

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ЛЕСОСИБИРСКИЙ ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ –  
филиал Сибирского федерального университета

Высшей математики, информатики и естествознания  
кафедра

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

09.03.02 Информационные системы и технологии

**РАЗРАБОТКА БАЗЫ ДАННЫХ ПО УЧЕТУ РАБОТЫ СТО**  
**(НА ПРИМЕРЕ «АВТОТЕХЦЕНТР»)**

Руководитель

  
подпись

С.С. Ахтамова  
инициалы, фамилия

Студент

  
подпись

А.В. Мальцев  
инициалы, фамилия

Лесосибирск 2019

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ЛЕСОСИБИРСКИЙ ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ –  
филиал Сибирского федерального университета

Высшей математики, информатики и естествознания  
кафедра

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

09.03.02 Информационно-управляющие системы

РАЗРАБОТКА БАЗЫ ДАННЫХ ПО УЧЕТУ РАБОТЫ СТО  
(НА ПРИМЕРЕ «АВТОТЕХЦЕНТР»)

Работа защищена «24» июня \_\_\_\_\_ 2019 г. с оценкой «удовлетворительно»

Председатель ГЭК

Кучумов  
подпись

Кучумов Е.Г.  
инициалы, фамилия

Члены ГЭК

Захарова  
подпись

Т.В. Захарова  
инициалы, фамилия

Киргизова  
подпись

Е.В. Киргизова  
инициалы, фамилия

Степанов  
подпись

А.А. Степанов  
инициалы, фамилия

Фирер  
подпись

В.В. Фирер  
инициалы, фамилия

Руководитель

Ахтамова  
подпись

С.С. Ахтамова  
инициалы, фамилия

Выпускник

Мальцев  
подпись

А.В. Мальцев  
инициалы, фамилия

Лесосибирск 2019

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка базы данных по учету работы СТО (на примере «Автотехцентр»)».

Выпускная квалификационная работа состоит из введения, 2 глав, выводов, списка использованных источников, состоящего из 35 источников, 3 приложения. Работа изложена на 68 листах печатного текста, содержит 33 рисунка.

Целью данной работы является разработка и внедрение базы данных по учету работы станции технического обслуживания.

В соответствии с целью необходимо решить следующие задачи:

- провести анализ научной и специальной литературы по теме исследования;
- раскрыть теоретические основы баз данных;
- спроектировать организационную и процессную модели СТО;
- разработать базу данных организации с использованием Borland Delphi.

В ходе выполнения выпускной квалификационной работы проанализирована специальная и научная литература по теме исследования.

В результате проведения исследования был создан комплекс программных приложений для станции технического обслуживания с использованием СУБД Microsoft Office Access и ООП Delphi 7.0.

В работе показан поэтапный процесс проектирования программного обеспечения, начиная от постановки задачи, разработки процессной модели данных до проектирования организационной системы и их реализация на языке программирования.

## СОДЕРЖАНИЕ

Ведение.....	5
1 Теоретические основы базы данных.....	7
1.1 Определение и основные модели базы данных.....	7
1.2 Информационная безопасность, виды угроз и защита информации.....	19
1.3 Организационная модель СТО.....	21
1.4 Процессная модель СТО.....	23
2 Разработка программного продукта .....	27
2.1 Техническое задание на разработку.....	27
2.2 Обоснования выбора Microsoft Office Access.....	30
2.3 Обоснования выбора Borland Delphi 7.0.....	34
2.4 Интерфейс конечного пользователя.....	39
Заключение.....	48
Список использованных источников.....	49
Приложение А (листинг программных модулей).....	52
Приложение Б (очерк научной статьи).....	63
Приложение В (сертификат о публикации).....	68

## ВВЕДЕНИЕ

Автомобильный транспорт представляет собой отрасль народного хозяйства, обладает значительными резервами повышения эффективности использования подвижного состава, кроме этого выполняет важную роль для экономики нашей страны.

Во время использования транспортного средства постепенно снижаются его эксплуатационные свойства из-за изнашивания материалов, появления коррозии, повреждения и поломок деталей и узлов, усталости свойств материала. Это является причиной поломок, которые в итоге сокращают эффективность использования транспорта. Для того чтобы снизить вероятность появления дефектов и своевременно устранить поломки создается станция технического обслуживания (ТО) и ремонта оборудования.

Рост производственно-технической базы прочно связано со строительством новых, расширением, реконструкцией и техническим перевооружением существующих предприятий. Техническое развитие, поднятие жизненного уровня, растущие требования владельцев автомобилей в большей степени вытесняют ту привычную практику, когда сам водитель начинает заниматься техническим обслуживанием, проверкой и уходом за автомобилем. Широкое распространение получает представление о том, что уход, определение неисправностей и их устранение является задачами станции технического обслуживания.

Для рыночной экономики необходима минимизация себестоимости любой продукции, для преодоления конкурентоспособности по ценам на ремонт. Участных владельцев автомобилей не хватает свободного времени на их ремонт – заработки в условиях рынка даются все труднее, в большинстве случаев время уходит для обеспечения нормальных условий жизни. Этот сектор бытового обслуживания населения активно прогрессирует, но при этом остается довольно молодым и подвергается ряду серьезных проблем.

Выпускная квалификационная работа направлена на разработку базы данных по учету работы СТО (на примере «Автотехцентр»).

Цель работы – разработать базу данных по учету работы станции технического обслуживания.

Объектом исследования является – база данных станции технического обслуживания.

Предмет исследования – изучение особенностей и создание базы данных станции технического обслуживания.

Основные задачи:

- Проанализировать научную и специальную литературу по теме исследования.
- Раскрыть теоретические основы базы данных.
- Спроектировать организационную и процессную модели СТО.
- Разработать базу данных организации с использованием Borland Delphi.

Работа состоит из введения, двух глав, заключения, списка используемой литературы и приложений.

# 1 Теоретические основы базы данных

## 1.1 Определение и основные модели базы данных

Одна из основных областей применения компьютеров является хранение и переработка больших объемов информации в различных сферах деятельности человека: в экономике, банковском деле, торговле, транспорте, медицине, науке и т. д. [11].

Информационная система – это система, в которой реализуется автоматизированный сбор, обработка и манипулирование данными, включающая технические средства для обработки данных, программное обеспечение и обслуживающий персонал.

Целью любой информационной системы является обработка данных об объектах реального мира. Основа информационной системы – база данных. В широком смысле слова база данных – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. Предметная область – часть реального мира, которая подлежит изучению для организации управления его объектами и, в конечном итоге, автоматизации, например предприятие, вуз и т. д.

Для того чтобы создать базу данных, пользователю нужно упорядочить информацию по разным признакам и быстро произвести выборку с произвольным сочетанием признаков. При этом необходимо выбрать правильную модель данных. Модель данных – это формальное представление ключевых категорий восприятия реального мира, которые представлены его объектами, связями, свойствами, а также их взаимодействиями.

Система управления базами данных (СУБД) – система программ, которая позволяет создавать базу данных, обновлять хранимую в ней информацию, обеспечить удобный доступ к ней с целью просмотра и поиска.

Объединение базы данных и программы системы управления базами данных образует информационно-поисковую систему, которая называется банком данных.

В данном разделе раскрыты основные понятия информационной системы, базы данных, предметной области, модели данных и системы управления базами данных.

Группа связанных между собой элементов данных обычно называют записью. Существует три основных модели организации данных и связей между ними: иерархический (в виде дерева), сетевой и реляционный [3].

В иерархической базе данных (рисунок 1) присутствует упорядоченность элементов в записи, где один элемент считается главным, а остальные – подчиненными. Данные в записи построены по порядку в определенной последовательности, как лестничные ступеньки, а поиск данных осуществляется последовательно спусканием со ступеньки на ступеньку. Иерархическую базу данных формирует каталог файлов, который хранится на диске. Такой же базой данных является родовое генеалогическое дерево.

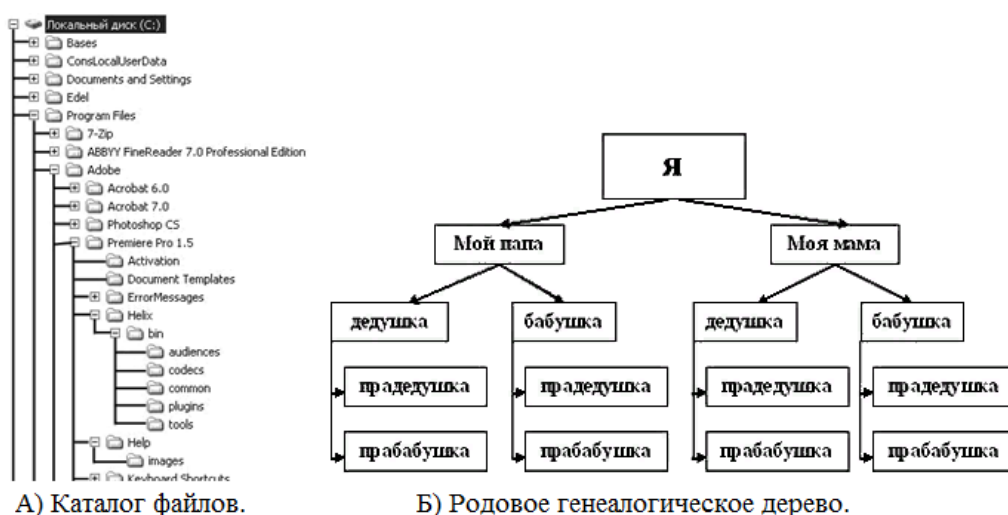


Рисунок 1 – Иерархическая модель базы данных

Сетевая база данных (рисунок 2) обладает преимущественной гибкостью, так как в ней, можно добавочно установить к вертикальным



иерархическим связям горизонтальные связи. Это упрощает процесс поиска необходимых элементов данных, поскольку уже не нуждается в обязательном прохождении всех предшествующих ступеней.

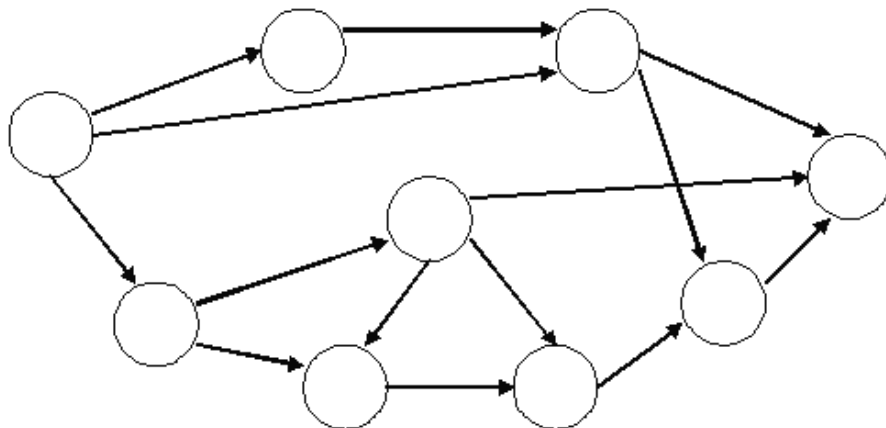


Рисунок 2 – Сетевая модель базы данных

Главное отличие реляционной модели (рисунок 3) от иерархической или сетевой – отсутствие физических отношений. Вся информация сохраняется в виде таблиц (отношений), которые состоят из рядов и столбцов. Данные двух таблиц соединены общими столбцами, а не физическими указателями или ссылками. Для манипулирования с рядами данных существуют специальные операторы.

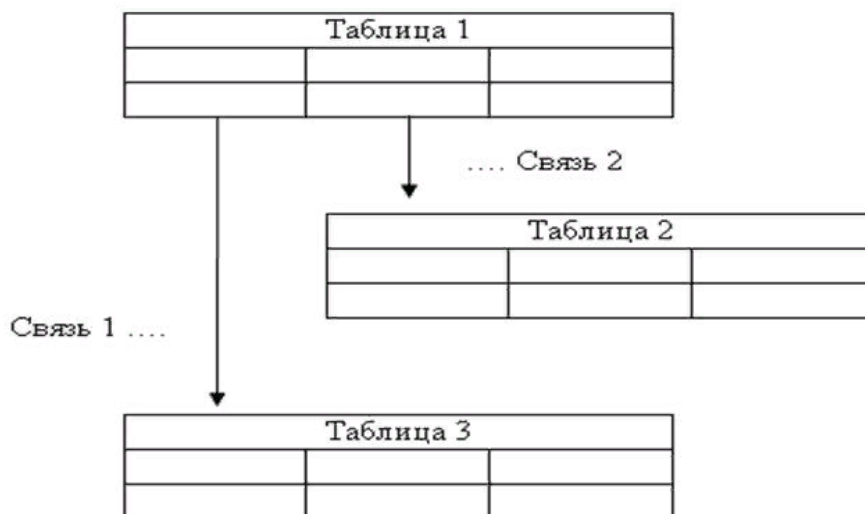


Рисунок 3 – Реляционная модель базы данных

Реляционная модель, в отличие от сетевой и иерархической, как выяснилось более удобна. Не обязательно просматривать все указатели, что упрощает выполнение запросов на выборку информации.

В реляционной модели, как объекты, так и их отношения представлены только таблицами, и ничем более.

В данном разделе раскрыты основные модели, понятия и классификации базы данных.

### **Требования к базе данных, этапы проектирования**

Процесс развития технологий баз данных формируется рядом факторов: увеличение роста информационной потребности пользователя, условиями эффективного подхода к информации, возникновением новых видов машинной памяти и повышением ее емкости, иными методами и средствами в сфере телекоммуникаций и др.

К современным базам данных предъявляются следующие основные требования:

1. Высокое быстродействие (малое время отклика на запрос). Время отклика - интервал времени от момента запроса к базе данных до фактического принятия данных. Похожий термин время доступа - интервал времени между выдачей команды записи (считывания) и фактическим получением данных. Под доступом рассматривается операция чтения, поиска данных или записи их. Зачастую операции записи, удаления и модификации данных называются операциями обновления.

2. Легкость обновления данных. Важнейшими являются эти первые два противоречивых требования: для повышения быстродействия требуется упрощенная структура базы данных, что, в свою очередь, усложняет процесс обновления данных, увеличивает их избыточность.

3. Независимость данных - возможность преобразования физической и логической структуры базы данных без изменения представлений пользователей. Независимость данных подразумевает инвариантность к характеру хранения данных, техническим средствам и программному

обеспечению. Она минимизирует преобразования структуры базы данных при изменениях стратегии доступа к данным и структуры самих исходных данных.

4. Совместное использование данных многими пользователями.

5. Безопасность данных - защита данных от умышленного или неумышленного нарушения секретности, разрушения или искажения. В безопасность данных включена их целостность и защита. Целостность данных - устойчивое хранение данных от разрушения и уничтожения, которое связано с неполадками технических средств, системными ошибками и ошибочными действиями пользователей.

#### Этапы проектирования баз данных

Для того чтобы создать базу данных требуется подробное ее описание, а именно необходим проект. Проект – это эскиз некоторого устройства, который будет впоследствии реализован в реальность.

Процесс проектирования очень тяжелый, следовательно, необходимо разделить его на логически завершенные части – этапы (рисунок 4).



Рисунок 4 – Основные этапы проектирования базы данных

Процесс проектирования базы данных представляет собой процесс переходов от неформального словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели [20].

Ниже рассмотрим основные этапы проектирования базы данных:

- Сбор сведений и системный анализ предметной области – это первый и важнейший этап при проектировании базы данных. В нем необходимо провести подробное словесное описание объектов предметной области и реальных связей, присутствующих между реальными объектами. Желательно чтобы в описании определялись взаимосвязи между объектами предметной области.

- Инфологическое (концептуальное) проектирование — построение семантической модели предметной области, а именно информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную систему управления базы данных и модель данных.

На сегодняшний день наиболее широкое распространение получила модель Чена «Сущность-связь», она стала реальным стандартом в инфологическом моделировании, и приобрела название ER – модель.

- Выбор систем управления базой данных реализуется на основании различных требований к базам данных и, в соответствии, возможностей систем управления базой данных, а также в зависимости от имеющегося опыта разработчиков.

- Даталогическое проектирование — описание базы данных в терминах принятой даталогической модели данных. Даталогическое или логическое проектирование в реляционных базах данных приводит к разработке схемы базы данных, а значит совокупности схем отношений, которые адекватно моделируют семантические связи между объектами и объекты предметной области. Функциональные зависимости между атрибутами базы данных являются основой анализа корректности схемы.

Физическое проектирование — увязка физической среды хранения и логической структуры базы данных для обеспечения наиболее эффективного размещения данных, т.е. отображение логической структуры базы данных в структуру хранения. Решается вопрос выбора эффективных методов доступа к различным компонентам «физической» базе данных, распределения хранимых данных в пространстве памяти, решаются вопросы обеспечения сохранности и безопасности данных.

Таким образом, были рассмотрены основные требования к базам данных, а так же основные этапы проектирования баз данных.

### **Внедрение базы данных в малый бизнес**

Малый бизнес – это бизнес, базирующийся на предпринимательскую деятельность небольших фирм, малых предприятий, которые формально не входят в объединения.

Роль малого бизнеса трудно переоценить: это та форма, в которой легче всего начинать самостоятельный бизнес, а также осуществлять его, если это позволяют критерии малого предприятия. Русскоязычный синоним бизнеса – предпринимательство [9].

Научный термин понятия «предприниматель» возникло в работах известного английского экономиста начала XVIII века Ричарда Кантильона, раскрывшего термин предпринимателя как человека, который за определенную цену приобретает средства производства, чтобы затем создавать определенную продукцию и продавать ее намеренно для получения дохода. Предприниматель берет все издержки по обязательствам на себя, часто не зная, по какой цене будет осуществляться реализация. Именно он создал одну из первых концепций предпринимательства, а под предпринимателем представлял человека, который подвергается риску, на основании того, что фермеры, торговцы, ремесленники и остальные собственники приобретают по определенной цене, а продают по неизвестной.

Другой французский экономист Тюрго считал, что предприниматель должен обладать не только определенной информацией, но и капиталом. Он

подчеркнул, основа всей экономики это капитал, а цель успеха предпринимателя это прибыль, основа роста производства.

Таким образом, предпринимательство — это особенный вид экономической активности (согласно которой воспринимается целесообразная деятельность, направленная на извлечение прибыли), которая основана на самостоятельной инициативе, ответственности и инновационной предпринимательской идее (рисунок 5).



Рисунок 5 – Сущность творческой, инициативной деятельности человека в рыночной экономике

Базы данных и малый бизнес понятия неразделимые. Базы данных могут содержать любую возможную информацию, и доступ к ней возможен на расстоянии в один клик. Коммерческие базы данных служат поддержкой собственникам малого бизнеса для систематизации информации о своих клиентах, сотрудниках и партнерах.

Базы данных являются объединенной составляющей электронного бизнеса. Они помогают предпринимателям анализировать и хранить самую

существенную информацию о товарах, услугах, продажах затем, чтобы правильно среагировать на быстро меняющиеся условия рынка.

#### Информация о продукции:

1. Электронную базу данных используют для хранения информации о продукции, которая была произведена или продана. В ней содержится, подробное описание продукта, его технические характеристики, справочный номер, цена, доступность, рекламные предложения, информация о поставщиках.

2. Складская база данных может подсказать, сколько единиц продукции находится в данный момент на складах, в подсобках или на магазинных полках. Основанная на штрих-кодах система товарного учета позволит отслеживать все перемещения товара и соответственно обновлять информацию в базе данных, поэтому вам никогда не придется вручную пересчитывать запасы хранимой на складах продукции.

Точно так же база данных может уведомлять вас об исчерпании товаров определенной категории, так что вы сможете заказать дополнительную партию до того, как товар данной номенклатуры полностью закончится, а вовремя обновляя информацию в базе данных, можете быть уверены, что посетители вашего сайта всегда будут обеспечены исключительно свежей и актуальной информацией.

#### Информация о клиентах:

В базе данных обязательно должна присутствовать информация о клиентах, включая полные имена и достоверные контактные данные. Записи для бизнес-клиентов могут содержать информацию о типе и величине их бизнеса, а также о сфере деятельности. Что касается потребителей, то в этом случае ценна любая информация относительно покупательной способности и предпочтениях вашей целевой аудитории.

#### Анализ транзакций:

Для успешного ведения электронного бизнеса необходимо обладать о своих клиентах определенным объемом существенной информации. Тщательно

отслеживая и фиксируя запросы пользователя, историю его покупок, просмотры тех или иных товаров, вы можете составить детальную картину личных предпочтений и покупательского потенциала своих клиентов.

#### Персонализация:

Составление детальных личностных профилей позволит вам персонализировать предложения, основываясь на покупательских предпочтениях ваших клиентов. Если ваш сайт продает, например, спортивную одежду, вы можете провести рекламную кампанию, рассчитанную на молодых женщин, информация о которых содержится в вашей базе данных, предпочитающих спортивные костюмы определенного бренда. Для назначения цены также можно использовать информацию из базы данных – например, для определенной категории клиентов база данных может возвращать цену со скидкой на некоторые виды товаров.

#### Управление маркетинговой кампанией:

Вы можете оценить эффективность проведенных вами маркетинговых компаний и товарных предложений, проведя анализ транзакций и сгенерированных базой данных отчетов. Сравните продажи до и после и определите наиболее успешную кампанию. Проанализировав тенденции продаж той или иной продукции, вы сможете внести нужные корректировки в цены, ассортимент и количество товарных запасов.

#### Самообслуживание:

Если уровень защищенности вашей базы данных достаточно высок, вы можете предоставить своим клиентам, поставщикам и партнерам доступ к определенным частям базы данных. Клиенты, например, могут просматривать историю своих заказов, а также статусы доставки невыполненных заказов и заказов, находящихся на стадии выполнения. Точно так же вы можете предоставить поставщикам доступ к товарному складу, чтобы они могли более эффективно планировать свои производственные графики, а специалистам по логистике – к информации о текущих заказах для составления графика доставки.



В этом параграфе работы изучили основные понятия малого бизнеса, предпринимательства. Рассмотрели объединенные составляющие электронного бизнеса. В процессе изучения был сделан следующий вывод, что базы данных и малый бизнес понятия неразделимые.

### **Технология обработки данных, способ доступа к данным**

По технологии обработки данных базы делятся на: централизованные и распределенные. Централизованная база данных хранится в памяти одной вычислительной системы. При условии, что вычислительная система будет являться компонентом сети ЭВМ, имеется возможность распределенного доступа к такой базе. Такой способ применения баз данных зачастую используют в локальных сетях ПК. Распределенная база – база данных, состоящая из нескольких, пересекающихся или дублирующих друг друга частей, которые хранятся в вычислительной сети. Работа в такой базе выполняется при помощи системы управления распределенной базой данных (СУРБД).

По способу доступа к данным базы данных подразделяются на локальный доступ и удаленный (сетевой доступ) баз данных. Системы централизованных баз данных с сетевым доступом располагают различными архитектурами таких систем как: файл-сервер; клиент-сервер.

#### Файл-сервер:

В файл-серверных (рисунок 6) системах управления базами данных файлы данных находятся централизованно на файл-сервере. Система управления базами данных располагается на любом клиентском компьютере (рабочей станции). Через локальную сеть осуществляется доступ систем управления базами данных к данным. Синхронизация чтений и обновлений выполняется путем файловых блокировок.

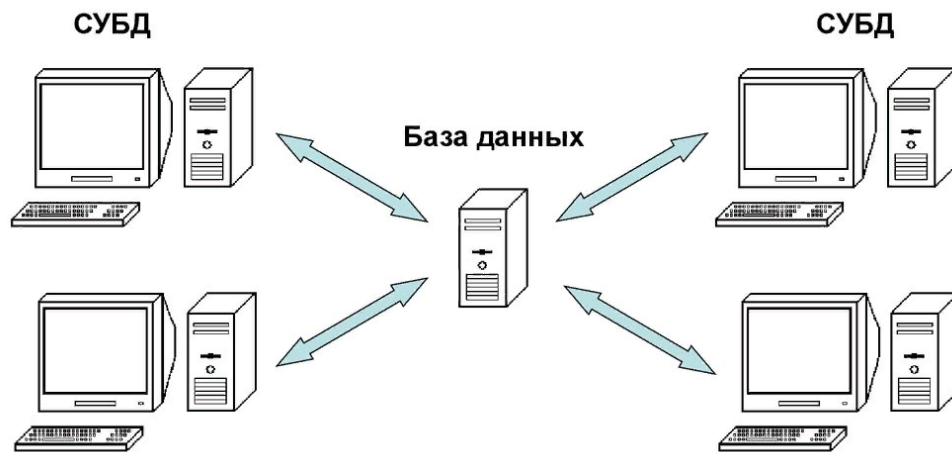


Рисунок 6 – Файл-серверная архитектура СУБД

Достоинство: данная архитектура представляет низкую нагрузку на процессор файлового сервера.

Недостатки: повышенная нагрузка на локальную сеть; усложненность или невозможность централизованного управления; трудность в обеспечении очень важных характеристик, как высокая доступность, высокая безопасность и высокая надёжность.

Применение: очень часто в локальных приложениях, в которых используется функция управления базы данных; в системах с очень низкой обработкой данных и низкими пиковыми нагрузками на базу данных.

В настоящее время файл-серверную технологию считают устаревшей, а дальнейшая её эксплуатация в крупных информационных системах — недостатком.

Клиент-сервер:

Клиент-серверная (рисунок 7) система управления базой данных находится на сервере вместе с базой данных, а доступ к базе данных осуществляется непосредственно, в монопольном режиме. Все запросы клиентов на обработку данных перерабатываются клиент-серверной системой управления базой данных централизованно.

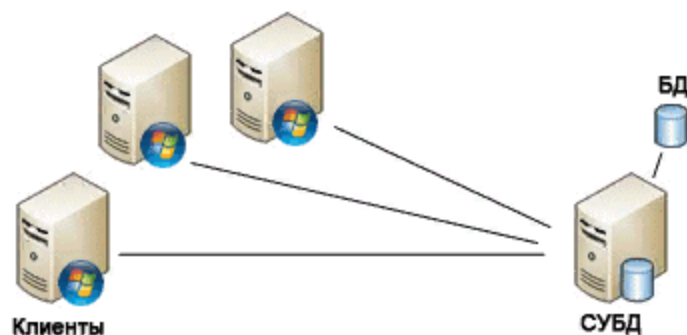


Рисунок 7 – Клиент-серверная архитектура СУБД

Недостаток: повышенные требования к серверу.

Достоинства: потенциально более низкая нагрузка локальной сети; удобство централизованного управления; преимущество обеспечения очень важных характеристик, как высокая доступность, высокая надёжность и высокая безопасность.

В данном разделе была изучена технология обработки данных, способ доступа к данным, рассмотрены достоинства и недостатки архитектур таких систем как: файл-сервер; клиент-сервер.

## 1.2 Информационная безопасность, виды угроз и защита информации

Информационная безопасность – защищенность информации и поддерживающей инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера, которые могут нанести неприемлемый ущерб субъектам информационных отношений, в том числе владельцам и пользователям информации и поддерживающей инфраструктуры [37].

Основные типы угроз информационной безопасности:

1. Угрозы конфиденциальности – несанкционированный доступ к данным (например: контактные данные, перехваченные посторонними лицами, о состоянии счетов клиентов банка).

2. Угрозы целостности – несанкционированная модификация, дополнение или уничтожение данных (например: хищение денежных средств, с помощью внесения изменений бухгалтерских записей).

3. Угрозы доступности – ограничение или блокирование доступа к данным (например: нет возможности подключения к серверу с базой данных в следствии DDoS-атаки).

Источникам угроз информационной безопасности (рисунок 9) нужно уделять особое внимание.

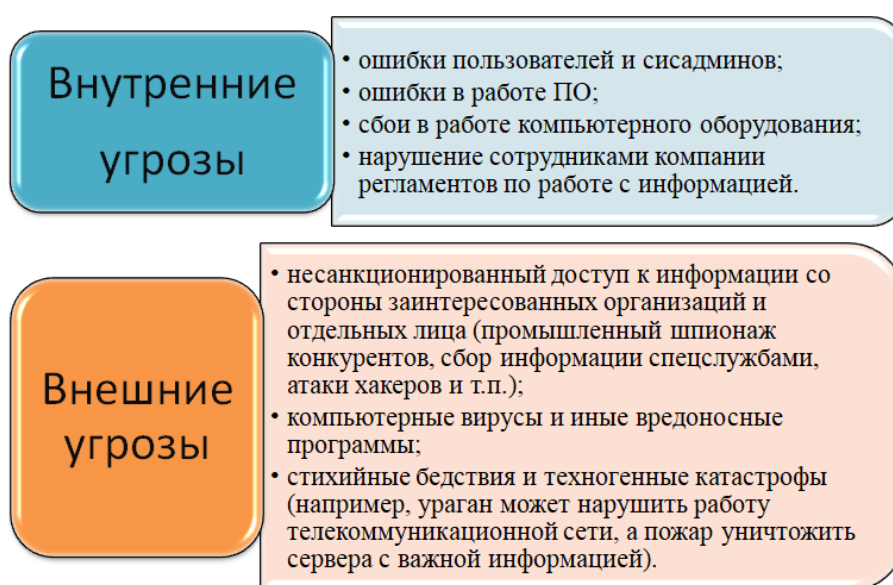


Рисунок 9– Источники угроз информационной безопасности

Цель защиты информации – снизить потери, вызванные в результате нарушения целостности и конфиденциальности данных, а также их недоступности для потребителей.

Средства защиты информации:

- Технические (аппаратные) средства – сигнализация, решетки на окнах, электронные ключи, генераторы помех, которые препятствуют передаче данных по радиоканалам и т.д.

- Программные средства – антивирусы, программы-шифровальщики данных, системы для аутентификации пользователя и т.п.

- Смешанные средства – комбинирование программных и аппаратных средств.

- Организационные средства – регламенты, правила работы, законодательные акты в области защиты информации, обустройство помещений с компьютерной техникой и проводка сетевых кабелей с учетом требований по ограничению доступа к информации и пр.

Таким образом, были рассмотрены основные типы и источники угроз информационной безопасности и средства защиты информации.

### **1.3 Организационная модель СТО**

Организационная модель создается в начале моделирования бизнес-процессов. В иерархической модели отображаются подразделения предприятия, наименование должностей, фамилии руководителей подразделений и отдельных сотрудников.

Сложные иерархические структуры могут быть декомпозированы, например, структура подразделения может быть раскрыта на более детальной схеме. Также могут быть построены модели иерархии подчиненности во временных коллективах, группах и т.д. Все объекты организационных моделей могут быть использованы при построении моделей бизнес-процессов.

Организационная модель имеет иерархическое строение: на верхнем уровне находятся самостоятельные подразделения, структура которых детализируется на более низких уровнях.

На низшем уровне приводится описание должностей — штатных единиц, занимаемых конкретными сотрудниками. При наличии в подразделении нескольких одинаковых должностей производится их нумерация.

На рисунке 10 приведен пример иерархической организационной модели.

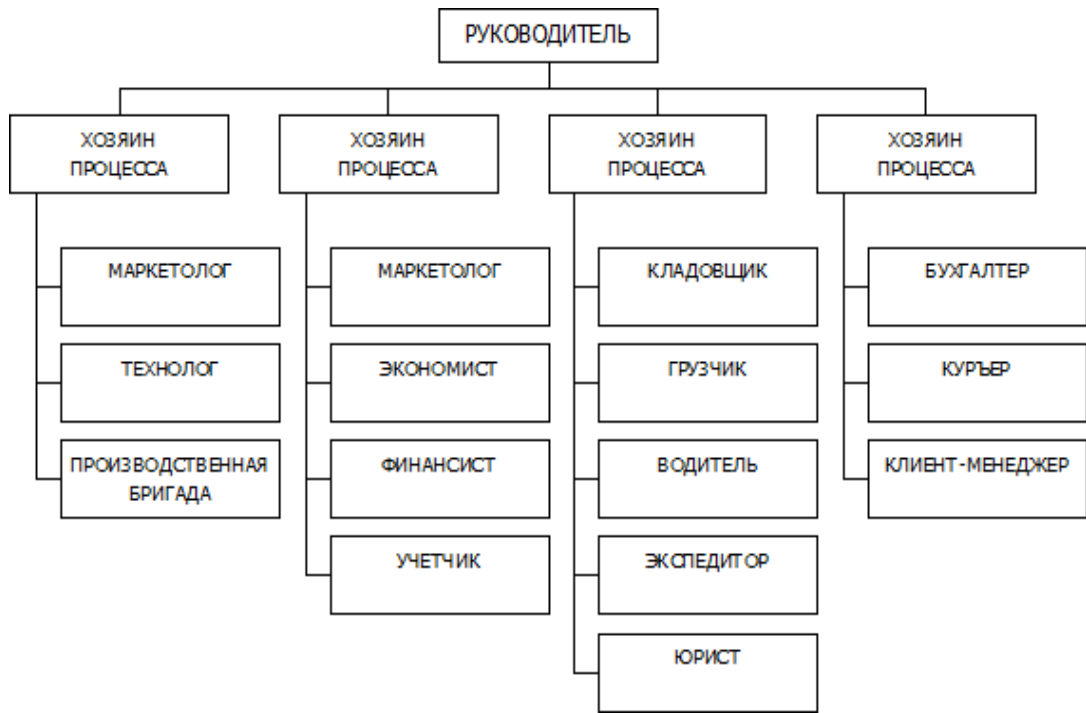


Рисунок 10 – Иерархическая организационная модель

Организационные единицы направлены для обозначения различных компонентов на предприятия, им может быть присвоен тип, определяющий их принадлежность к отделу или группе.

Виды связей, заложенные в методологии, позволяют отражать различные виды отношений между объектами организационной структуры.

Организационная модель (рисунок 11) предприятия станции технического обслуживания выглядит следующим образом:

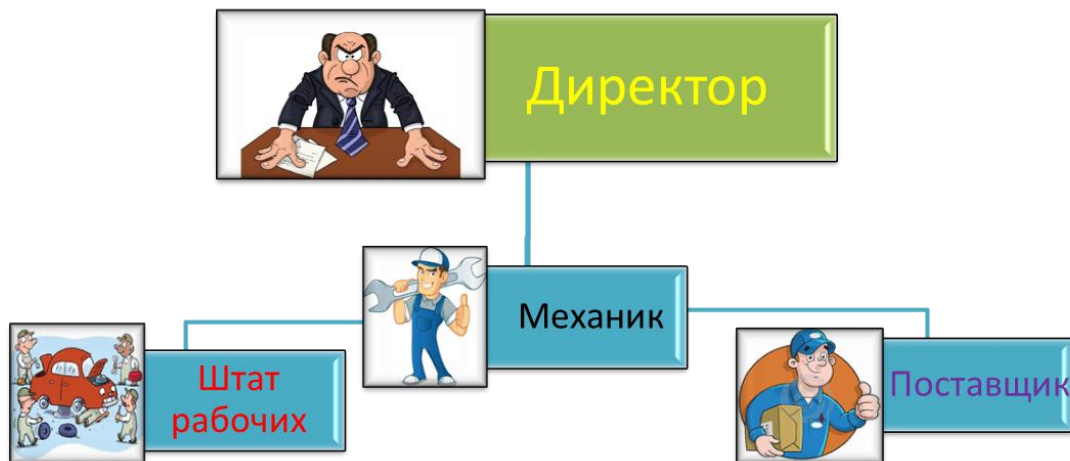


Рисунок 11 – Организационная модель СТО

В управлении предприятия стоит директор, в его подчинении находятся: механик (по совместительству заведующий складом запчастей) он выполняет функцию контроля за исполнением работы по ремонту и обслуживанию транспортного средства, а так же отвечает за контроль поставки запчастей. Далее следом идет штат рабочих: автослесарь, автоэлектрик (отвечающие за услуги выполнения работ по обслуживанию транспортных средств), а также поставщик (поставляет расходные материалы автозапчастей).

На основе созданной модели, ярко выделены связи отчетностей. Для разработки информационной системы требуется создание одного прикладного приложения, формирующего отчет деятельности работы предприятия в виде базы данных.

#### **1.4 Процессная модель**

Процессная модель представляет собой описание предприятия, как целостная система со всеми неразрывно связанными в ней бизнес-процессами, структурными подразделениями, производственными и управленческими функциями.

Процессные системы основываются из нескольких базовых принципов:

- принцип объединения процедур: операции, которые выполняются различными сотрудниками, интегрируются в одну, а значит, выполняется горизонтальное сжатие процесса. Когда не получается привести все шаги процесса к одной работе, создается команда, которая отвечает за данный процесс;

- принцип неразрывной последовательности: шаги процесса соблюдаются в естественном порядке, работа совершается в том месте, где это целесообразно, смешанными группами, которые состоят из работников различной специализации или предметной (функциональной) принадлежности;

- принцип владельца процесса: управляющий менеджер играет роль буфера между заказчиком и сложным процессом, он обеспечивает единую точку контакта, и ведет себя с заказчиком так, как если бы был ответственным за весь процесс;
- принцип самостоятельности выбора: исполнители несут ответственность за получение заданного результата деятельности и самостоятельно принимают решения;
- принцип горизонтального контроля: потребитель проверяет качество результата - последующим элементом процессной цепочки;
- принцип системности (целостности) управления: управление затратами осуществляется по месту их возникновения, система управления издержками создается вместе с организационной структурой, без отрыва от деятельности, «один процесс - одно подразделение - один бюджет».

Предлагаемая методика применяется на достаточно успешных предприятиях, достаточно внимательна к существующему опыту, старается его сохранить и систематизировать. Существует и более радикальный «нулевой» вариант методики, когда система процессов фактически строится заново. Он может быть применен для тех предприятий, которые нуждаются в серьезной перестройке бизнесов, очистке от старой неэффективной деятельности и ее носителей.

На рисунке 12 представлена процессная модель организации.



Рисунок 12 – Процессная модель



В каждой организации ежедневно протекает множество процессов. Чтобы систематизировать и упорядочить бизнес-процессы, протекающие на предприятии, их принято делить на следующие основные группы:

- Основные процессы: генерируют доходы компании. Примерами процессов закупки, продажи и производства продукта; ее складирования и транспортировки.

- Процессы обеспечения: поддерживают инфраструктуру компании. В эту категорию входит техническое, информационное и административно – хозяйственное обеспечение; обеспечение безопасности, социальной сферы и финансовых операций; консультационное и аудиторское обеспечение.

- Процессы управления: управляют компанией. К этой категории относят такие процессы как процессы управления финансами, маркетингом, персоналом и логистикой; экономическое, стратегическое и организационное управление, а также управленческий учет.

- Процессы развития: развивают компанию. Представляют инвестиционные виды деятельности, нацеленные на прибыль от сегодняшних вложений в долгосрочной перспективе.

На основании анализа осуществляемых процессов, в организации формируется процессная модель (рисунок 13):



Рисунок 13 – Процессная модель станции технического обслуживания

Описание процесса:

- Приём заявки заказчика на обслуживания транспортного средства;
- Регистрация транспортного средства в базе данных;
- Диагностики выявления неисправностей транспортного средства;
- Ремонт выявленных неисправностей транспортного средства;
- Формирование отчёта по выполненной работе для последующей сдачи транспортного средства заказчику.

Для того чтобы результат на выходе максимально соответствовал ожиданиям клиента, предприятие нуждается в слаженной организованной информационной системе, которая бы помогала персоналу в осуществлении процессов, и могла хранить большие объемы данных.

В данной главе рассмотрен процесс теоретической основы разработки базы данных по учету работы СТО. Данное изучение позволяет сделать ряд выводов:

1. Внедрение информационных систем позволяет улучшить эффективность работы на предприятии.
2. Определили соответствующий требованиям язык программирования объектно-ориентированное программирование Borland Delphi 7.0.
3. Выявили наиболее эффективные средства защиты.
4. Сопоставив теоретические основы информационных систем на предприятии, создали организационную и процессную модели для лучшего представления организации процессов на предприятии.

## 2 Разработка программного продукта

### 2.1 Техническое задание на разработку

Наименование программного продукта: «Разработка базы данных по учету работы СТО (на примере «Автотехцентр»)».

Область применения:

- Программный продукт разработан для персонала организации: механика (он же заведующий складом), штаба рабочих (автослесарь, автоэлектрик, поставщик).

- Разработчик данного продукта студент ЛПИ – филиал СФУ, Мальцев А.В.

- Заказчик программного продукта – ИП «Микрон» Красноярский край, г. Лесосибирск, улица Енисейская, 23А. Генеральный директор Котин Владимир Александрович.

Факторы эффективности и качества программы:

- 1) Социальный фактор. Программная разработка проста в обучении и предназначена не только для опытных, но и на начинающих пользователей, работающих под Windows. Просто интуитивно понятный интерфейс даёт возможность работать с программным продуктом без специальной и предварительной подготовки;

- 2) Аналоговое или превосходящее состояние на рынке программного оборудования данного профиля. Технология разработки прикладных программ в визуальных средах программирования ставит ее интерфейс универсальным и совместимым с ОС Windows;

- 3) Экономические факторы. Программный продукт соответствует наилучшему соотношению цены и предлагаемых ею характеристик и займет свою нишу на рынке бюджетных программ. Основными заказчиками станут системные инженеры или пользователи, у которых нет возможности закупить дорогой программный продукт.

Цели разработки программы:

- Разработка программного продукта преследует ряд экономическо-технических целей;
- Разработка программного продукта, необходимого для упрощения работы персонала предприятия;
- Разработка интуитивно понятного интерфейса программы с удобным и универсальным для экономии времени;
- Разработка дешевой альтернативы существующим в настоящее время дорогим программам данного типа.

Требование к функционалу:

1. Отображать подробную информацию о выполняемой работе;
2. Сохранять таблицу;
3. Предоставлять отчёт.

Организация входных и выходных данных является отображение на монитор полной информации в виде таблиц.

Требования к надежному функционированию:

- Бесперебойная работа программ при нормальной работе ЭВМ. Если во время функционирования программы произошел сбой в работе ЭВМ, восстановление нормальной работы программы должно производиться после:

1. Запуска исполняемого файла программы;
2. Перезагрузки операционной системы;
3. Повторного выполнения действий, потерянных до последнего сохранения информации в файл на магнитном диске.

Требование надежности программы должно соответствовать технологии программирования, предусматривающей:

- проверку исходных текстов программы;
- автоматическое тестирование модулей программы;
- тестирование сопряжений модулей программы;
- комплексное тестирование программы.

Требование к эксплуатации и хранению:

- Программный продукт должен храниться в виде копий на лазерном компакт диске и обязательно эталонная версия программы.

Минимальные требования к техническим устройствам:

- 1) Pentium 3 – 800 МГц,
- 2) 64Мб RAM,
- 3) клавиатура,
- 4) мышь,
- 5) монитор,
- 6) принтер.

Для решения задачи использовать частично автоматизированный метод.

В ходе разработки программного продукта использовать следующие программные средства:

1. Microsoft Office Access;
2. Borland Delphi v7.0 и выше.

Виды испытаний:

Проверка документации к разработанному продукту проводится руководителем станции технического обслуживания ИП «Микрон».

В процессе разработки программы испытания и тестирование должны проводиться самим разработчиком:

- 1) С использованием контрольных тестов, которые позволяют добиться проверки правильности работоспособности и взаимной совместимости максимального числа функций и операторов программы или модуля при минимальных затратах временных и финансовых ресурсов.

- 2) Путем пошагового исполнения программы или модуля в соответствии с набором тестовых примеров и сравнения полученных в процессе тестирования значений с контрольными значениями тестовых примеров.

Общие требования к приёмке:

Приёмка программы и ввод в тестовый режим будет, осуществляется руководителем организации ИП «Микрон».

Таким образом, определены требования к предстоящей разработке программного продукта его характеристики и функционал. В соответствии предъявляемых требований разрабатывается база данных по учету работы станции технического обслуживания.

## **2.2 Обоснования выбора Microsoft Office Access**

Программный комплекс MS Office является самым распространенным пакетом автоматизации работы в офисе. Поэтому СУБД (Система управления базами данных) Access, входящая в комплект профессиональной версии комплекса стала де-факто стандартной базой данных, используемой в современном бизнесе. СУБД крайне полезна в бизнесе для организации документооборота, так как позволяет исключить дублирование и ускорить обработку документов клиентов, и, следовательно, увеличить оборот и доход фирмы. Улучшение вида документов при этом положительно влияет на имидж фирмы и тоже позволяет привлекать клиентов. Основные функции СУБД — это определение данных (описание структуры баз данных), обработка данных и управление данными.

База данных в MS Access представляет собой совокупность инструментов для ввода, хранения, просмотра, выборки и управления информацией. К этим средствам относятся таблицы, формы, отчеты, запросы. В MS Access поддерживаются два способа создания базы данных. Можно создать пустую базу данных, а затем добавить в нее таблицы, формы, отчеты и другие объекты. Такой способ является наиболее гибким, но требует отдельного определения каждого элемента базы данных. Кроме этого имеется возможность создать с помощью мастера базу данных определенного типа со всеми необходимыми таблицами, формами и отчетами. Так как MS Access содержит большой выбор подготовленных баз

данных, второй способ во многих случаях может оказаться предпочтительным. В обоих случаях останется возможность в любое время изменить и расширить созданную базу данных.

Система управления базой данных Microsoft Access состоит из системы управления реляционной базой данных, которая включает в себя все обязательные инструментальные средства для формирования локальной базы данных, общей базы данных в сети с файловым сервером или для создания приложения пользователя, занимающегося обработкой с базой данных на SQL-сервере.

Локальную одноранговую сеть или сеть с файловым сервером используют для создания многопользовательской базы данных Access, а также получения одновременного доступа нескольких пользователей к общей базе данных.

Один из основных плюсов разработки на MS Access – плотная связь с Microsoft Office. Для нормальной работы созданного приложения достаточно установить Офис. При этом так же устанавливаются все нужные для работы Access библиотеки, ODBC и т.д.

Каждая задача при решении включает в себя выбор способа решения. Access при всей его внешней простоте позволяет решать весьма сложные задачи. Access достаточно серьезный инструмент, на котором быстро и хорошо можно сделать довольно сложные вещи. Для многих проектов, где важна скорость разработки, Access подходит как нельзя лучше.

Каждый язык программирования предназначен под определенные цели и задачи. И Access решает достаточно большое количество офисных, бухгалтерских и других подобных задач. Программирование в Access, создание структуры базы данных аналогичны как и в других средах.

Освоив основные принципы, легко перейти к другим средствам разработки. Отличие будет только в обертке - названиях функций, процедур, синтаксических правилах и т.д. Но основное - принцип создания баз и алгоритмов останется.

Access изначально представлялся как настольная база данных, и в этой сфере ему трудно найти равного соперника. Так же Access очень хороший контейнер для хранения запросов и таблиц при работе с ними через ADO. Если не устраивает типовой набор контролов, то можно написать интерфейс для работы при помощи других средств, например, Visual Studio или Delphi.

В Microsoft Access включены мастера, помогающие производить анализ структуры данных, импортировать электронные таблицы и текстовые данные, повышать быстродействие приложения, создавать и настраивать одно из более, чем двадцати типов приложений с использованием встроенных шаблонов. Чтобы полностью автоматизировать работу приложения, можно использовать макросы для связывания данных с формами и отчетами. Большинство приложений можно создать, не написав ни единой строки программного кода. Однако при необходимости построения действительно сложного приложения можно использовать язык программирования – Visual Basic для приложений.

Access позволяет разработать профессиональную базу данных в более короткие сроки и за меньшие деньги, и вполне удовлетворить запросы заказчика. Кроме того, такое решение обладает большей гибкостью, настраиваемостью, в него легко можно внести дополнения и изменения, быстрее и дешевле чем в промышленные базы данных на основе Oracle и C++. Что касается трат на покупку MS Access, то он входит в MS Office, а этот продукт уже имеется на большинстве предприятий.

Преимуществом Microsoft Access является то, что она состоит из простого графического интерфейса, который позволяет создать собственную базу данных, а так же разрабатывать приложения, с использованием встроенных средств. По сравнению с другими настольными системами управления базой данных, Microsoft Access может хранить все данные в одном файле, однако распределение происходит по разным таблицам, как и положено реляционной системе управления базой данных.



Недостатком Microsoft Access является то, что отсутствуют надежные стандартные средства по отношению защиты информации и разграничения доступа. Под стандартными способами защиты понимается защита с применением пароля базы данных и защита с использованием пароля пользователя. Снять такую защиту для специалиста не составит большого труда. Тем не менее, такой защиты от просмотра или неосторожных действий вполне хватит для сотрудников небольшой компании. Но отсюда можно сделать вывод, что не стоит доверять MS Access секретные или очень ценные данные.

Преимущества Microsoft Access по сравнению с другими системами подобного класса:

1. Развитая система справки и понятный интерфейс;
2. Данные хранятся в виде таблиц;
3. Легкость в связывании данных, которые хранятся в разных таблицах;
4. Возможность работы с другими базами данных;
5. Богатый набор визуальных средств разработки;
6. Улучшенная система восстановления поврежденных файлов;
7. Поддерживает множество языков;
8. Предоставляется возможность интеграции с Visual Basic Application или Delphi;
9. Встроенный SQL, позволяющий максимально гибко работать с данными и значительно ускоряющий доступ к внешним данным;
10. Позволяет устанавливать связь с другими программами из пакета Microsoft Office;
11. Простой графический интерфейс, позволяющий не только создавать собственную базу данных, но и так же разрабатывать простые и сложные приложения.

Microsoft Access обладает всеми чертами классической системы управления базами данных (СУБД). Access - это не только мощная, гибкая и

простая в использовании СУБД, но и система для разработки приложений баз данных.

Для проектирования базы данных необходимо располагать описанием выбранной предметной области, которое должно охватывать реальные объекты и процессы, определять все необходимые источники информации для обеспечения предполагаемых запросов пользователя и решаемых в приложении задач. Следует заметить, что чаще всего базы данных создаются средствами СУБД в области организационно-экономического управления.

Исходя из анализа, выбранная система управления базами данных, удовлетворяет поставленным требованиям, поэтому выбор был остановлен на данной системе программирования.

### **2.3 Обоснования выбора Borland Delphi 7.0**

При выполнении поставленной задачи, необходимо использовать для представления информационных материалов язык программирования, поэтому была выбрана система программирования Delphi версии 7 фирмы Enterprise (Borland), поскольку она обладает наиболее широкими возможностями для программирования приложений операционной системы Windows.

Delphi – это продукт Borland International для быстрого создания приложений. Высокопроизводительный инструмент визуального создания приложений, предоставляющий средства визуального программирования, которые в некоторой степени похожи как у Microsoft Visual Basic или в других инструментах для визуального проектирования. Основу Delphi составляет язык Object Pascal, являющийся расширением объектно-ориентированного языка Pascal.

В основном, Delphi рассчитан для профессиональных разработчиков, которые хотят очень быстро создавать приложения в архитектуре клиент-сервер. В Delphi создаются незначительные по размерам (до 15-30 Кбайт)

высокоэффективные исполняемые модули (.exe и .dll), благодаря этому Delphi привлекает тех, кто разрабатывает продукты для продажи.

Требования для клиентских рабочих мест значительно снижаются, из-за небольших размеров и быстро исполняемых модулей, это имеет весомое значение для конечных пользователей.

Среда визуального программирования Delphi7.0 которая работает в среде Windows XP, 7,8,8.1,10 и предоставляет программисту возможность реализации всех достоинств графического интерфейса в этой системе. Так как большинство пользователей персональных компьютеров работают в среде операционных систем семейства Windows, то такой интерфейс является для них наиболее привычным и удобным.

Многие системы разработки приложений для операционной системы генерируются код-успеваемости, который не может быть выполнен процессом без дополнительной трансляции во время работы самой программы, что снижает производительность компьютера. Delphi использует настоящий компоновщик и компилятор, и генерирует стопроцентный машинный код. Такая реализация лишена непроизводительных затрат, что делает программы, написанные на языке Delphi, максимально эффективными.

Так как Delphi 7.0 является средой программирования для Windows, то как и сама операционная система Delphi поддерживает длинные имена папок и файлов.

Для того что бы запустить программы, которые написаны на языке Delphi не требуется никаких дополнительных библиотек, интерпретаторов кода и прочих. Достаточно взять один-единственный сгенерированный исполняемый файл и запустить его там, где нужно. Для установки программы на другой компьютер не требуется создания каких-либо дистрибутивов, не нужен процесс инсталляции, достаточно переписать исполняемы файл программы.

Среда визуального программирования Delphi 7.0 является мощным средством для качественной и быстрой разработки программ для

операционной системы (ОС) Windows XP. Имеющаяся библиотека визуальных компонентов позволяет создать интерфейс пользователем за считанные минуты. Объектно-ориентированный язык Object Pascal, положенный в основу Delphi, является расширением языков Borland Pascal и Turbo Pascal фирмы Borland и нашел в себе отражение новых веяний в программировании. Компонентный принцип, который используется в Delphi, позволяет создавать полноценные Windows-приложения, написав минимальное количество строк кода. Delphi представляет собой открытую систему, позволяя добавлять в систему свои компоненты, модифицировать уже имеющиеся стандартные компоненты благодаря тому, что представлены их исходные коды. Благодаря всему этому разработка программ в среде Delphi становится приятной и легкой.

Преимущества Delphi (рисунок 15) по сравнению с аналогичными программными продуктами.



Рисунок 15 – Преимущества Delphi по сравнению с аналогичными программными продуктами

Хотя Delphi не имеет своего формата таблиц базы данных, но она тем не менее обеспечивает мощную поддержку различным СУБД – как промышленных, так и локальных. Средства Delphi для работы с БД можно разделить на два вида:

- 1) компоненты;
- 2) инструментальные средства.

Компоненты предназначены для создания приложений, осуществляющих операции с базой данных. К инструментальным средствам относятся специальные пакеты и программы, обеспечивающие обслуживание базы данных вне разрабатываемых приложений.

Для операций с базой данных система Delphi предлагает следующий набор инструментальных средств:

- 1) Data base Desktop – программа редактирования и создания таблиц, SQL-запросов и запросов QBE.

- 2) Borland Data base Engine (BDE) – процессор баз данных, который представляет собой набор драйверов и динамических библиотек, предназначенных для организации доступа к БД из Delphi-приложений BDE является центральным звеном при организации доступа к данным.

- 3) SQL Builder – программа визуального конструирования SQL-запросов.

- 4) BDE Administrator – утилита для настройки различных параметров BDE.

- 5) SQL Monitor – программа отслеживания порядка выполнения SQL-запросов к удаленным базам данных.

- 6) SQL Explorer – проводник БД, позволяющий редактировать и просматривать БД и словари данных.

- 7) IB Console – программа для управления удаленными БД.

- 8) SQL Links – драйверы для доступа к удаленным промышленному СУБД, таким как Microsoft SQL Server и Oracle. К промышленному серверу

Inter Base, который поставляется совместно с Delphi и является для него родным, доступ также можно организовать напрямую через BDE, не используя драйверы SQL Links.

9) Inter Base Server Manage – программа для запуска сервера Inter Base.

10) Data Pump – программа для переноса данных между БД.

11) db Express – набор драйверов для доступа к базам данных SQL с помощью таких компонентов, как SQL Connection, SQL Data Set, SQL L Query, SQL Stored Proc SQL Table. Db Express включает в свой состав следующие драйверы:

- DB2 – DBEXPDB2.DLL;
- MySQL – DBEXPMYS.DLL;
- InterBase – DBEXPINT.DLL;
- Oracle – DBEXPORA.DLL.

12) Inter Base Server – серверная и клиентская часть SQL.

Компоненты, предназначенные для работы с базой данных, находятся на страницах Data Control, Data Access, DBE. Db Express, ADO, Inter Base и Decision Cube палитры компонентов. Некоторые компоненты предназначены специально для работы с удалёнными БД в архитектуре «клиент-сервер».

Среда программирования Delphi предусмотрена на программирование разного рода приложений и обеспечивает большой численностью компонентов для этого. Прежде всего, работодателей интересует качество и скорость создания программ, а для этих характеристик подходит только среда визуального проектирования, которая способна взять на себя значительные объёмы рутинной работы для подготовки приложений, кроме этого согласовывать деятельность группы кодировщиков, технических писателей, тестеров и постановщиков.

Таким образом, выбранная платформа, удовлетворяет поставленным требованиям, поэтому выбор был остановлен на данной системе программирования.

## 2.4 Интерфейс конечного пользователя

Приступая к разработке программы, предусмотрен доступ с запрашиванием пароля (рисунок 16) для каждого создаваемого приложения.

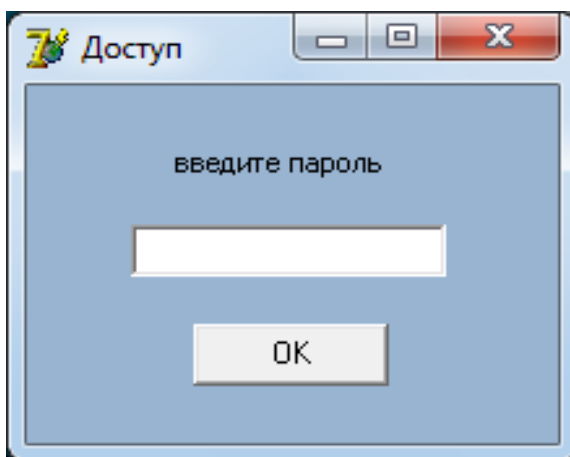


Рисунок 16– Доступ к приложениям

Если пароль введен верно, далее следует переход непосредственно к приложению, в противном случае на экране будет выведено (рисунок 17) уведомляющее сообщение о неправильности пароля.

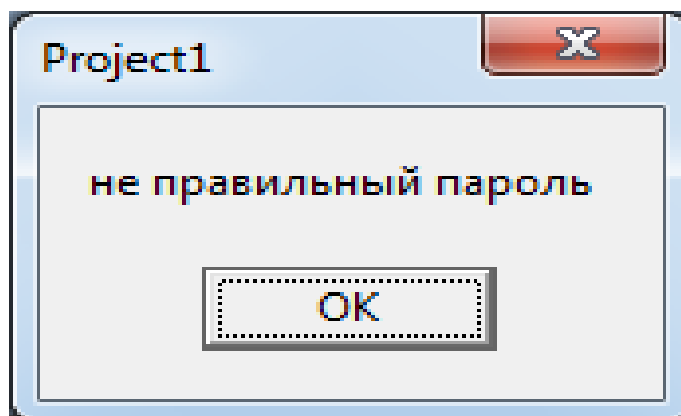


Рисунок 17– Сообщение о не верности пароля

Приложение включает в себя получение некоторых общих сведений, отчетов по работе от нижестоящих сотрудников, просмотреть которые

возможно при переходе по соответствующей кнопке. Главная форма приложения представлена на рисунке 18.

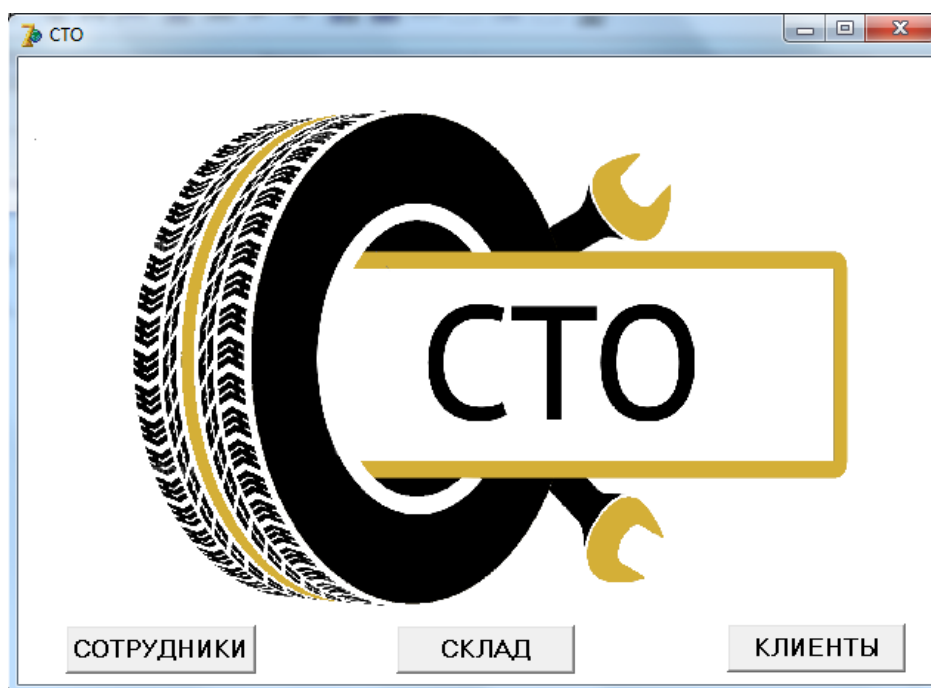


Рисунок 18 – Интерфейс главного окна приложения

Программа «сотрудники» (рисунок 19) включает в себя базу данных о сотрудниках с краткой информацией о них.

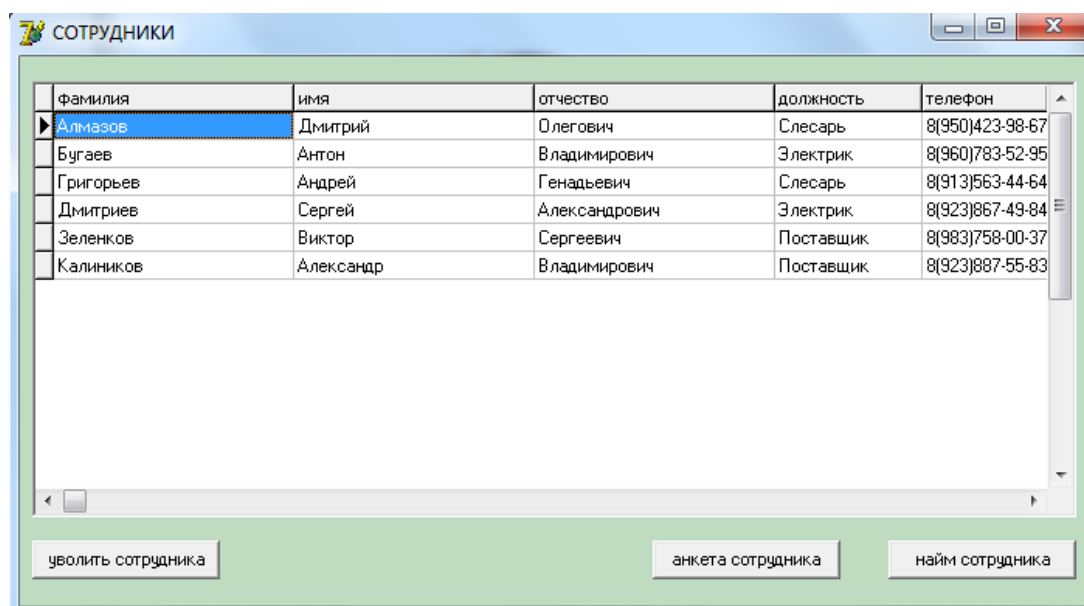
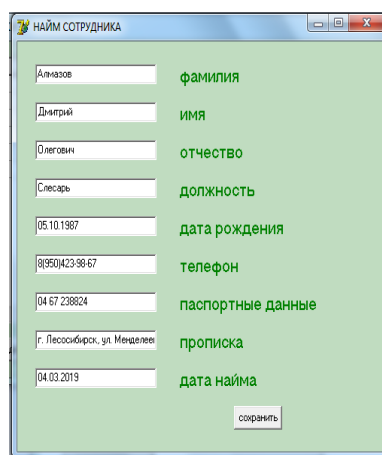


Рисунок 19 – Интерфейс приложения Сотрудники



Во время нажатия кнопки (рисунок 20) «анкета сотрудника», можно просмотреть более подробную информацию по конкретному сотруднику. В расширенной базе предусмотрены следующие пункты:

- 1) ФИО;
- 2) Должность;
- 3) Дата рождения;
- 4) Телефон;
- 5) Паспортные данные;
- 6) Прописка;
- 7) Дата найма.



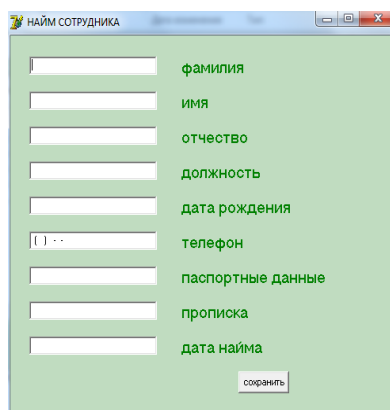
The screenshot shows a window titled "НАЙМ СОТРУДНИКА" with a green background. It contains a form with the following fields and values:

Алмазов	фамилия
Дмитрий	имя
Олегович	отчество
Слесарь	должность
05.10.1987	дата рождения
8(950)423-98-67	телефон
04 67 238824	паспортные данные
г. Лесосибирск, ул. Менделеев	прописка
04.03.2019	дата найма

A "сохранить" button is located at the bottom right of the form.

Рисунок 20 – Анкета сотрудника

Кнопка «найм сотрудника», запускает эту же анкету, но с чистым бланком для заполнения и последующего его сохранения в базе данных организации рисунок 21.



The screenshot shows the same window titled "НАЙМ СОТРУДНИКА" with a green background. The form fields are empty, ready for input:

	фамилия
	имя
	отчество
	должность
	дата рождения
	телефон
	паспортные данные
	прописка
	дата найма

A "сохранить" button is located at the bottom right of the form.

Рисунок 21 – Найм сотрудника

Кнопка «уволить сотрудника», удаляет анкету из таблицы и базы данных рисунок 22.

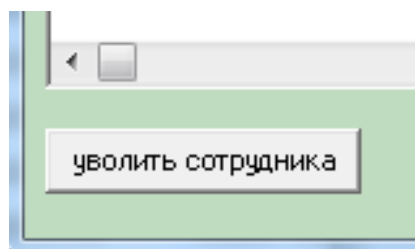


Рисунок 22 – Уволить сотрудника

Приложение «склад» для механика/заведующий складом (рисунок 23) будет состоять из нескольких пунктов:

- 1) Запрос;
- 2) Приход;
- 3) Расход.

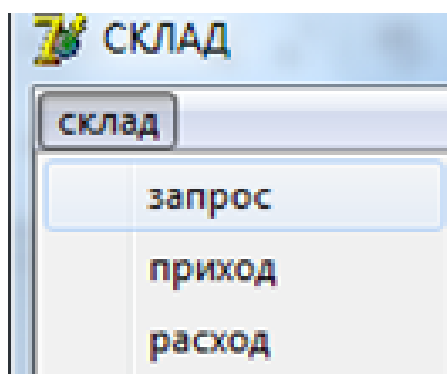


Рисунок 23 – Верхнее меню приложения Склад

На вкладке склад –запрос (рисунок 24) отображена таблица, в которой содержится краткая информация о товарах. В данной вкладке предусмотрен отбор и поиск комплектующих на складе.

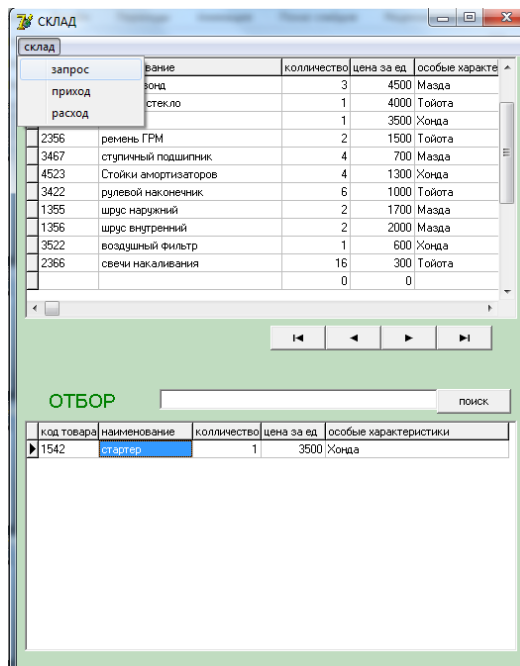


Рисунок 24 – Запрос

Вкладка склад – приход (рисунок 25) так же состоит из таблицы. При нажатии на кнопку «приход» можно добавить в базу основные параметры товара, а при нажатии на кнопку «просмотр прихода» отобразится информация о товаре, который находится на складе запчастей.

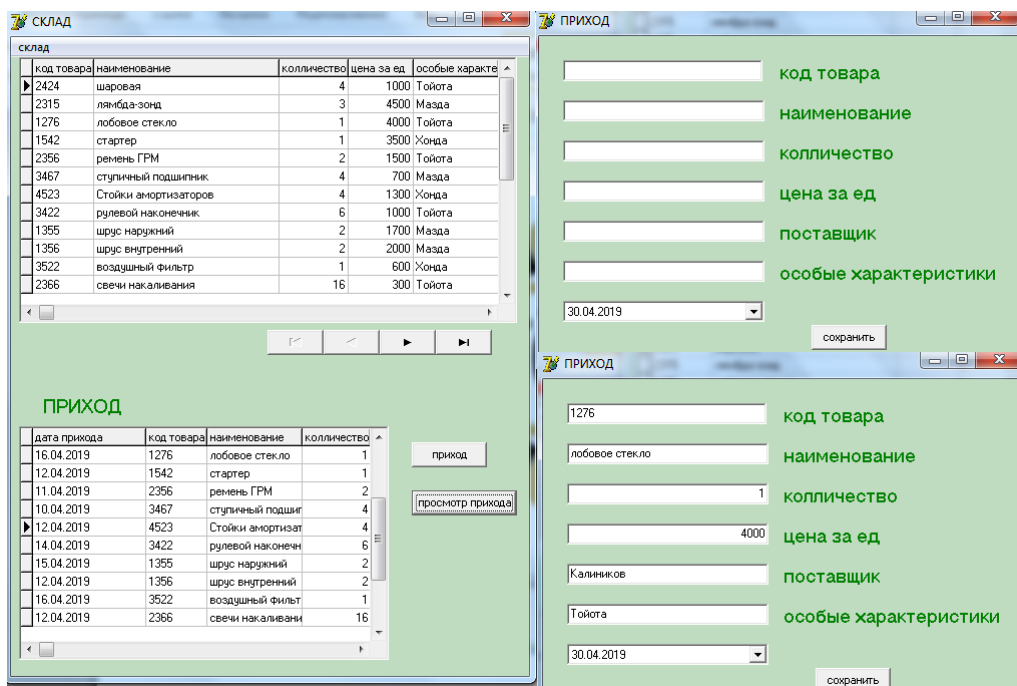


Рисунок 25 – Приход

Вкладка склад – расход (рисунок 26) представляет собой таблицу. Перейдя по кнопке «расход» можно добавить и сохранить краткое описание товара. При нажатии кнопки «просмотр расхода» будет отображена информация о конкретном товаре.

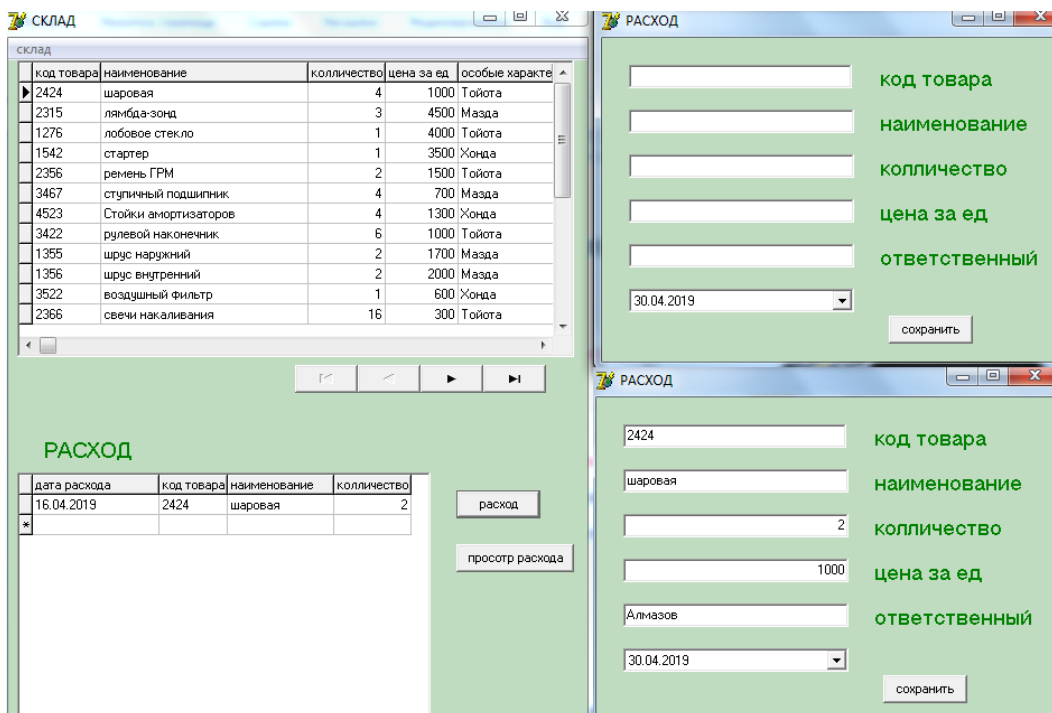


Рисунок 26 – Расход

Приложение «клиенты» (рисунок 27) будет состоять из нескольких пунктов:

- 1) Приём авто;
- 2) Сдача авто;
- 3) Запрос.

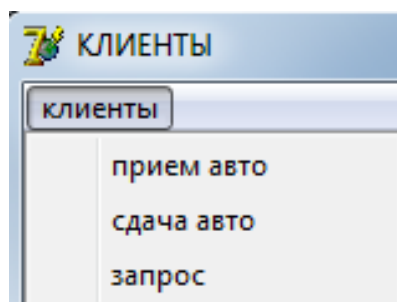


Рисунок 27 – Клиенты

При нажатии на вкладку клиенты – приём авто (рисунок 28) можно посмотреть таблицу клиентов, у которых авто в данный момент находится в ремонте. Так же предусмотрена возможность при нажатии кнопки «посмотреть за» увидеть краткую информацию об авто, дату его приёма и запрашиваемый ремонт.

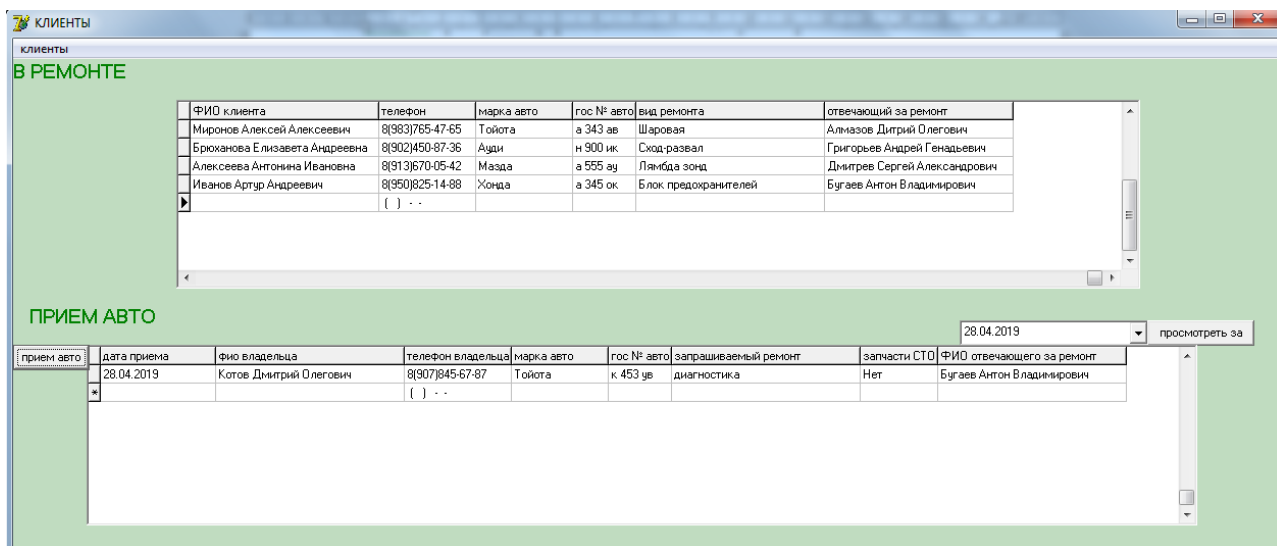


Рисунок 29 – Приём авто

Кнопка «приём авто» (рисунок 30) регистрирует в базе данных клиента, а так же есть возможность использовать запчасти сто либо свои.

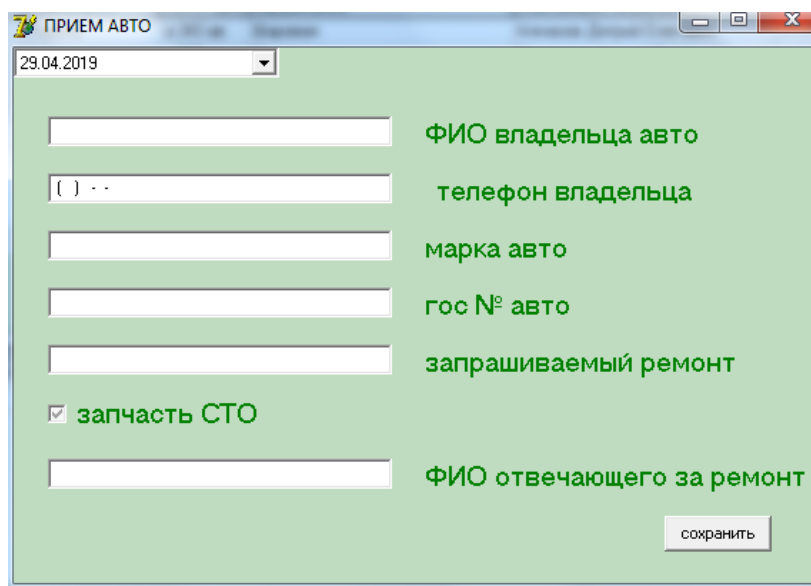


Рисунок 30 – Приём авто и регистрация

Вкладка клиенты – сдача авто (рисунок 31) так же отображает в таблице информацию о клиентах, у которых авто в данный момент находится в ремонте. Далее если нажать кнопку «просмотреть за» внизу в таблице отобразятся характеристики авто, которое было отремонтировано, а именно:

- 1) Дата окончания ремонта;
- 2) ФИО владельца;
- 3) Марка авто;
- 4) Гос. № авто;
- 5) ФИО отвечающего за ремонт;
- 6) Выполненные ремонтные работы;
- 7) Стоимость ремонта;
- 8) Запчасти СТО;
- 9) Стоимость запчастей;
- 10) Итого к оплате.

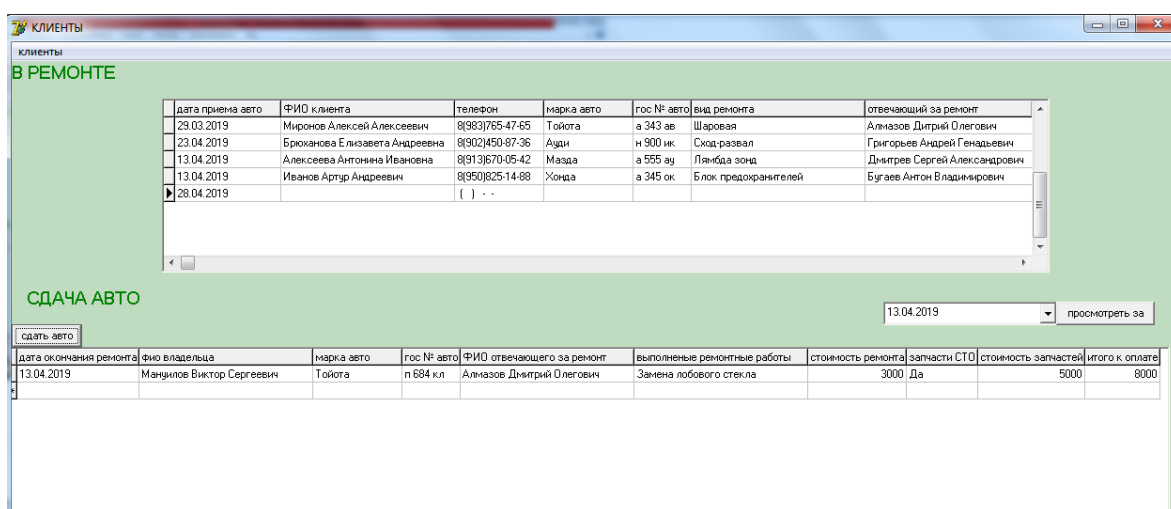


Рисунок 31 – Сдача авто

Перейдя по кнопке «сдача авто» (рисунок 32) следует ввести краткую информацию о клиенте, вид выполненной работы, а так же стоимость ремонта.

02.04.2019

Миронов Алексей Алексеевич **ФИО владельца авто**

Тойота **марка авто**

а 343 ав **гос № авто**

Алмазов Дмитрий Олегович **ФИО отвечающего за ремонт**

замена шаровой **выполненные ремонтные работы**

1500 **стоимость ремонта**

**запчасти СТО**

1000 **стоимость запчастей**

итого к оплате **2500**

**клиент оплатил**

Рисунок 32 – Сдача авто, оплата

На вкладке клиенты – запрос (рисунок 33) отображена таблица, в данной вкладке предусмотрен запрос поиска по гос. № авто.

КЛИЕНТЫ

**В РЕМОНТЕ**

дата приема авто	ФИО клиента	телефон	марка авто	гос № авто	вид ремонта	отвечающий за ремонт
29.03.2019	Миронов Алексей Алексеевич	8(983)765-47-65	Тойота	а 343 ав	Шаровая	Алмазов Дмитрий Олегович
23.04.2019	Брюханова Елизавета Андреевна	8(902)450-67-36	Ауди	н 900 ик	Сход-развал	Григорьев Андрей Геннадьевич
13.04.2019	Алексеева Антонина Ивановна	8(913)670-05-42	Мазда	а 555 ау	Лямбда зонд	Дмитриев Сергей Александрович
13.04.2019	Иванов Артур Андреевич	8(950)825-14-88	Хонда	а 345 ок	Блок предохранителей	Бугаев Антон Владимирович
28.04.2019		{ } . .				

запрос по гос № авто

Project1

ВВЕДИТЕ гос № авто

найти

ФИО владельца	марка авто	гос № авто	выполнен ремонт	ФИО отвечающего за ремонт	дата приема	дата сдачи	стоимость ремонта
Мануилов Виктор Сергеевич	Тойота	п 684 кл	Замена лобового стекла	Алмазов Дмитрий Олегович	10.04.2019	13.04.2019	8000

Рисунок 33 – Запрос по гос. № авто

В данном разделе рассмотрен готовый интерфейс с его полным описанием функционала. Все технические требования к визуализации программы выполнены.

## ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе по теме «Разработка базы данных по учету работы СТО (на примере «Автотехцентр»)» была спроектирована, запрограммирована программа. В ходе проведения анализа СТО «Автотехцентр», был сделан вывод, что лучшим способом хранения данных на станции технического обслуживания является специально адаптированная база данных. Были рассмотрены теоретические основы информационных систем на предприятии, а так же созданы организационная и процессная модели для лучшего представления организации процессов на предприятии.

Далее был разработан алгоритм создания программы. Выбрана система управления базами данных Microsoft Office Access, среда визуального программирования Delphi 7.0 и пользовательский интерфейс.

База данных позволила быстро и эффективно контролировать деятельность компании и вести учет заключаемых договоров. Удобный интерфейс программы, с одной стороны, позволяет легко ориентироваться в программе, не требуя от пользователя каких-либо специальных навыков работы с электронно-вычислительными машинами, с другой стороны предоставляет пользователю необходимую оперативную информацию. Данные сведения позволяют осуществлять достаточно точную работу организации, на основе которой можно анализировать её текущее положение среди конкурентов и планировать дальнейшую деятельность.

Разработанная программа помогает организации вести учет работы станции технического обслуживания, а так же повышает сохранность документов и удобства их хранения.

Таким образом, в ходе выполнения выпускной квалификационной работы в полном объеме решены все поставленные задачи и достигнута цель.



## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Алешин, Л.И. Информационные технологии : учебное пособие / Л.И. Алешин. – Москва : Маркет ДС, 2011. – 384 с.
2. Алиев, В.С. Информационные технологии и системы финансового менеджмента: учебное пособие / В.С. оглы Алиев. – Москва : Форум, ИНФРА-М, 2011. - 320 с.
3. Андерсен, В. Базы данных Microsoft Access. Проблемы и решения: Прост, пособ. / Пер. с англ. – Москва : Издательство ЭКОМ, 2001. – 384 с.
4. Архангельский, А.Я. Delphi 2006. Язык Delphi, классы, функции Win32 и NET: справочное пособие / А.Я. Архангельский. – Москва : Бином–Пресс, 2011. – 661 с.
5. Васильев, А. VBA в Office 2000: учебный курс/А. Васильев, А. Андреев. – Санкт–Петербург : 2001. – 432 с.
6. Ветитнев, А.М. Информационные технологии в социально-культурном сервисе и туризме. Оргтехника: учебное пособие / А.М. Ветитнев. – Москва : Форум, 2010. – 400 с.
7. Вдовин, В.М. Информационные технологии в налогообложении: практикум / В.М. Вдовин, Л.Е. Суркова. – Москва : Дашков и К, 2012. – 248 с.
8. Вдовин, В.М. Информационные технологии в финансово-банковской сфере: Практикум / В.М. Вдовин. – Москва : Дашков и К, 2012. – 248 с.
9. Виллариал, Б. Программирование Access 2002 в примерах: Пер. с англ. – Москва : КУДИЦ – ОБРАЗ, 2003.– 496 с,
10. Гаврилов, М.В. Информатика и информационные технологии: учебник для бакалавров / М.В. Гаврилов, В.А. Климов; Рецензент Л.В. Кальянов, Н.М. Рыскин. – Москва : Юрайт, 2013. – 378 с.
11. Гарнаев, А.Ю. Самоучитель VBA /А.Ю. Гарнаев. – Санкт–Петербург : БХВ – Петербург, 2001. – 512 с.

12. Гвоздева, В.А. Информатика, автоматизированные информационные технологии и системы: учебник / В.А. Гвоздева. – Москва : ИД ФОРУМ, НИЦ ИНФРА–М, 2013. – 544 с.
13. Голицына, О.Л. Информационные технологии: учебник / О.Л. Голицына, Н.В. Максимов, Т.Л. Партыка, И.И. Попов. – Москва : Форум, ИНФРА–М, 2013. – 608 с.
14. Голубенко, Н.Б. Информационные технологии в библиотечном деле / Н.Б. Голубенко. – Ростов-на-Дону : Феникс, 2012. – 282 с.
15. Горев, А. Эффективная работа с СУБД /А.Горев, Р. Ахаян, С. Макашарипов. – Санкт–Петербург : Питер, 1997. – 704 с.
16. Гофман, В.Э. Работа с базами данных в Delphi / В.Э. Гофман, А.Д. Хомоненко. – Санкт–Петербург : БХВ–Петербург, 2006. – 656 с.
17. Грэм, Малкольм. Программирование для Microsoft SQL SERVER 2000 с использованием XML. Пер. с англ. М. Гаврилов, Л.П. Информационные технологии в коммерции: учебное пособие / Л.П. Гаврилов. – Москва : НИЦ ИНФРА-М, 2013. – 238 с.
18. Диго, С.М. Проектирование и использование баз данных / С.М. Диго.– Москва : Финансы и статистика, 1995. – 487 с.
19. Жданова, М.М. Вопросы формирования профессионально важных качеств инженера Вестник Таджикского технического университета 2011. Т. 4. № – 4. С. 122–124.
20. Желонкин, А. Основы программирования в интегрированной среде DELPHI:— Санкт–Петербург, Бином. Лаборатория знаний, 2004 г. – 240 с.
21. Киммел, Пол. Освой самостоятельно программирование для Microsoft Access 2002 за 24 часа.: Пер. с англ. – Москва : Издательский дом "Вильямс", 2003. – 480 с.
22. Когловский, М.Р. Технология баз данных на персональных ЭВМ. – Москва. Финансы и статистика, 1992. – 315 с.

23. Козырев, А.А. Самоучитель работы на персональном компьютере. Учебное пособие. Изд. 2–е, переработанное и дополненное. Санкт–Петербург : издательство Михайлова В.А., 2006 304 с.
24. Культин, Н. Основы программирования в Turbo Delphi (+CD-ROM):— Санкт–Петербург, БХВ–Петербург, 2012 г. – 38 с.
25. Львович, И.Я. Разработка системы поддержки принятия, управленческих решений деятельности предприятия. Вестник Воронежского государственного технического университета 2007. Т. 3.№ 8. С. 191–192
26. Львович, Я.Е. Принятие проектных решений на основе формирования экспертно-виртуальной среды САПР Я.Е Львович. Д.С. Лковлев Вестник Воронежского института высоких технологий 2013. С. 128–130.
27. Магда, Ю. Разработка приложений Microsoft Office 2007 в Delphi:— Санкт–Петербург, БХВ–Петербург, 2009 г. – 160 с.
28. Преображенский, А.П. Некоторые проблемы при управлении предприятием Современные тенденции в экономике и управлении: новый взгляд 2010. С. 168–169.
29. Рыбин, Н.Н. Организационно- производственные структуры и управления технической службой предприятий автотранспортного комплекса // Издательство Курганского государственного университета, 2013–180с.
30. Фаронов, В. Delphi 2005. Язык, среда, разработка приложений:— Санкт–Петербург : Питер, 2007 г. – 560 с.
31. Хомоненко, А., Гофман В., Мещеряков, Е. Delphi 7 (+ CD-ROM):— Москва, БХВ–Петербург, 2010 г.- 1120 с.
32. Хомоненко, А. — Москва, Книга по Требованию, 2006 г. – 454 с.
33. Шпак, Ю. А. Разработка приложений в Delphi 2005/2006 (+ CD-ROM):— Москва : МК-Пресс, 2006 г. – 544 с.
34. <https://studfiles.net/preview/2012615/>
35. <https://studfiles.net/preview/5879150/>

## ПРИЛОЖЕНИЕ А

### Листинги программных модулей

#### ФОРМА 1 (Главная)

```
procedure TForm1.Button1Click(Sender: TObject);
begin
Form2.ShowModal; //переход на форму сотрудники
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
Form4.ShowModal; //переход на форму склад
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
Form7.ShowModal; //переход на форму клиенты
end;
```

#### ФОРМА 2 (Сотрудники)

```
procedure TForm2.Button1Click(Sender: TObject); //нажата кнопка
наймсотрудника,переход на форму анкеты
begin
Form2.ADOTable1.Append;
Form3.ShowModal;
end;

procedure TForm2.Button3Click(Sender: TObject); //
открытьанкетупросмотрсотрудника
begin
Form3.ShowModal;
end;

procedure TForm2.Button2Click(Sender: TObject); //
нажатакнопкауволитьсотрудника
begin
Form2.ADOTable1.Delete; // удаление записи о сотруднике
end;
```

#### ФОРМА 3 (Анкета сотрудника)

```
procedure TForm3.Button1Click(Sender: TObject); // кнопкасохранить
begin
ifForm2.ADOTable1.ModifiedthenForm2.ADOTable1.Post; //
еслиестьизмененияванкетесотрудника, сохранитьзапись
```

```
Close; \\ закрыть окно анкеты сотрудника
end;
```

## ФОРМА 4 (Склад)

```
procedure TForm4.FormCreate(Sender: TObject);
begin
  // очистка таблицы запроса
  Form4.ADOTable2.First;
  while not Form4.ADOTable2.Eof do
  begin
    Form4.ADOTable2.Delete;
  if Form4.ADOTable2.Modified then Form4.ADOTable2.Post;
    Form4.ADOTable2.Next;
  end;
end;

  procedure TForm4.N2Click(Sender: TObject); //
  настройка видимости объектов по меню отбор
  begin
    form4.DBGrid2.Visible:=true;
    form4.DBGrid3.Visible:=false;
    form4.DBGrid4.Visible:=false;
    form4.Label1.Visible:=true;
    form4.Label1.Caption:='ОТБОР';
    form4.Edit1.Visible:=true;
    form4.Button1.Visible:=true;
    form4.Button2.Visible:=false;
    form4.Button3.Visible:=false;
    form4.Button4.Visible:=false;
    form4.Button5.Visible:=false;
  end;

  procedure TForm4.Button1Click(Sender: TObject); //поиск по названию,
  формируем запрос
  begin
    Form4.ADOTable2.First;
    if Form4.Edit1.text="" then Showmessage('ВВЕДИТЕ название')
  \\если название не было введено вывести уведомляющее сообщение
  ELSE \\иначе
  begin
    //очистить таблицу от предыдущего запроса
    while not Form4.ADOTable2.Eof do
    begin
      Form4.ADOTable2.Delete;
    if Form4.ADOTable2.Modified then Form4.ADOTable2.Post;
      Form4.ADOTable2.Next;
    end;

    //формирование таблицы по запросу
    Form4.ADOTable1.First;
    while not Form4.ADOTable1.Eof do
    begin
```

```

if Form4.ADOTable1['наименование']=Form4.edit1.Text then
begin
    Form4.ADOTable2.insert;
    Form4.ADOTable2.FieldByName('кодтовара').AsString:=Form4.ADOTable1.FieldByName('кодтовара').AsString;
    Form4.ADOTable2.FieldByName('наименование').AsString:=Form4.ADOTable1.FieldByName('наименование').AsString;
    Form4.ADOTable2.FieldByName('количество').AsInteger:=Form4.ADOTable1.FieldByName('количество').AsInteger;
    Form4.ADOTable2.FieldByName('ценазаед').AsInteger:=Form4.ADOTable1.FieldByName('ценазаед').AsInteger;
    Form4.ADOTable2.FieldByName('особыехарактеристики').AsString:=Form4.ADOTable1.FieldByName('особыехарактеристики').AsString;
    if Form4.ADOTable2.Modified then Form4.ADOTable2.Post;
    Form4.ADOTable1.Next;
end
else Form4.ADOTable1.Next;
end;
    Form4.Edit1.Text:="";
end;
end;

procedure TForm4.N3Click(Sender: TObject); //
настройкавидимостиобъектоввменюприход
begin
    form4.DBGrid2.Visible:=false;
    form4.DBGrid3.Visible:=true;
    form4.DBGrid4.Visible:=false;
    form4.Label1.Visible:=true;
    form4.Label1.Caption:='ПРИХОД';
    form4.Edit1.Visible:=false;
    form4.Button1.Visible:=False;
    form4.Button2.Visible:=true;
    form4.Button3.Visible:=true;
    form4.Button4.Visible:=false;
    form4.Button5.Visible:=false;
end;

procedure TForm4.Button2Click(Sender: TObject); // нажатакнопкаприход
begin
    Form4.ADOTable3.Append;
    Form5.ShowModal; \\переходнаформуснакладнойприхода
end;

procedure TForm4.Button3Click(Sender: TObject); //нажатакнопкапросмотрприхода
begin
    Form5.ShowModal; \\переход на форму с накладной прихода
end;

procedure TForm4.N4Click(Sender: TObject); // настройка видимости объектов по меню расход
begin

```

```

form4.DBGrid2.Visible:=false;
form4.DBGrid3.Visible:=false;
form4.DBGrid4.Visible:=true;
form4.Label1.Visible:=true;
form4.Label1.Caption:='РАСХОД';
form4.Edit1.Visible:=false;
form4.Button1.Visible:=false;
form4.Button2.Visible:=false;
form4.Button3.Visible:=false;
form4.Button4.Visible:=true;
form4.Button5.Visible:=true;
end;

```

```

procedure TForm4.Button4Click(Sender: TObject); // нажатакнопкарасход
begin
  Form4.ADOTable4.Append;
  Form6.ShowModal; \\переходнаформуснакладнойрасхода
end;

```

```

procedure TForm4.Button5Click(Sender: TObject); //нажатакнопкапросмотррасхода
begin
  Form6.ShowModal;\\переход на форму с накладной расхода
end;

```

### **ФОРМА 5 (Накладнаяприхода)**

```

procedure TForm5.FormCreate(Sender: TObject);
begin
  form5.DateTimePicker1.Date:=Date(); \\получитьдату
end;

```

```

procedure TForm5.Button1Click(Sender: TObject);
begin
  \\формированиетаблицыпоприходу
  if Form4.ADOTable3.Modified then begin
    Form4.ADOTable3.FieldByName('датаприхода').Value:=trunc(form5.DateTimePicker1.
date);
    Form4.ADOTable3.Post; end;
  \\формирование таблицы склад
  \\если код прихода совпадает с имеющимся на складе + к количеству
  Form4.ADOTable1.First;
  while not Form4.ADOTable1.Eof do begin
    if form4.ADOTable1['кодтовара']= form4.ADOTable3['кодтовара'] then
      begin
        x:= strtoint(Form4.ADOTable1['количество']);
        y:= strtoint(Form4.ADOTable3['количество']);
        x:=x+y;
        form4.adotable1.edit;
        form4.ADOTable1['количество']:=inttostr(x);
        if Form4.ADOTable1.Modified then Form4.ADOTable1.Post; break;
      end
    else Form4.ADOTable1.Next; end;

```

```

//если достигли конца таблицы Склад и условие совпадения кода не выполнилось, то
создание новой записи с копированием данных из таблицы Приход
if (Form4.ADOTable1.eof) and
(Form4.ADOTable1.FieldByName('кодтовара').AsString<>Form4.ADOTable3.FieldByName('
кодтовара').AsString) then
begin
Form4.ADOTable1.insert;
Form4.ADOTable1.FieldByName('кодтовара').AsString:=Form4.ADOTable3.FieldByName('ко
дтовара').AsString;
Form4.ADOTable1.FieldByName('наименование').AsString:=Form4.ADOTable3.FieldByNam
e('наименование').AsString;
Form4.ADOTable1.FieldByName('количество').AsInteger:=Form4.ADOTable3.FieldByName
('количество').AsInteger;
Form4.ADOTable1.FieldByName('ценазаед').AsInteger:=Form4.ADOTable3.FieldByName('це
назаед').AsInteger;
Form4.ADOTable1.FieldByName('особыххарактеристики').AsString:=Form4.ADOTable3.Fiel
dByName('особыххарактеристики').AsString;
end;
Close; \\закреть накладную прихода
end;

```

### **ФОРМА 6 (Накладнаярасхода)**

```

procedure TForm6.FormCreate(Sender: TObject);
begin
form6.DateTimePicker1.Date:=Date(); //получитьдату
end;

procedure TForm6.Button1Click(Sender: TObject);
begin
// формирования таблицы по расходу
if Form4.ADOTable4.Modified then
begin
Form4.ADOTable1.First;
// если код прихода совпадает с имеющимся кодом на складе
while not Form4.ADOTable1.Eof do
begin
if form4.ADOTable1['кодтовара']= form4.ADOTable4['кодтовара'] then
begin

x:= strtoint(Form4.ADOTable1['количество']);
y:= strtoint(Form4.ADOTable4['количество']);
//если кол-во таблицы РАСХОД не превышает кол-во таблицы СКЛАД. Форируем
таблицу РАСХОД, изменяем кол-во в таблице СКЛАД
if x>y then
begin
Form4.ADOTable4.FieldByName('датарасхода').Value:=trunc(form6.DateTimePicker1.date);
Form4.ADOTable4.Post;

z:=x-y;
form4.adotable1.edit;
form4.ADOTable1['количество']:=inttostr(z);
if Form4.ADOTable1.Modified then Form4.ADOTable1.Post; break;

```



```

end;
//если кол-во Таблицы расход=количеству таблицы склад, формируем таблицу расход,
удаляем строку из таблицы склад
if x=y then
begin
Form4.ADOtable4.FieldName('датарасхода').Value:=trunc(form6.DateTimePicker1.date);
Form4.ADOtable4.Post;
Form4.ADOtable1.Delete;
if Form4.ADOtable1.Modified then Form4.ADOtable1.Post; break;
end;
//если данного количества на складе нет
if x<y then
begin
Form4.ADOtable4.Delete;
Showmessage('товар с данным кодом, в указанном количестве на складе отсутствует');
break;
end;
end
else Form4.ADOtable1.Next;
end;
end;
Close; \\закреть накладную расхода
end;

```

### **ФОРМА 7 (Окно клиентов)**

```

procedure TForm7.N2Click(Sender: TObject); // видимость объектов по меню прием
begin
form7.DBGrid2.Visible:=true;
form7.Button2.Visible:=true;
form7.Label2.Visible:=true;
form7.Label2.Caption:='ПРИЕМАВТО';
form7.DBGrid3.Visible:=false;
Form7.Button3.Visible:=false;
form7.DateTimePicker1.Visible:=true;
form7.Button1.Visible:=true;
form7.DBGrid4.Visible:=false;
form7.Edit1.Visible:=false;
form7.Button4.Visible:=false;

DateTimePicker1.Date:=Date(); \\получить дату
\\ фильтр по дате
Form7.ADOtable2.Filtered:=false;
Form7.ADOtable2.Filter:='[датаприема]='+DateToStr(DateTimePicker1.Date);
Form7.ADOtable2.Filtered:=true;
end;

procedure TForm7.N3Click(Sender: TObject); // видимость объектов по меню сдача авто
begin
form7.DBGrid3.Visible:=true;
form7.Button3.Visible:=true;
form7.Label2.Visible:=true;

```

```

form7.Label2.Caption:='СДАЧААВТО';
form7.DBGrid2.Visible:=false;
Form7.Button2.Visible:=false;
form7.DateTimePicker1.Visible:=true;
form7.Button1.Visible:=true;
form7.DBGrid4.Visible:=false;
form7.Edit1.Visible:=false;
form7.Button4.Visible:=false;

\\ фильтр по дате
Form7.ADOtable3.Filtered:=false;
Form7.ADOtable3.Filter:='[датаокончанияремонта]='+DateToStr(DateTimePicker1.Date);
Form7.ADOtable3.Filtered:=true;
end;

procedure TForm7.Button2Click(Sender: TObject); // переход к форме – анкета приема авто
begin
    Form7.ADOtable2.Append;
    Form8.ShowModal;
end;

procedure TForm7.Button3Click(Sender: TObject); // переход к форме – анкета сдачи авто
begin
    Form7.ADOtable3.Append;
    Form9.ShowModal;
end;

procedure TForm7.Button1Click(Sender: TObject); // фильтр по дате при изменении даты
begin
    \\ если изменена дата приема авто
    Form7.ADOtable2.Filtered:=false; //2
    Form7.ADOtable2.Filter:='[датаприема]='+DateToStr(DateTimePicker1.Date);
    Form7.ADOtable2.Filtered:=true;

    \\ если изменена дата окончания ремонта
    Form7.ADOtable3.Filtered:=false; //3
    Form7.ADOtable3.Filter:='[датаокончанияремонта]='+DateToStr(DateTimePicker1.Date);
    Form7.ADOtable3.Filtered:=true;
end;

procedure TForm7.N4Click(Sender: TObject); // настройка видимости объектов по меню запрос
begin
    form7.DBGrid3.Visible:=false;
    form7.Button3.Visible:=false;
    form7.Label2.Visible:=true;
    form7.Label2.Caption:='запрос по гос № авто';
    form7.DBGrid2.Visible:=false;
    Form7.Button2.Visible:=false;
    form7.DateTimePicker1.Visible:=false;
    form7.Button1.Visible:=false;

```

```

form7.DBGrid4.Visible:=true;
form7.Edit1.Visible:=true;
form7.Button4.Visible:=true;

//очистить таблицу от предыдущего запроса
form7.ADOTable4.First;
while not Form7.ADOTable4.Eof do
begin
    Form7.ADOTable4.Delete;
    Form7.ADOTable4.Next;
if Form7.ADOTable4.Modified then Form7.ADOTable4.Post;
end;
end;
procedure TForm7.Button4Click(Sender: TObject); \\ нажата кнопка найти
begin
Form7.ADOTable2.First;
Form7.ADOTable3.First;

if Form7.Edit1.text="" then Showmessage('ВВЕДИТЕ гос № авто') \\если гос № не был задан, вывести уведомляющее сообщение
ELSE \\иначе
begin
    \\формирование таблицы по запросу
Form7.ADOTable2.First;
while not Form7.ADOTable2.Eof do
begin
if Form7.ADOTable2['гос № авто']=Form7.edit1.Text then
begin
                Form7.ADOTable4.insert;
Form7.ADOTable4.FieldName('ФИОвладельца').AsString:=Form7.ADOTable2.FieldName('фиовладельца').AsString;
Form7.ADOTable4.FieldName('маркаавто').AsString:=Form7.ADOTable2.FieldName('маркаавто').AsString;
Form7.ADOTable4.FieldName('гос № авто').AsString:=Form7.ADOTable2.FieldName('гос № авто').AsString;
Form7.ADOTable4.FieldName('ФИОотвечающегозаремонт').AsString:=Form7.ADOTable2.FieldName('ФИОотвечающегозаремонт').AsString;
Form7.ADOTable4.FieldName('датаприема').Value:=Form7.ADOTable2.FieldName('датаприема').Value;
                Form7.ADOTable2.Next;
end
else Form7.ADOTable2.Next;
end;

    Form7.ADOTable3.First;
while not Form7.ADOTable3.Eof do
begin
if Form7.ADOTable3['гос № авто']=Form7.edit1.Text then
begin
                Form7.ADOTable4.insert;

```

```

Form7.ADOTable4.FieldByName('выполненремонт').AsString:=Form7.ADOTable3.FieldByN
ame('выполненыремонтныеработы').AsString;
Form7.ADOTable4.FieldByName('датасдачи').Value:=Form7.ADOTable3.FieldByName('дата
окончанияремонта').Value;
Form7.ADOTable4.FieldByName('стоимостьремонта').Asinteger:=Form7.ADOTable3.FieldBy
Name('итогокоплате').Asinteger;
                Form7.ADOTable3.Next;
end
else Form7.ADOTable3.Next;
end;
if Form7.ADOTable4.Modified then Form7.ADOTable4.Post;
    Form7.Edit1.Text:="";
end;
end;

```

### **ФОРМА 8 (Анкета приема авто)**

```

procedure TForm8.FormCreate(Sender: TObject); // получить дату
begin
    form8.DateTimePicker1.Date:=Date();
end;

procedure TForm8.Button1Click(Sender: TObject); // сохранение приема
begin
    // формирование таблицы приема
    if Form7.ADOTable2.Modified then begin
        Form7.ADOTable2.FieldByName('дата приема').Value:=trunc(form8.DateTimePicker1.date);
        Form7.ADOTable2.Post;
        // формирование таблицы в ремонте
        Form7.ADOTable2.last;
        Form7.ADOTable1.insert;
        Form7.ADOTable1.FieldByName('ФИО клиента').AsString:=Form7.ADOTable2.FieldByName
('ФИО владельца').AsString;
        Form7.ADOTable1.FieldByName('телефон').AsString:=Form7.ADOTable2.FieldByName('теле
фон владельца').AsString;
        Form7.ADOTable1.FieldByName('марка авто').AsString:=Form7.ADOTable2.FieldByName('м
арка авто').AsString;
        Form7.ADOTable1.FieldByName('гос №
авто').AsString:=Form7.ADOTable2.FieldByName('гос № авто').AsString;
        Form7.ADOTable1.FieldByName('вид ремонта').AsString:=Form7.ADOTable2.FieldByName('
запрашиваемый ремонт').AsString;
        Form7.ADOTable1.FieldByName('отвечающий за ремонт').AsString:=Form7.ADOTable2.Field
ByName('ФИО отвечающего за ремонт').AsString;
        Form7.ADOTable1.FieldByName('дата приема авто').Value:=trunc(form8.DateTimePicker1.dat
e);
        Form7.ADOTable1.Post;
    end;
    Close; // закрыть анкету приема
end;

```

### **ФОРМА 9 (Анкета сдачи авто)**

```

procedure TForm9.FormCreate(Sender: TObject);
begin
  form9.DateTimePicker1.Date:=Date();//получить дату
  form9.Label8.Caption:=""; \\ обнуление суммы оплаты
end;

procedure TForm9.DBCheckBox1Click(Sender: TObject); // если нажата кнопка что
запчасти СТО, отобразить ввод стоимости
begin
  if form9.DBCheckBox1.Checked=true then begin
    form9.DBEdit7.Visible:=true;
    form9.Label1.Visible:=true;
  end;
  \\ иначе не отображать
  If form9.DBCheckBox1.Checked<>true then begin
    form9.DBEdit7.Visible:=false;
    form9.Label1.Visible:=false;
  end;
end;

procedure TForm9.Button2Click(Sender: TObject); // подсчет итого к оплате
begin
  if form9.DBCheckBox1.Checked=true then begin
    x:=Form7.ADOTable3['стоимость ремонта']; y:=Form7.ADOTable3['стоимость запчастей'];
  end
  else begin x:=Form7.ADOTable3['стоимость ремонта']; y:=0;end;
  x:=x+y;
  form9.Label8.Caption:=inttostr(x);
end;

procedure TForm9.Button1Click(Sender: TObject); // продолжить работу,
если клиент оплатил, внести данные в базу выдачи авто
begin
  if form9.CheckBox1.Checked=true then begin
    Form7.ADOTable3.FieldByName('дата окончания ремонта').Value:=trunc(form9.DateTimePicker1.date);
    form7.ADOTable3.FieldByName('итого к оплате').ASInteger:=x;
    Form7.ADOTable3.Post;
    Form7.ADOTable1.First;
    while not Form7.ADOTable1.Eof do
      begin
        if form7.ADOTable1.FieldByName('год №
авто').ASstring=form7.ADOTable3.FieldByName('год № авто').ASstring then
          Form7.ADOTable1.Delete;
          Form7.ADOTable1.Next;
        end;
        form9.Label8.Caption:="";
      end;
    close;
  end;
  \\ если клиент не оплатил, авто не сдано
  if form9.CheckBox1.Checked<>true then Form7.ADOTable3.Delete;

```

```
form9.Label8.Caption:="";  
close;\\ закрытьанкетусдачиавто  
end;
```

### **ФОРМА 10 (Пароль – 555)**

```
procedure TForm10.Button1Click(Sender: TObject);  
begin  
if form10.Edit1.text='555' then begin form1.ShowModal; form10.Close; end else  
showmessage('неправильныйпароль');  
end;
```

## ПРИЛОЖЕНИЕ Б

### Научно публицистическая статья

09.03.02.07 Информационно-управляющие системы

#### Разработка базы данных по учету работы СТО (на примере «Автотехцентр»)

**Мальцев Александр Владимирович**

студент группы ЛФ ЗФМФ14-01БИ направления подготовки 09.03.02 информационные системы и технологии.

**Ахтамова Светлана Станиславовна**

научный руководитель, кандидат педагогических наук, доцент, доцент кафедры высшей математики, информатики и естествознания

#### **Аннотация**

#### **Предмет.**

Предметом исследования является «Разработка базы данных по учету работы СТО (на примере «Автотехцентр»)».

#### **Цели.**

Целью работы является разработка базы данных по учету работы станции технического обслуживания

**Методология.** Для достижения поставленной цели необходимо решить следующие задачи:

- Проанализировать научную и специальную литературу по теме исследования.
- Раскрыть теоретические основы базы данных.
- Спроектировать организационную и процессную модели СТО.

**Ключевые слова:** информационная система, базы данных, проектирование, модель.

#### **(текст статьи)**

Автомобильный транспорт представляет собой отрасль народного хозяйства, обладает значительными резервами повышения эффективности использования подвижного состава, кроме этого выполняет важную роль для

экономики нашей страны.

Во время использования транспортного средства постепенно снижаются его эксплуатационные свойства из-за изнашивания материалов, появления коррозии, повреждения и поломок деталей и узлов, усталости свойств материала, из которого они были изготовлены. Из-за этого в машине образуются различные поломки, которые в итоге сокращают эффективность его использования. Для того чтобы снизить вероятность появления дефектов и своевременно устранить их создается станция технического обслуживания (ТО) и ремонта.

Одна из основных областей применения компьютеров является хранение и переработка больших объемов информации в различных сферах деятельности человека: в экономике, банковском деле, торговле, транспорте, медицине, науке и т. д.

Информационная система – это система, в которой реализуется автоматизированный сбор, обработка и манипулирование данными, включающая технические средства для обработки данных, программное обеспечение и обслуживающий персонал.

Целью любой информационной системы является обработка данных об объектах реального мира. Основа информационной системы – база данных. В широком смысле слова база данных – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. Предметная область – часть реального мира, которая подлежит изучению для организации управления его объектами и, в конечном итоге, автоматизации, например предприятие, вуз и т. д.

Для того чтобы создать базу данных, пользователю нужно упорядочить информацию по разным признакам и быстро произвести выборку с произвольным сочетанием признаков. При этом необходимо выбрать правильную модель данных. Модель данных – это формальное представление ключевых категорий восприятия реального мира, которые представлены его объектами, связями, свойствами, а также их



взаимодействиями.

Система управления базами данных (СУБД) – система программ, которая позволяет создавать базу данных, обновлять хранимую в ней информацию, обеспечить удобный доступ к ней с целью просмотра и поиска.

Организационная модель создается в начале моделирования бизнес-процессов. В иерархической модели отображаются подразделения предприятия, наименование должностей, фамилии руководителей подразделений и отдельных сотрудников.

Сложные иерархические структуры могут быть декомпозированы, например, структура подразделения может быть раскрыта на более детальной схеме. Также могут быть построены модели иерархии подчиненности во временных коллективах, группах и т.д. Все объекты организационных моделей могут быть использованы при построении моделей бизнес-процессов.

Организационная модель (рисунок 1) предприятия станции технического обслуживания выглядит следующим образом:



Рисунок 1 – Организационная модель СТО

В управлении предприятия стоит директор, в его подчинении находится механик (он выполняет функцию контроля за исполнением работы по ремонту и обслуживанию транспортного средства, а так же он отвечает за контроль поставки запчастей от курьера). Далее следом идет штат рабочих

(отвечающих за услуги выполнения работ по обслуживанию транспортных средств), затем курьер (поставляет расходные материалы автозапчастей).

На основе созданной модели, ярко выделены связи отчетностей. Для разработки информационной системы требуется создание одного прикладного приложения, формирующего отчет деятельности работы предприятия в виде базы данных.

Процессная модель представляет собой описание предприятия, как целостная система со всеми неразрывно связанными в ней бизнес-процессами, структурными подразделениями, производственными и управленческими функциями.

В каждой организации ежедневно протекает множество процессов. Чтобы систематизировать и упорядочить бизнес-процессы, протекающие на предприятии, их принято делить на следующие основные группы:

- Основные процессы: генерируют доходы компании. Примерами процессов закупки, продажи и производства продукта; ее складирования и транспортировки.

- Процессы обеспечения: поддерживают инфраструктуру компании. В эту категорию входит техническое, информационное и административно – хозяйственное обеспечение; обеспечение безопасности, социальной сферы и финансовых операций; консультационное и аудиторское обеспечение.

- Процессы управления: управляют компанией. К этой категории относят такие процессы как процессы управления финансами, маркетингом, персоналом и логистикой; экономическое, стратегическое и организационное управление, а также управленческий учет.

- Процессы развития: развивают компанию. Представляют инвестиционные виды деятельности, нацеленные на прибыль от сегодняшних вложений в долгосрочной перспективе.

На основании анализа осуществляемых процессов, на предприятии формируется процессная модель (рисунок 2):

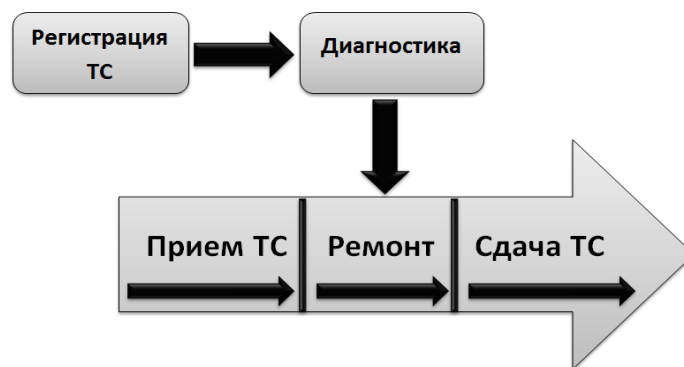


Рисунок 2 – Процессная модель станции технического обслуживания

Описание процесса:

- Приём заявки заказчика на обслуживания транспортного средства;
- Регистрация транспортного средства в базе данных;
- Диагностики выявления неисправностей транспортного средства;
- Ремонт выявленных неисправностей транспортного средства;
- Формирование отчёта по выполненной работе для последующей сдачи транспортного средства заказчику.

Для того чтобы результат на выходе максимально соответствовал ожиданиям клиента, предприятие нуждается в слаженной организованной информационной системе, которая бы помогала персоналу в осуществлении процессов, и могла хранить большие объемы данных.

Таким образом, спроектированная информационная система готова к реализации. Для её реализации необходимо разработать программный код, спроектировать пользовательский интерфейс и внедрить в действующее предприятие.

Список используемых источников:

1. Гаврилов, М.В. Информатика и информационные технологии: Учебник для бакалавров / М.В. Гаврилов, В.А. Климов; Рецензент Л.В. Кальянов, Н.М. Рыскин. - М.: Юрайт, 2013. - 378 с.

2. Диго, С.М. Проектирование и использование баз данных. - М.: Финансы и статистика, 1995. - 487 с.

3. Когловский, М.Р. Технология баз данных на персональных ЭВМ. - М. Финансы и статистика, 1992. - 315 с.

## ПРИЛОЖЕНИЕ В

### Сертификат о публикации



## СЕРТИФИКАТ УЧАСТНИКА

XXIV МЕЖДУНАРОДНАЯ  
НАУЧНО-ПРАКТИЧЕСКАЯ  
КОНФЕРЕНЦИЯ

Научный форум:  
технические и физико-  
математические науки

13.05.2019

Дата

Настоящий сертификат подтверждает участие

*Мальцев Александр Владимирович,  
Ахтамова Светлана Станиславовна*

и публикацию в сборнике трудов конференции статьи:

«Разработка базы данных по учету работы СТО (на примере  
«Автотехцентр»)»



Лебедева Н. А.

Председатель Оргкомитета

г. Москва