
```

```
end
```

```
else
```

```
begin
```

```
Form10.ADOQuery2.Parameters.ParamByName('ap'). Value: =Edit1.Text;
```

```
Form10.ADOQuery2.ExecSQL;
```

```
end;
```

```
Form7.ADOTable1.Active: =false;
```

```
Form7.ADOTable1.Active: =true;
```

```
Form10.Close;
```

```
end;
```

```

end.
unit Unit11;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, DB, ADODB;

type
  TForm11 = class (TForm)
    Label1: TLabel;
    Edit1: TEdit;
    BitBtn1: TBitBtn;
    ADOQuery1: TADOQuery;
    ADOQuery2: TADOQuery;
    procedure FormShow (Sender: TObject);
    procedure BitBtn1Click (Sender: TObject);
  private
    {Private declarations}
  public
    {Public declarations}
    f2: boolean;
  end;

var
  Form11: TForm11;

implementation

uses Unit1, Unit8;

```

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

```
procedure TForm11.FormShow(Sender: TObject);
```

```
begin
```

```
  if f2
```

```
  then
```

```
    Form11.Edit1.Text: =form8.ADOTable1DSDesigner.asstring
```

```
  else
```

```
    Form11.Edit1.Text: ="";
```

```
end;
```

```
procedure TForm11.BitBtn1Click (Sender: TObject);
```

```
begin
```

```
  if f2
```

```
  then
```

```
    begin
```

```
      Form11.ADOQuery1.Parameters.ParamByName('ao').Value: =Edit1.Text;
```

```
      Form11.ADOQuery1.Parameters.ParamByName('ID').Value:
```

```
=form8.ADOTable1ID.Value;
```

```
      Form11.ADOQuery1.ExecSQL;
```

```
    end
```

```
  else
```

```
    begin
```

```
      Form11.ADOQuery2.Parameters.ParamByName('ao').Value: =Edit1.Text;
```

```
      Form11.ADOQuery2.ExecSQL;
```

```
    end;
```

```
  Form8.ADOTable1.Active: =false;
```

```
  Form8.ADOTable1.Active: =true;
```

```
  Form11.Close;
```

```
end;
```

```

end.
unit Unit12;

interface

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

type
  TForm12 = class (TForm)
    Label1: TLabel;
    Edit1: TEdit;
    BitBtn1: TBitBtn;
    ADOQuery1: TADOQuery;
    ADOQuery2: TADOQuery;
    Label2: TLabel;
    ADOTable1: TADOTable;
    DBLookupComboBox1: TDBLookupComboBox;
    DataSource1: TDataSource;
    procedure FormShow (Sender: TObject);
    procedure BitBtn1Click (Sender: TObject);
  private
    {Private declarations}
  public
    {Public declarations}
    f3: boolean;
  end;

var
  Form12: TForm12;

```

implementation

uses Unit1, Unit9;

{ \$R *.dfm }

procedure TForm12.FormShow(Sender: TObject);

begin

if f3

then

begin

Form12.Edit1. Text: =form9.ADOTable1nam.asstring;

form12.DBLookupComboBox1. KeyValue: =form9.ADOTable1_obid.

AsInteger;

end

else

Form12.Edit1. Text: =";

end;

procedure TForm12.BitBtn1Click (Sender: TObject);

begin

if f3

then

begin

Form12.ADOQuery1. Parameters.ParamByName('an'). Value: =Edit1.Text;

Form12.ADOQuery1. Parameters.ParamByName('ID'). Value:

=form9.ADOTable1ID.Value;

form12.ADOQuery1. Parameters.ParamByName('ob'). Value:

=form12.DBLookupComboBox1. KeyValue;

Form12.ADOQuery1. ExecSQL;

end

```

else
begin
    Form12.ADOQuery2.Parameters.ParamByName('an'). Value: =Edit1.Text;
    form12.ADOQuery2.        Parameters.ParamByName('ob').        Value:
=form12.DBLookupComboBox1. KeyValue;
    Form12.ADOQuery2. ExecSQL;
end;
Form9.ADOTable1. Active: =false;
Form9.ADOTable1. Active: =true;
Form12.Close;
end;

```

end.

```

{$A8, B-, C+, D+, E-, F-, G+, H+, I+, J-, K-, L+, M-, N+, O+, P+, Q-, R-, S-, T-,
U-, V+, W-, X+, Y+, Z1}

```

```

{$MINSTACKSIZE $00004000}
{$MAXSTACKSIZE $00100000}
{$IMAGEBASE $00400000}
{$APPTYPE GUI}
{$WARN SYMBOL_DEPRECATED ON}
{$WARN SYMBOL_LIBRARY ON}
{$WARN SYMBOL_PLATFORM ON}
{$WARN UNIT_LIBRARY ON}
{$WARN UNIT_PLATFORM ON}
{$WARN UNIT_DEPRECATED ON}
{$WARN HRESULT_COMPAT ON}
{$WARN HIDING_MEMBER ON}
{$WARN HIDDEN_VIRTUAL ON}
{$WARN GARBAGE ON}
{$WARN BOUNDS_ERROR ON}
{$WARN ZERO_NIL_COMPAT ON}
{$WARN STRING_CONST_TRUNCED ON}

```


{ \$WARN FOR_LOOP_VAR_VARPAR ON }
{ \$WARN TYPED_CONST_VARPAR ON }
{ \$WARN ASG_TO_TYPED_CONST ON }
{ \$WARN CASE_LABEL_RANGE ON }
{ \$WARN FOR_VARIABLE ON }
{ \$WARN CONSTRUCTING_ABSTRACT ON }
{ \$WARN COMPARISON_FALSE ON }
{ \$WARN COMPARISON_TRUE ON }
{ \$WARN COMPARING_SIGNED_UNSIGNED ON }
{ \$WARN COMBINING_SIGNED_UNSIGNED ON }
{ \$WARN UNSUPPORTED_CONSTRUCT ON }
{ \$WARN FILE_OPEN ON }
{ \$WARN FILE_OPEN_UNITSRC ON }
{ \$WARN BAD_GLOBAL_SYMBOL ON }
{ \$WARN DUPLICATE_CTOR_DTOR ON }
{ \$WARN INVALID_DIRECTIVE ON }
{ \$WARN PACKAGE_NO_LINK ON }
{ \$WARN PACKAGED_THREADVAR ON }
{ \$WARN IMPLICIT_IMPORT ON }
{ \$WARN HPPEMIT_IGNORED ON }
{ \$WARN NO_RETVAL ON }
{ \$WARN USE_BEFORE_DEF ON }
{ \$WARN FOR_LOOP_VAR_UNDEF ON }
{ \$WARN UNIT_NAME_MISMATCH ON }
{ \$WARN NO_CFG_FILE_FOUND ON }
{ \$WARN MESSAGE_DIRECTIVE ON }
{ \$WARN IMPLICIT_VARIANTS ON }
{ \$WARN UNICODE_TO_LOCALE ON }
{ \$WARN LOCALE_TO_UNICODE ON }
{ \$WARN IMAGEBASE_MULTIPLE ON }
{ \$WARN SUSPICIOUS_TYPECAST ON }
{ \$WARN PRIVATE_PROPACCESSOR ON }

```

{$WARN UNSAFE_TYPE OFF}
{$WARN UNSAFE_CODE OFF}
{$WARN UNSAFE_CAST OFF}
unit Unit13;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, unit1, DB, ADODB, DBCtrls, ComCtrls, StdCtrls, Buttons;

type
  TForm13 = class (TForm)
    Label1: TLabel;
    DateTimePicker1: TDateTimePicker;
    DBLookupComboBox1: TDBLookupComboBox;
    DataSource1: TDataSource;
    ADOTable1: TADOTable;
    Label2: TLabel;
    ADOTable2: TADOTable;
    DataSource2: TDataSource;
    Label3: TLabel;
    Label4: TLabel;
    ADOTable3: TADOTable;
    DataSource3: TDataSource;
    DBLookupComboBox2: TDBLookupComboBox;
    DBLookupComboBox3: TDBLookupComboBox;
    Label5: TLabel;
    ADOTable4: TADOTable;
    DataSource4: TDataSource;
    DBLookupComboBox4: TDBLookupComboBox;
    Label6: TLabel;
  end;

```

```

ADOTable5: TADOTable;
DataSource5: TDataSource;
DBLookupComboBox5: TDBLookupComboBox;
ADOTable6: TADOTable;
Label7: TLabel;
DataSource6: TDataSource;
DBLookupComboBox6: TDBLookupComboBox;
Label8: TLabel;
Memo1: TMemo;
BitBtn1: TBitBtn;
ADOQuery1: TADOQuery;
ADOQuery2: TADOQuery;
ADOTable1ID: TAutoIncField;
ADOTable1DSDesigner: TWideStringField;
ADOTable2ID: TAutoIncField;
ADOTable2DSDesigner: TWideStringField;
ADOTable2DSDesigner2: TIntegerField;
procedure BitBtn1Click (Sender: TObject);
procedure DBLookupComboBox2Click (Sender: TObject);
procedure FormShow (Sender: TObject);
procedure FormClose (Sender: TObject; var Action: TCloseAction);
procedure FormCreate (Sender: TObject);
private
  {Private declarations}
public
  {Public declarations}
  fz: boolean;
end;

var
  Form13: TForm13;

```

implementation

uses Unit2;

{ \$R *.dfm }

procedure TForm13.BitBtn1Click (Sender: TObject);

begin

if fz

then

begin

Form13.ADOQuery2. Parameters.ParamByName('a1'). Value: =datetostr
(form13.DateTimePicker1. Date);

Form13.ADOQuery2. Parameters.ParamByName('a2'). Value:
=form13.DBLookupComboBox1. KeyValue;

form13.ADOQuery2. Parameters.ParamByName('a3'). Value:
=form13.DBLookupComboBox3. KeyValue;

form13.ADOQuery2. Parameters.ParamByName('a4'). Value:
=form13.DBLookupComboBox2. KeyValue;

form13.ADOQuery2. Parameters.ParamByName('a5'). Value: =form1.id;

form13.ADOQuery2. Parameters.ParamByName('a6'). Value:
=form13.DBLookupComboBox5. KeyValue;

form13.ADOQuery2. Parameters.ParamByName('a7'). Value:
=form13.DBLookupComboBox4. KeyValue;

form13.ADOQuery2. Parameters.ParamByName('a8'). Value: ='g';

form13.ADOQuery2. Parameters.ParamByName('a9'). Value:
=form13.Memo1. Text;

form13.ADOQuery2. Parameters.ParamByName('a10'). Value:
=form13.DBLookupComboBox6. KeyValue;

Form13.ADOQuery2. Parameters.ParamByName('id'). Value:
=Form1.ADOTable1ID.AsInteger;

Form13.ADOQuery2. ExecSQL;

```

end
else
begin
{
Заявки.Дата_подачи,
Заявки.Подразделение,
Заявки.Тип_неисправности,
Заявки.Оборудование,
Заявки.Заявитель,
Заявки.Статус,
Заявки.Исполнитель,
Заявки.Срочность,
Заявки.Комментарий,
Заявки.Работы
}
form13.ADOQuery1. Parameters.ParamByName('a1'). Value: =datetostr
(form13.DateTimePicker1. Date);
form13.ADOQuery1. Parameters.ParamByName('a2'). Value:
=form13.DBLookupComboBox1. KeyValue;
form13.ADOQuery1. Parameters.ParamByName('a3'). Value:
=form13.DBLookupComboBox3. KeyValue;
form13.ADOQuery1. Parameters.ParamByName('a4'). Value:
=form13.DBLookupComboBox2. KeyValue;
form13.ADOQuery1. Parameters.ParamByName('a5'). Value: =form1.id;
form13.ADOQuery1. Parameters.ParamByName('a6'). Value:
=form13.DBLookupComboBox5. KeyValue;
form13.ADOQuery1. Parameters.ParamByName('a7'). Value:
=form13.DBLookupComboBox4. KeyValue;
form13.ADOQuery1. Parameters.ParamByName('a8'). Value: ='g';
form13.ADOQuery1. Parameters.ParamByName('a9'). Value: =form13.Memo1.
Text;

```

```

        form13.ADOQuery1.      Parameters.ParamByName('a10').      Value:
=form13.DBLookupComboBox6. KeyValue;
        form13.ADOQuery1. ExecSQL;

end;
Form1.ADOTable1. Active: =false;
Form1.ADOTable1. Active: =true;
Form13.Close;
end;

procedure TForm13.DBLookupComboBox2Click (Sender: TObject);
begin
form13.ADOTable2. Active: =false;
form13.ADOTable2.  Filter:  ='  [Оборудование]  =  '+'  inttostr
(form13.DBLookupComboBox2. KeyValue);
form13.ADOTable2. Filtered: =true;
form13.ADOTable2. Active: =true;
end;

procedure TForm13.FormShow(Sender: TObject);
begin
if fz
then
begin
Form13.DateTimePicker1. Date: =Form1.ADOTable1_date. AsDateTime;
Form13.DBLookupComboBox1.  KeyValue:  =Form1.ADOTable1a2_podr.
AsInteger;
Form13.DBLookupComboBox2.  KeyValue:  =Form1.ADOTable1a4_obor.
AsInteger;
Form13.DBLookupComboBox3.  KeyValue:  =Form1.ADOTable1a4_neispr.
AsInteger;

```

```

        Form13.DBLookupComboBox4. KeyValue: =Form1.ADOTable1a6_ispol.
AsInteger;
        Form13.DBLookupComboBox5. KeyValue: =Form1.ADOTable1a7_stat.
AsInteger;
        Form13.DBLookupComboBox6. KeyValue: =Form1.ADOTable1a8_rab.
AsInteger;
        Form13.Memo1. Text: =Form1.ADOTable1coment.AsVariant;
end;

end;

procedure TForm13.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Form1.ADOTable2. Close;
Form1.ADOTable2. Open;
end;

procedure TForm13.FormCreate(Sender: TObject);
begin
form13.ADOTable1. Active: =true;
form13.ADOTable2. Active: =true;
form13.ADOTable3. Active: =true;
form13.ADOTable4. Active: =true;
form13.ADOTable5. Active: =true;
form13.ADOTable6. Active: =true;
end;

end.
unit Unit14;

interface

```

uses

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

type

TForm14 = class (TForm)

private

{Private declarations}

public

{Public declarations}

end;

var

Form14: TForm14;

implementation

{ \$R *.dfm }

end.

ПРИЛОЖЕНИЕ 3. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Техническое задание

Введение

Эффективное управление персоналом одна из важнейших задач руководителя подразделения организации. Основными критериями эффективности управления подразделением обслуживания вычислительной техники является оперативность реагирования сотрудников на заявки пользователей, возможность адекватного распределения трудовых ресурсов, контроль выполнения заявок.

Основания для разработки

Разработка программы велась по заказу руководства Муниципального автономного учреждения дополнительного образования «Центр дополнительного образования детей» г.Усинска.

Назначение разработки

Обеспечение автоматизации управления отделом обслуживания вычислительной техники.

Требования к программе или программному изделию

Требования к функциональным характеристикам

Программа должна обладать защитой от несанкционированного доступа. Программа должна поддерживать возможность ввода и хранения нормативно-справочной информации в виде справочников. Программа должна давать возможности пользователя делать заявки на ремонт информационно-вычислительной техники и программного обеспечения. Программа должна обладать возможностью сотрудников отдела получать заявки, реагировать на них. Программа должна обеспечивать руководителя отдела контролировать процесс выполнения заявок пользователей.

Требования к надежности

Программа должна обладать средствами защиты целостности данных. Программа должна обладать средствами разграничения прав доступа пользователей. Для предотвращения потери данных в случае разрыва соединения с базой данных, программа должна сохранять результаты каждой операции при её успешном окончании. Программа должна обладать средствами обработки ошибок с выводом корректного сообщения пользователю.

Требования к эксплуатации

Эксплуатация программы, установленной на персональном компьютере, должна осуществляться с использованием требований по эксплуатации компьютерной техники, с применением мер безопасности, согласно ГОСТ Р МЭК 60950-20002 Безопасность оборудования информационных технологий. Место использования программного обеспечения должно быть оборудовано согласно СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» (в редакции СанПиН 2.2.2/2.4.2198-07. Изменения N1 к СанПиН 2.2.2/2.4.1340).

Обслуживающий персонал, должен состоять из одного квалифицированного системного администратора.

Требования к составу и параметрам технических средств

Программное обеспечение разрабатывается для персональной ЭВМ (IBM PC-совместимой) со следующими характеристиками: процессор с частотой не ниже 1 ГГц; объем ОЗУ не менее 128 Мб; графический адаптер SVGA; стандартная клавиатура. манипулятор типа «мышь» или совместимый с ним. Сетевой адаптер (при использовании программы в сетевом режиме). Объем дискового пространства для программного модуля не менее 4Мб. Объем дискового пространства, занимаемого базой данных, зависит от количества хранящейся в ней информации.

Требования к информационной и программной совместимости

Минимальные требования. Операционная система Windows 2000/XP/2003/Vista/7. Библиотека доступа к данным MSJect 4.0. Драйверы и утилиты для поддержки устройств.

Требования к программной документации

Инструкция обслуживающему персоналу. Инструкция пользователя.

Технико-экономические показатели

По сравнению программными продуктами, выполняющими аналогичные функции, данная система обладает следующими преимуществами. Низкой стоимостью. Быстротой развёртывания. Простой переносимостью. Низкими системными требованиями. Простотой использования.

Стадии и этапы разработки

Изучение предприятия. Выделение автоматизируемых бизнес-процессов. Разработка идеи программного продукта. Ознакомление со стандартами и протоколами, анализ схожих существующих программных средств. Разработка технического задания. Согласование технического задания с заказчиком, внесение корректировок. Разработка концептуальной модели функционирования будущей программы. Разработка эскизного проекта программного средства и согласование его с заказчиком. Непосредственная разработка законченного программного средства (рабочий проект). Внедрение.

Разработка ИС учета и контроля выполнения заявок на техническое и программное обслуживание компьютерной техники

Фамилия Имя Отчество
Номер группы

Презентация



Цели и задачи, актуальность

- “ **Цель:** Разработка ИС учета и контроля выполнения заявок на техническое и программное обслуживание компьютерной техники
- “ **Задачи:**
 - “ Изучить предметную область.
 - “ Изучить бизнес-процесс приёма заявок, выявить его слабые стороны.
 - “ Выделить задачи, требующие разработки бизнес-процессов.
 - “ Обосновать необходимость использования вычислительной техники.
 - “ Разработать методику автоматизации бизнес-процессов
 - “ Выбрать средства решения задачи.
 - “ Разработать техническое задание на создание автоматизированной системы.
 - “ Разработать базу данных.
 - “ Разработать программный модуль.
 - “ Разработать интерфейс программного продукта.
 - “ Разработать техническую документацию по работе с программным продуктом.
- “ **Актуальность:** Эффективность организации службы поддержки и обслуживания информационно вычислительной техники и программного обеспечения.



Причины выбора

- “ Возможность формирования собственной информационной системы с базой знаний по решаемым проблемам конкретной предметной области.
- “ Нежелание зависеть от сторонних разработчиков программного обеспечения.



Требования к системе

- “ Программа должна поддерживать возможность ввода и хранения нормативно-справочной информации в виде справочников.
- “ Программа должна давать возможности пользователя делать заявки на ремонт оборудования.
- “ Программа должна обладать возможностью сотрудников отдела получать заявки, реагировать на них.
- “ Программа должна обеспечивать руководителя отдела контролировать процесс выполнения заявок пользователей.



Выбор среды реализации

- “ Среда разработки программных модулей информационной системы – Delphi 7 Studio.
- “ СУБД – MS Access
- “ Технические характеристики рабочей станции:
 - “ Процессор Athlon - II X3 450 (3.2 GHz)
 - “ Чипсет - nForce630
 - “ Память - DIMM 2GB
 - “ Жёсткий диск - HDD 500GB
 - “ Оптический привод - DVD±RW/CR



Проект системы



Структуры баз данных



Вид программы

The image displays a software application interface for managing requests. It consists of several windows:

- Вход (Login):** A window for user authentication. It includes a dropdown for "Тип учётной записи" (Account type) with options: "Пользователь" (User), "Техник" (Technician), and "Дипетчер" (Dispatcher). Below are input fields for "Логин:" (admin) and "Пароль:" (masked with xxx), and a "Вход" button.
- Система редактирования заявок (Request Management System):** The main application window. It features a menu bar with "Файл", "Задачи", "Справочники", "Поиск", and "Сортировка". Below the menu is a toolbar with icons for home, search, print, and navigation. A tabbed interface shows "Активные заявки" (Active requests) and "Архив заявок" (Request archive). A table displays a list of requests with columns: "Дата подачи" (Submission date), "Заявитель" (Applicant), "Неисправность" (Malfunction), "Оборудование" (Equipment), "Работы" (Works), "Исполнитель" (Executor), "Подразделение" (Department), and "Статус" (Status).

Дата подачи	Заявитель	Неисправность	Оборудование	Работы	Исполнитель	Подразделение	Статус
10.02.2019	Иванов	Ошибки в работе	Оборудование 1	Настройка подключен	Васильева	Бухгалтерия	На рассмотрении
22.02.2019	Иванов	Устройство не работает	Оборудование 1	Настройка подключен	Васильева	Бухгалтерия	На рассмотрении
22.02.2019	Иванов	Отсутствие доступа к сет	Оборудование 1	Настройка подключен	Васильева	Бухгалтерия	На рассмотрении
- Редактирование заявок (Request Editing):** A window for editing request details. It contains dropdown menus for "Дата:" (10.04.2016), "Место неисправности:", "Оборудование:", "Тип неисправности:", "Техник:", "Статус:", and "Работы:". There is a text area for "Комментарии:" and a "Сохранить" button.
- Редактирование записи (Record Editing):** A window for editing user records. It includes input fields for "Фамилия:", "Имя:", "Отчество:", "Логин:", and "Пароль:". The "Доступ:" field is a dropdown menu currently set to "Пользователь". A "Сохранить" button is at the bottom.

Экономический расчет

Затраты

Наименование статьи расходов	Сумма, руб.
Материалы	442,00
Командировочные расходы	2790,00
Контрагентские расходы	8475,00
Покупные изделия	820,00
Расходы, связанные с оплатой труда	169309,09

Результаты внедрения

- “ Увеличение скорости обработки заявки.
- “ Повешение качества работы информационно-технического персонала.
- “ Централизованный контроль выполнения заявок.
- “ Выполнение заявок в соответствии с приоритетами.
- “ Накопление знаний.
- “ Экономический эффект 560 160 руб. в год



Выводы

- “ Проанализированы бизнес-процессы распределения заявок, выявленные его слабые стороны, разработаны методы по его усовершенствованию.
- “ Выбраны средства разработки, которые будут использоваться при разработки автоматизированной системы.
- “ Разработано техническое задание.
- “ Разработан интерфейс пользователей для автоматизации создания заявок - требований по устранению неисправностей информационно вычислительной техники и программного обеспечения.
- “ Разработан инструмент диспетчера для распределения заявок по исполнителям из инженерно-технического обслуживающего персонала.
- “ Разработан интерфейс мастера (инженера) для получения заявок и в работу.
- “ Разработан интерфейс для обслуживания разрабатываемой системы её настройки и ввода всех необходимых для правильного функционирования данных.
- “ Разработана структура БД системы автоматизации.
- “ Разработана техническая документация по работе информационной системой.
- “ **Все поставленные цели были выполнены. Программное обеспечение было отлажено и запущено в тестовую эксплуатацию.**



Конец

Спасибо за внимание



Слайд 1

Уважаемый председатель и члены Государственной аттестационной комиссии! Вашему вниманию представляется дипломный проект на тему **«Разработка ИС учета и контроля выполнения заявок на техническое и программное обслуживание компьютерной техники».**

Основной целью деятельности Муниципального ресурсного центра по работе с одаренными детьми является совершенствование работы по выявлению, поддержке и развитию одаренных и талантливых детей на территории МО ГО «Усинск»

Работа информационно-технической службы МАУДО "ЦДОД" (в рамках обслуживания информационно-вычислительной техники) сводится к приёму и выполнению заявок выполнения заявок на техническое и программное обслуживание компьютерной техники.

В рамках данной работы было поручено изучить систему поддержки пользователей, определить, возможно, ли её усовершенствование и целесообразность решения данной задачи, и, в случае положительного ответа, произвести необходимые мероприятия.

Слайд 2

Рассмотрев данную задачу можно сделать вывод, что данная область нуждается в автоматизации, которая должна состоять в уменьшении количества промежуточных пунктов, сообщением сотрудников и устранением неисправности, уменьшении количества преобразований информации, ускорении процесса, уменьшении трудозатрат и исключении человеческого фактора, как главного источника ошибок.

Таким образом была сформированная главная цель работы - Разработка ИС учета и контроля выполнения заявок на техническое и программное обслуживание компьютерной техники.

Для достижения цели работы необходимо решить указанный перечень задач.

Слайд 3.

В рамках решения данной задачи были рассмотрены такие способы приобретения системы автоматизации как покупка готового решения, заказ разработки у сторонних производителей и собственная разработка программного обеспечения. В результате анализа решений была выбрана собственная разработка, поскольку такое решение позволяет не зависеть от сторонних разработчиков программного обеспечения, обеспечить возможность доработки «под себя». Кроме того, предприятие является закрытым, и возможность внедрения сторонних программ ограничено службой безопасности.

Слайд 4

Для разработки программного продукта было составлено и утверждено техническое задание, в одном из пунктов к разрабатываемой системе были предъявлены следующие требования:

- Программа должна поддерживать возможность ввода и хранения нормативно-справочной информации в виде справочников.
- Программа должна давать возможности пользователя делать заявки на ремонт оборудования.
- Программа должна обладать возможностью сотрудников отдела получать заявки, реагировать на них.
- Программа должна обеспечивать руководителя отдела контролировать процесс выполнения заявок пользователей.

Слайд 5

Программа разрабатывалась на оборудовании отдела и с применением интегрированной среды разработки Delphi 7 и СУБД MS Access.

Слайд 6

Разработанная программа состоит из клиентского модуля и серверной части (базы данных).

На данном слайде представлены дерево модулей физической реализации программного модуля и логическое дерево функциональности.

Слайд 7

На данном слайде представлена структура базы данных программного продукта. База построена с применением принципов реляционной теории баз данных и нормализована до третьей нормальной формы.

Слайд 8

Скриншоты программы приложены на данных рисунках. Более подробно с формами программного продукта, а так же с методикой их использования при решении задач, вы можете ознакомиться в тексте пояснительной записки.

Слайд 9

Экономический расчёт затрат показал, что суммарные расходы на разработку программного продукта затрчено более 200000 руб., в то время как доход от внедрения программного продукта составляет более 500000. Таким образом, разработка и внедрение программного продукта является, безусловно, рентабельной.

Кроме денежного выражения положительного эффекта были достигнуты следующие положительные моменты:

- Увеличение скорости обработки заявки.
- Повешение качества работы информационно-технического персонала.
- Централизованный контроль выполнения заявок.
- Выполнение заявок в соответствии с приоритетами.
- Накопление знаний.

Слайд 10.

В результате проделанной работы все обозначенные задачи были решены и цель проектирования достигнута.

Слайд11

Доклад закончен. Спасибо за внимание. Я бы хотел выразить благодарность моему научному руководителю (ФИО), рецензенту (ФИО) и всем, кто учувствовал.

Теперь я готов ответить на ваши вопросы.