

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Информационные системы

«Допущен к защите»
Заведующий кафедрой _____

Мамганов И.И.
(Ф.И.О., ученая степень, звание)

« _____ » _____ 20__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка БДП данных «Абитуланта»

Специальность Информационные системы

Выполнил (а) Талан Дина Дамировна ИИ-10-2
(Фамилия и инициалы) группа

Научный руководитель Алексов В.В., к.т.н., старший преподаватель
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Башиева А.И., к.э.н., доцент
(Фамилия и инициалы, ученая степень, звание)
« 04 » _____ 2014 г.
(подпись)

по безопасности жизнедеятельности:

Сейталиев А.С., старший преподаватель
(Фамилия и инициалы, ученая степень, звание)
« 09 » _____ 2014 г.
(подпись)

по применению вычислительной техники:

Каирбаева Б.К., к.т.н., старший преподаватель
(Фамилия и инициалы, ученая степень, звание)
« 10 » _____ 06 2014 г.
(подпись)

Нормоконтролер: Али А.Т., к.т.н., доцент

(Фамилия и инициалы, ученая степень, звание)
« 16 » _____ 06 2014 г.
(подпись)

Рецензент: Сайбагышев М.И., к.т.н., доцент
(Фамилия и инициалы, ученая степень, звание)

« 16 » _____ 06 2014 г.
(подпись)

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет информационных технологий
Специальность 5В070300 - Информационные системы
Кафедра информационных систем

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Талан Дина Дашоркина
(фамилия, имя, отчество)

Тема проекта Разработка базы данных "Автозапчасти"

утверждена приказом ректора № 115 от «24» сентября 2013 г.
Срок сдачи законченной работы «12» 06 2014 г.
Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

Постановка задачи дипломного проекта, выборные программные и инструментальные средства и их обоснование, анализ существующих аналоговых БД, Проектирование и разработка БД "Автозапчасти", архитектура БД, схемы, АМ-диаграммы, пользовательский интерфейс БД "Автозапчасти", описание функций БД, Технико-экономическое обоснование проекта, Безопасность существующих, Вопросы, Текстовое задание, список программ

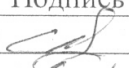
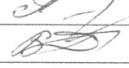

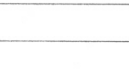
Перечень графического материала (с точным указанием обязательных чертежей)

1. Взаимодействие неформальной группы
2. Алгоритм функционирования группы
3. Архитектура автоматизированной системы контроля доступа частей и частей здания
4. Диаграммы рекомендаций
5. Диаграммы видов деятельности
6. Диаграммы субфункций
7. Диаграммы взаимодействия
8. Логическая схема баз данных
9. Функциональная схема баз данных

Рекомендуемая основная литература

1. Грабер М. SQL: 1 пер. с англ 1 / М. Грабер - М: Издательство "Лори", 2003 - 854 стр
2. Основы ИМЛ - Разработка диаграмм в среде Rational Rose. Инструкция, курс 10 - Ресурс Доступен: <http://2programmer.ru/any>
3. Величкова А.И. Методические указания к выполнению задания в цифровой среде (для студентов всех уровней обучения) - Астана: АУЭС, 2013 - 24 с
4. Величкова А.И. Методические указания к выполнению задания "Техническое задание" - Астана: АУЭС, 2009

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
ВМД	Величкова А.И.	10.06.14	
Экспертная часть	Величкова А.И.		
Основная часть	Дыков В.В.		
Специальная часть	Дыков В.В.		

предусматривает автоматизацию контроля и учета автозапчастей, регистрацию клиентов и резервное копирование данных в форме отчета.

База данных «Автозапчасти» разработана с целью оптимизирования работы оператора.

Аңдатпа

Дипломдық жобада «Автозапчасти» мәліметтер қоры құрастырылды. Программалық жүйе Borland Delphi 7 құрамында жасалынды. Жүйе

автоқұрылғыларды бақылаудың және санаудың автоматтандырылғанын, сатып алушылардың тіркелуін және есеп беру түрінде резервтік көшіруді қарастырады.

«Автозапчасти» мәліметтер қоры оператордың жұмысын оптимизациялау мақсатымен құрастырылды.

Annotation

In the degree project the database «Автозапчасти» is developed. The program system is developed in the environment of Borland Delphi 7. The system

provides automation of control and the accounting of auto parts, registration of clients and backup of data in the form of the report.

The database «Автозапчасти» is developed for the purpose of an optimization the work of the operator.

Содержание

Введение	9
1 Общее описание	11
1.1 Постановка задачи дипломного проекта	11

1.2 Выбранные программные и инструментальные средства и их обоснование	13
1.3 Анализ существующих на рынке БД «Автозапчасти»	16
2 Проектирование и разработка БД «Автозапчасти»	19
2.1 Требования к системе	19
2.2 Архитектура системы	19
2.3 Разработка UML-диаграмм БД «Автозапчасти»	20
2.4 Проектирование БД «Автозапчасти»	25
3 Разработка БД «Автозапчасти»	28
3.1 Пользовательский интерфейс БД «Автозапчасти»	28
3.2 Описание функций БД «Автозапчасти»	31
4 Техничко-экономическое обоснование проекта	38
4.1 Техничко-экономическое обоснование	38
4.2 Расчет трудоемкости разработки ПП	38
4.3 Расчет затрат на разработку ПП	39
4.4 Определение возможной (договорной) цены ПП	44
4.5 Расчет срока окупаемости ПП	44
4.6 Оценка социально - экономических результатов функционирования программного продукта	45
5 Безопасность жизнедеятельности	46
5.1 Анализ условий труда	46
5.2 Расчет естественного освещения	48
5.3 Расчет искусственного освещения	50
5.4 Расчет установок водяного и пенного пожаротушения	53
Заключение	56
Список литературы	58
Список сокращений	59
Приложение А Техническое задание	60
Приложение Б Листинг программы	74

Введение

Бурное развитие информационных технологий привело к их внедрению во все сферы деятельности. Не являются исключением и предприятия автосервиса. Большой объем данных о запчастях уже нецелесообразно хранить в бумажном виде и обрабатывать вручную. Поэтому проблема

автоматизации документооборота в предприятиях на сегодняшний день весьма актуальна.

В последнее время все чаще употребляются термины «электронный документ», «электронный документооборот», которые уже давно и прочно вошли в лексикон специалистов. Исследователи обращают внимание на постоянный рост объемов бумажной и электронной документации, причем темпы роста объемов электронных документов выше, чем бумажных. Увеличиваются затраты времени и средств на обработку документов. В первую очередь это касается рабочего времени (сотрудников и руководителей), технических ресурсов, расходных материалов и др.

Основными операциями, связанными с обработкой документов, являются: создание и редактирование, добавление информации о новых клиентах, хранение и поиск, контроль исполнения. Автоматизация позволяет снизить затраты на обработку документов, повысить качество управленческих решений за счет ускорения процессов поиска, обработки и предоставления нужной пользователю информации, что, в результате, обеспечивает рост эффективности функционирования управляемого объекта.

В настоящем дипломном проекте рассматриваются вопросы разработки программной системы учета клиентов и автозапчастей.

Дипломный проект состоит из введения, пяти разделов основной части пояснительной записки, заключения, списка литературы и приложений.

В первом разделе проведён подробный анализ предметной области, представлены его результаты, а также сформулированы задачи дипломного проекта, выбор средств программирования для решения поставленной задачи – создание базы данных «Автозапчасти». Реализация базы данных проводилась с использованием программ Borland Delphi 7, MS SQL Server Management Studio 2008.

Во втором разделе дано описание реализации базы данных, в частности выбор средств проектирования.

В третьем разделе содержатся инструкции и рекомендации по эксплуатации разработанного программного продукта по теме дипломной работы (проекта).

В четвёртом разделе приведено технико-экономическое обоснование проекта с оценкой экономической эффективности проекта и расчётом окупаемости базы данных «Автозапчасти».

Пятый раздел посвящен вопросам обеспечения жизнедеятельности персонала, охраны труда и промышленной экологии. В разделе анализируются вредные для человека факторы, связанные с использованием проектируемого объекта и предлагаются мероприятия, направленные на максимальное снижение последствий этих факторов.

В заключении рассмотрены основные итоги дипломного проекта и намечены перспективные направления дальнейшего развития.

1 Общее описание

1.1 Постановка задачи дипломного проекта

Системы автоматизации делопроизводства и электронного документооборота (САДЭД) как системы, позволяющие автоматизировать основные процедуры делопроизводства предприятия, охватывают процессы создания, обработки, тиражирования, передачи, хранения документов, контроля их исполнения и коллективной работы над документами и предназначены для эффективного управления современным предприятием. Быстрый темп развития экономики создает высокую востребованность САДЭД предприятиями. В разнообразии предложений на рынке СЭД, возникает вопрос, какую систему выбрать потребителю, чем они отличаются друг от друга, как их систематизировать. Системы, можно характеризовать по ряду критериев:

- функционалу;
- производителю;
- используемым технологиям.

Каждый потребитель САДЭД желает получить в свое пользование идеальный в своем внешнем и внутреннем содержании продукт. Система должна удовлетворять требованиям не только конкретных операторов, ежедневно работающих с ней на своем рабочем месте, но и системных администраторов, которым придется ее наладить и обслуживать в дальнейшем, а также бухгалтеров и управленческий персонал, одним словом тех, кто принимает решения на покупку и внедрение системы [1].

При решении широкого спектра задач предприятия каждый желает, чтобы система настраивалась «под себя» и в тоже время она должна быть «коробочным» продуктом, который будет внедряться, и поддерживаться силами коллектива. Конечно, можно получить идеальное настраиваемое решение, которое бы учитывало все бизнес-процессы предприятия, но внедрение таких систем, как правило, занимает от нескольких месяцев до года и даже более и требует немалых финансовых затрат. Но, как известно сроки внедрения должны быть минимальным, а обучение персонала быстрым, «безболезненным» и желательно без отрыва от производственной деятельности.

Система должна поддерживать как бумажный, так и электронный документооборот.

Особое требование предъявляется к принципам реализации защиты данных в системе, то есть, как организована система прав, интегрируется ли с антивирусными пакетами и т.д.

Не последнюю роль при выборе занимают такие вещи как система управления базами данных (СУБД), используемая системой, и средство разработки. Чем распространенней СУБД, тем проще с ней работать и тем

независимей пользователь от разработчика системы. На многих современных СУБД могут реализовываться несколько программных продуктов, а это еще один плюс в пользу их выбора. Также уделяется внимание средству разработки программы. Чем известнее и распространеннее средство разработки, тем больше специалистов, которые им владеют и смогут разрабатывать дополнительные компоненты для СЭД, опять же, тем меньше стоимость их услуг, т.к. они не являются «эксклюзивом» [2].

С одной стороны, продукт должен быть массовым, чтобы пользователь мог сам осуществлять некоторые настройки системы (например, такие как словари, справочники и т.д.), не прибегая лишней раз к услугам системного администратора. Администратор должен осуществлять только функции по обеспечению безопасности информации, контролируя права и действия пользователей.

В настоящий момент уже недостаточно предоставлять продукт с функционалом больше, чем у конкурента и лучше ориентированным на решение конкретных задач. Так устроен человек, что, прежде всего, он обращает внимание на обертку, а уж потом на содержание. В нашем случае оберткой является интерфейс программного продукта, и чем он привлекательнее (современнее, ярче) и доступнее в понимании, тем большим спросом пользуется.

Потребитель внимательно подходит и к выбору разработчика САДЭД. Оценивает, сколько он уже существует на рынке автоматизированных систем, как себя зарекомендовал среди пользователей и конкурентов, каким образом осуществляет техническую поддержку продуктов, как часто выпускает обновление системы и многое другое.

Таким образом, популярность той или иной автоматизированной системы зависит от дружелюбности ее интерфейса, простоты в установке и эксплуатации, легкости в освоении, надежности, актуальности времени, а также стоимости ее внедрения и поддержки. В погоне за рекламой и предлагаемыми супер возможностями покупаемого продукта компании нужно ясно представлять круг решаемых с её помощью задач, каким образом они будут реализовываться и реальное соотношение цена-качество. Конечно, развитые функциональные возможности с учетом современных требований, безусловно, плюс [3].

Служба предприятия периодически (ежеквартально, ежегодно) составляет сводки данных об объеме продаж автозапчастей. Такие сведения позволяют установить определенные закономерности роста и состояния продаж. Установлено, что плотность продаж автозапчастей существенно меняется в пределах года, квартала, месяца, недели. Таким образом, процесс продаж автозапчастей характеризуется различными параметрами, связанными с процессами документирования и со всей деятельностью предприятия.

1.2 Выбранные программные и инструментальные средства и их обоснование

Предъявляемое к системе требование хранения и обработки большого объема данных ставит задачу выбора СУБД или СУФ. Необходимо выбрать систему, которая наиболее полно подходит для решения поставленных в работе задач. Как правило, работа с такими средами как Oracle, MS SQL, My SQL, Informix и т.п. предполагает их установку на целевую систему, что в данном проекте и было сделано.

1.2.1 СУБД MS SQL

СУБД MS SQL является наиболее подходящим вариантом при реализации программного продукта автоматизированной системы контроля и учета автозапчастей.

MS SQL Server — это система анализа и управления реляционными базами данных в решениях электронной коммерции, производственных отраслей и хранилищ данных.

В MS SQL Server 2008 еще более улучшены критически важные возможности, представляемые в более ранней версии, за счет предоставления рекордной производительности, доступности и удобства управления для ваших критически важных приложений. SQL Server 2008 представляет новые функциональные средства работы в оперативной памяти, встроенные в основную базу данных для операций OLTP и хранения данных, которые дополняют существующие технологии хранилищ данных в оперативной памяти и функциональные возможности бизнес-аналитики для создания наиболее разностороннего решения по эксплуатации средств баз данных в оперативной памяти на рынке.

SQL Server 2008 также предоставляет новые решения для аварийного восстановления, резервного копирования и гибридной архитектуры в Windows Azure, позволяя клиентам использовать существующие навыки работы с локальными функциями, опирающимися на возможности глобальных центров обработки данных Microsoft. Кроме того, в SQL Server 2008 используются новые возможности Windows Server 2005, дающие несравнимую масштабируемость для приложений баз данных в физических и виртуальных средах.

Совершенствуясь от версии к версии, MS SQL стал инструментом, который может удовлетворить потребности самых разных категорий пользователей: от новичка, которому нравится дружественный интерфейс системы, позволяющий ему справиться с задачами, до профессионального разработчика, который имеет весь необходимый инструментарий для построения уникального решения для конкретного предприятия среднего бизнеса.

1.2.2 Borland Delphi 7

В качестве среды разработки программного продукта была выбрана среда Borland Delphi 7.0.

Delphi — это среда быстрой разработки, в которой в качестве языка программирования используется язык Delphi. Язык Delphi — строго типизированный объектно-ориентированный язык, в основе которого лежит хорошо знакомый программистам Object Pascal.

Borland Delphi 7 позволяет создавать самые различные программы: от простейших однооконных приложений до программ управления распределенными базами. В состав пакета включены разнообразные утилиты, обеспечивающие работу с базами данных, XML-документами, создание справочной системы, решение других задач.

Borland Delphi 7 может работать в среде операционных систем от Windows 98 до Windows 7/Vista. Особых требований, по современным меркам, к ресурсам компьютера пакет не предъявляет: процессор должен быть типа Pentium или Celeron с тактовой частотой не ниже 166 МГц (рекомендуется Pentium II 400 МГц), оперативной памяти - 128 Мбайт (рекомендуется 256 Мбайт), достаточное количество свободного дискового пространства.

Delphi 7 содержит развитые библиотеки и инструменты для создания приложений, полностью интегрирует соответствующие технологии и качественно повышает производительность разработчиков. Интегрируя ведущие приложения разработки в единый и легкий в использовании пакет, Delphi 7 сокращает жизненный цикл разработки приложений и ускоряет вывод создаваемых с его помощью продуктов на рынок ПО [7].

Таким образом, с учетом требований для разработки системы контроля и учета автозапчастей используются следующие среды:

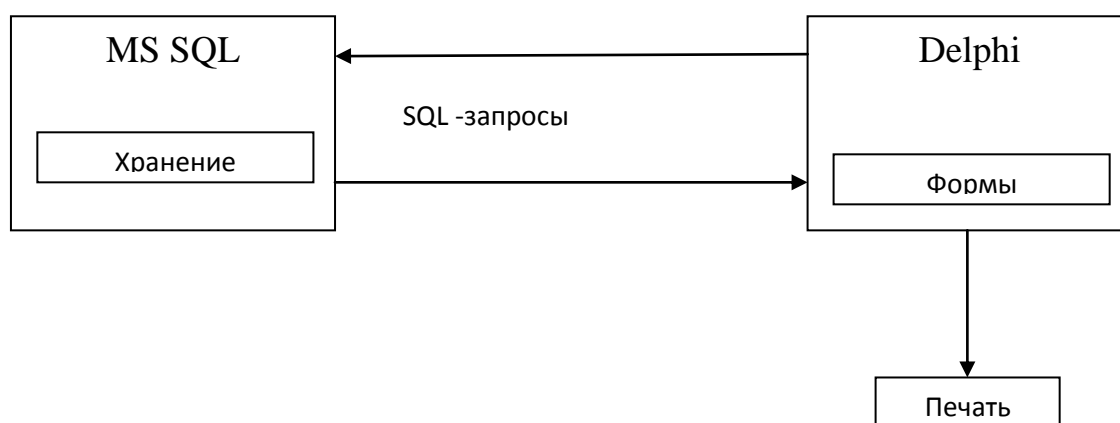


Рисунок 1.1 – Взаимодействия пользовательского приложения

- MS Access - для хранения данных;
- Borland Delphi 7 - для организации пользовательского интерфейса;

Для резервного копирования и восстановления архивных данных используется Microsoft Word.

Особенности взаимодействия используемых сред отражены на рисунке 1.1. На основании спроектированной структурной схемы системы контроля и учета автозапчастей получен алгоритм работы приложения, приведенный на рисунке 1.2.

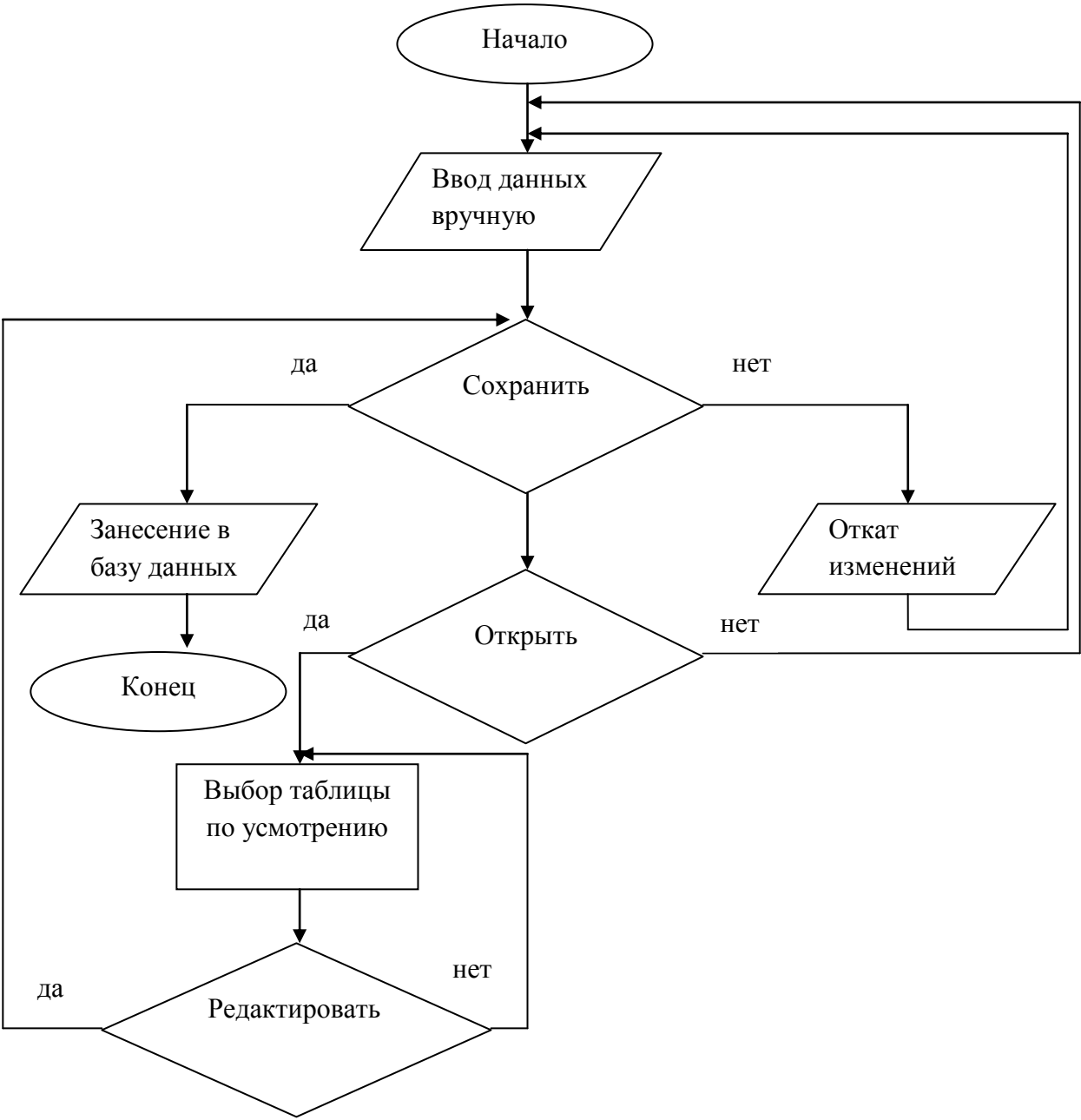


Рисунок 1.2 – Алгоритм функционирования приложения

1.3 Анализ существующих на рынке БД «Автозапчасти»

База данных, разрабатываемая в рамках дипломного проекта, занимается автоматизацией систем контроля автозапчастей и учетом клиентов.

Перед разработкой базы данных «Автозапчасти» проводился тщательный анализ существующих на рынке баз данных и были выявлены недостатки. Ниже приведены недостатки аналогичных баз данных и предложены пути решения.

1.3.1 БД «АвтоКаталог»

«АвтоКаталог» представляет собой электронную версию "бумажных" каталогов запасных частей по отечественным и иностранным автомобилям и двигателям. По сути, это электронный каталог запчастей, компьютерный справочник (база данных) с информацией об устройстве автомобилей - от крупных узлов и агрегатов до запчастей с их кодами (каталожными номерами), наименованиями и графическими изображениями (чертежами). АвтоКаталог обладает присущей компьютерным программам компактностью (всего 6 компакт-дисков, поставка может быть и на DVD), высокой скоростью поиска информации, широкими возможностями работы с ней (масштабирование, печать), потрясающим удобством и наглядностью в работе. Интерфейс программы можно увидеть на рисунке 1.3.

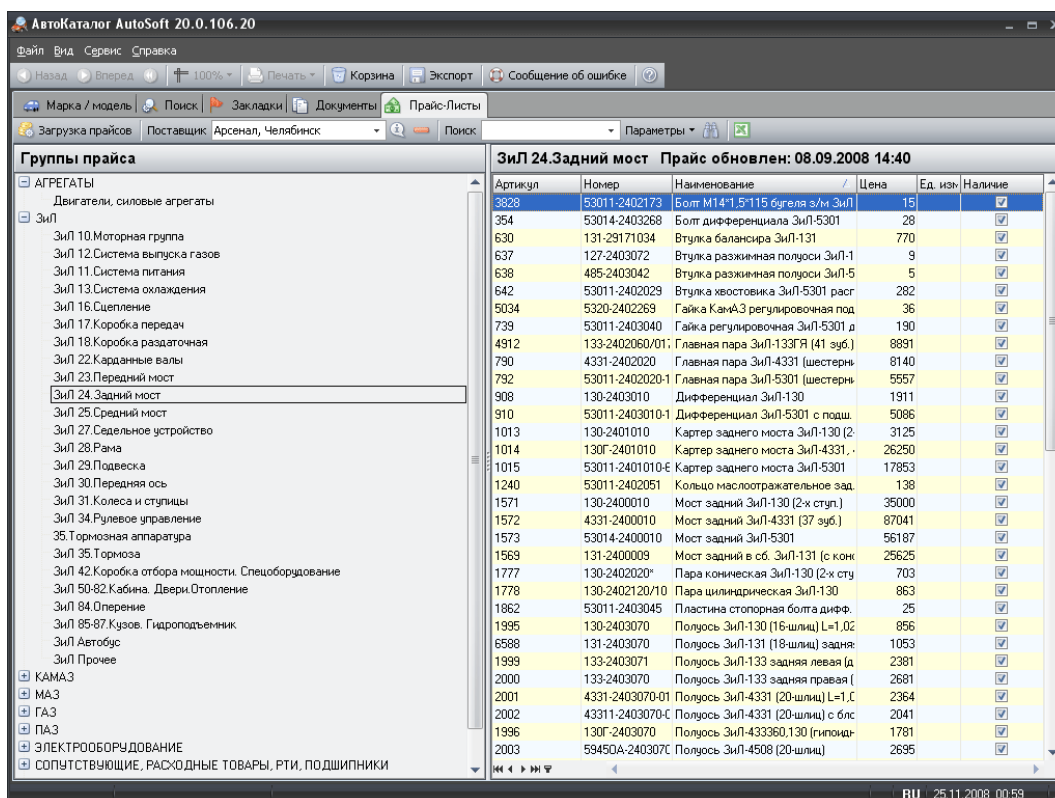


Рисунок 1.3 – Интерфейс программы «АвтоКаталог»

К большому недостатку программы можно отнести отсутствие учета и регистрации клиентов. Это один из главных атрибутов функций продажи автозапчастей. Перед нами большая база знаний без связи «клиент - продажа - автозапчасть».

1.3.2 БД «АвтоДилер»

Программа «АвтоДилер» занимается учетом в автосервисе и в автомагазине, имеет каталоги запчастей, нормы времени ремонта.

«АвтоДилер» - это специализированное программное обеспечение для автобизнеса. Система предназначена для автоматизации учета, планирования и анализа работы любых предприятий: крупных и мелких автомастерских, автосалонов, магазинов автозапчастей, автомоек, шиномонтажных мастерских и станций замены масла, автостраховщиков.

При открытии программы тут же выскакивает рекламное окно, которое представлено на рисунке 1.4.

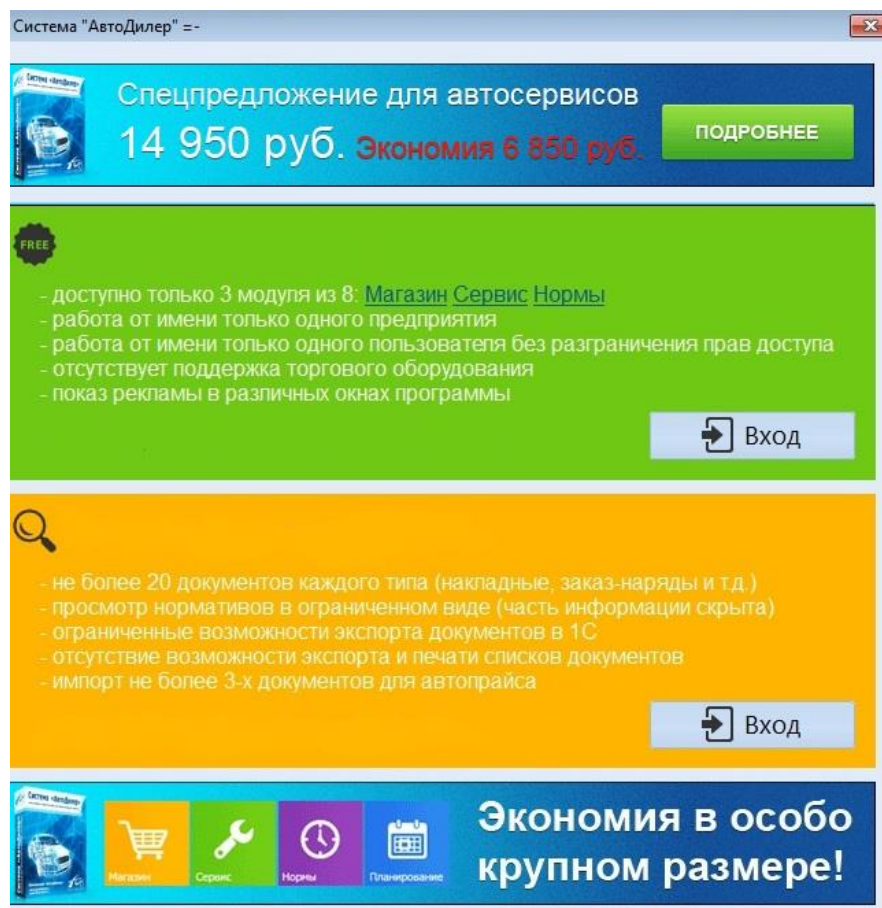


Рисунок 1.4 – Реклама при входе в систему «АвтоДилер»

Также имеется большой недостаток в отсутствии логистики. База данных имеет много таблиц и вкладок, что не позволяет пользователю с любым уровнем подготовки выполнять необходимые задачи (Рисунок 1.5).

Хорошая программа должна иметь понятный интерфейс и логику, а не коммерческую универсальность. То есть, еще одним недостатком системы «АвтоДилер» является то, что он не разработан под определенное предприятие, которое представлено на рисунке 1.5.

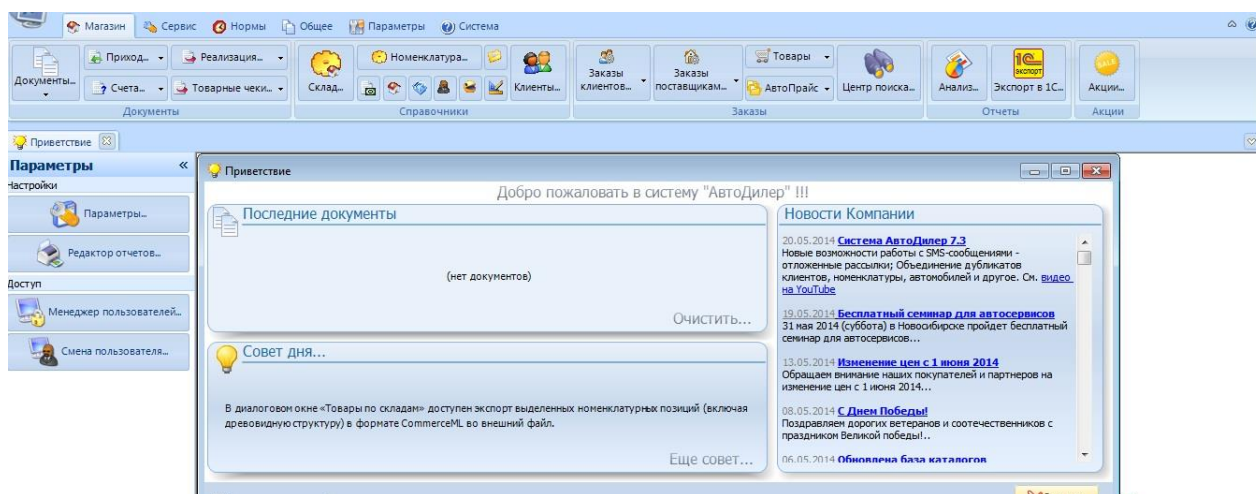


Рисунок 1.5 – Интерфейс программы «АвтоДилер»

При выходе из программы еще раз появляется реклама, которая показана на рисунке 1.6.

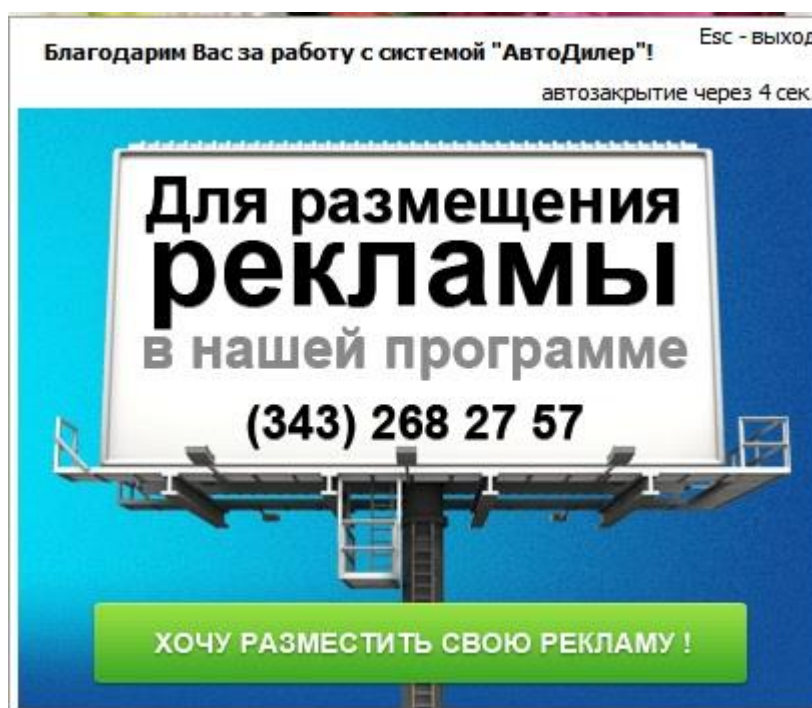


Рисунок 1.6 – Назойливая реклама при выходе из программы

В данном дипломном проекте были учтены ошибки аналогов базы данных «Автозапчасти» и устранены.

2 Проектирование и разработка базы данных «Автозапчасти»

2.1 Требования к системе

- а) обеспечение разработки системы на основе установленных на целевом компьютере ПО;
- б) простой и удобный пользовательский интерфейс;
- в) обеспечение обработки и хранения большого объема данных;
- г) формирование отчетности;
- д) возможность резервного копирования данных;
- е) переносимость системы.

2.2 Архитектура системы

Архитектура разрабатываемой программной системы представлена на рисунке 2.1.

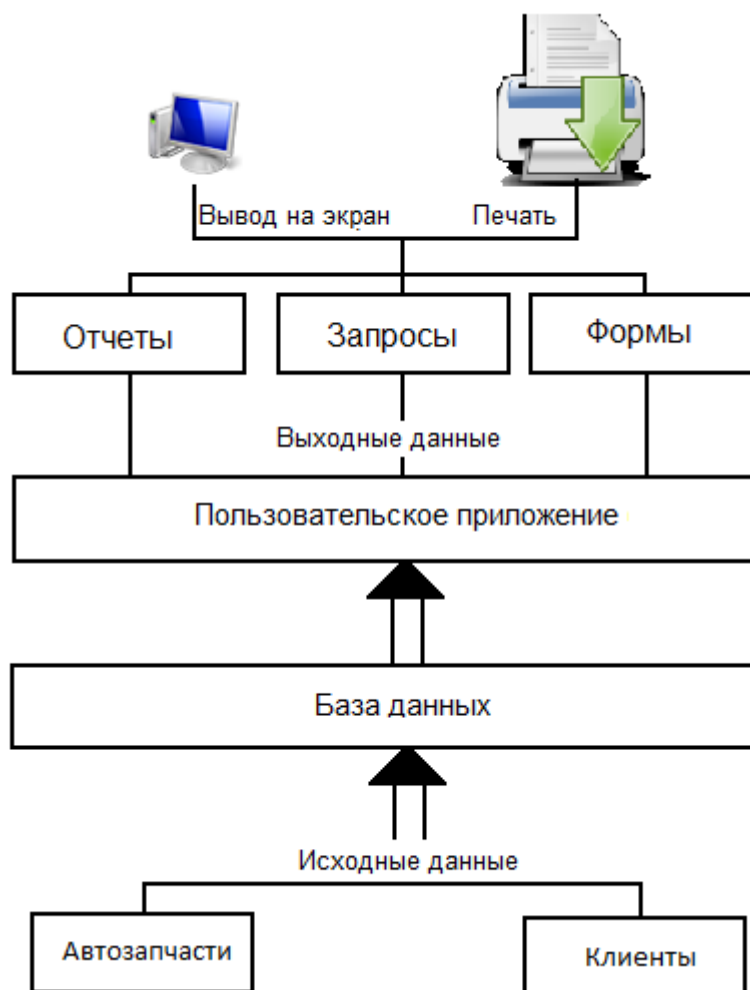


Рисунок 2.1 - Архитектура автоматизированной системы контроля автозапчастей и учета клиентов

2.3 Разработка UML-диаграмм БД «Автозапчасти»

2.3.1 Диаграмма прецедентов

На диаграмме прецедентов (вариантов использования) показано взаимодействие между вариантами использования и действующими лицами.

Она отражает требования к системе с точки зрения пользователя. Таким образом, варианты использования – это функции, выполняемые системой, а действующие лица – это заинтересованные по отношению к создаваемой системе [4].

Основная задача диаграммы вариантов использования - представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.



Рисунок 2.2 – Диаграмма прецедентов

Таблица 2.1 - Распределение требований по субъектам и прецедентам

	Описание требования	Субъект	Прецедент
1	Клиент должен иметь возможность оформить заказ на оказание услуг.	Клиент	Оформление заказа
2	Клиент должен быть зарегистрирован в системе, чтобы оформить заказ.	Клиент	Регистрация клиентов
3	Персонал автозапчасти должен получить заказ для его выполнения.	Персонал автозапчасти	Оформление заказа
4	Клиент должен иметь возможность посмотреть список доступных услуг.	Клиент	Информация об услугах

Продолжение таблицы 2.1

5	Клиент должен иметь возможность получить информацию по состоянию его заказа.	Клиент	Информация о заказе
6	Клиент должен получить окончательный счет за оказание услуг в автозапчасти с отчетом о покупке в печатном виде.	Клиент	Конец обслуживания клиента
7	Персонал автозапчасти должен иметь возможность ввести данные о выполненном заказе (номера услуг, стоимость и т.д.) для формирования окончательного счета.	Персонал автосервиса	Конец обслуживания клиента

2.3.2 Вид с точки зрения процесса

Диаграммы видов деятельности - это один из пяти видов диаграмм, применяемых в UML для моделирования динамических аспектов поведения системы. Диаграмма видов деятельности - это, по существу, блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой.

Диаграммы деятельности можно использовать для моделирования динамических аспектов поведения системы. Как правило, они применяются, чтобы промоделировать последовательные (а иногда и параллельные) шаги вычислительного процесса [5].

Основными элементами диаграмм видов деятельности являются обозначения состояния («начало», «конец»), действия (овал) и момента синхронизации действий (линейка синхронизации, на которой сходятся или разветвляются несколько стрелок).

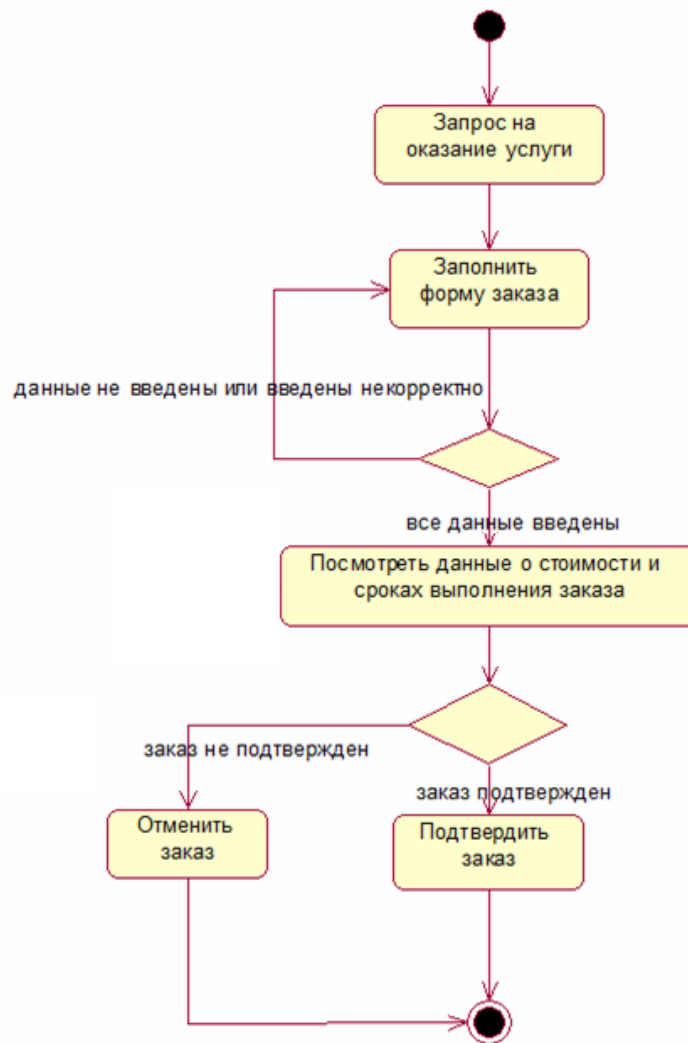


Рисунок 2.3 - Диаграмма видов деятельности для прецедента «Оформление заказа»

2.3.3 Вид с точки зрения проектирования

Диаграмма последовательности действий призвана наглядно отобразить набор процессов, их последовательность и взаимодействие по времени их появления. Например, когда нужно проработать буквально по шагам какой-то важный участок выполнения программы.

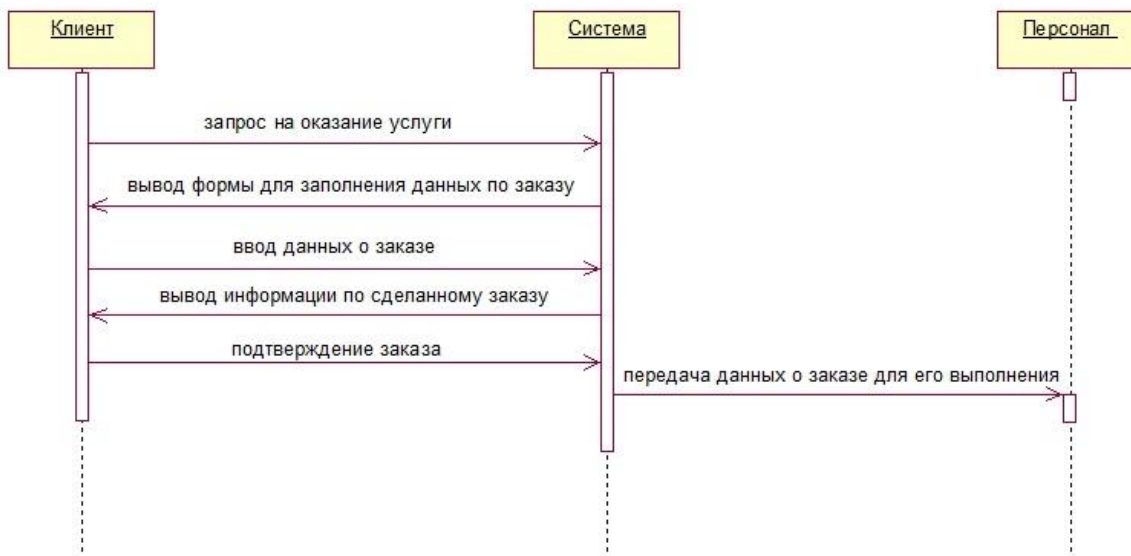


Рисунок 2.3 - Диаграмма последовательности прецедента «Оформление заказа»

Рассмотрим каждый элемент диаграммы, по отдельности:

Объект, Участник (Object, Participant). Обозначается прямоугольником, в котором показывается информация об участнике действий. Размещаются объекты (как правило) вдоль верхнего края диаграммы. От прямоугольника вниз опускается Линия Жизни.

Линия жизни (Life Line). Линия, исходящая вниз от участника, означающая отведенное объекту время жизни. Обозначается пунктирной линией.

Активация, фрагмент выполнения (Activation Bar, Execution Occurances). Обозначается узким прямоугольником (серого или белого цвета), размещенным на линии жизни. Показывает начало и завершение действия, в котором участвует объект. Поскольку линия жизни - это метафора времени, то прямоугольник на линии жизни указывает на активизацию объекта во времени.

2.3.4 Вид с точки зрения развертывания

Физическое представление программной системы не может быть полным, если отсутствует информация о том, на какой платформе и на каких вычислительных средствах она реализована. Для представления общей конфигурации и топологии распределенной программной системы в UML предназначены диаграммы размещения.

Диаграмма размещения предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения. При этом представляются только компоненты-экземпляры программы, являющиеся исполняемыми файлами или динамическими библиотеками. Те компоненты, которые не используются на этапе исполнения, на диаграмме развертывания не показываются. Так, компоненты с исходными текстами программ могут присутствовать только на диаграмме компонентов. На диаграмме размещения они не указываются.

Диаграмма размещения отражает физические взаимосвязи между программными и аппаратными компонентами системы. Она является хорошим средством для того, чтобы показать маршруты перемещения объектов и компонентов в распределенной системе. Каждый узел на диаграмме размещения представляет собой некоторый тип вычислительного устройства – в большинстве случаев, часть аппаратуры. Эта аппаратура может быть простым устройством или датчиком, а может быть и мэйн фреймом [3].

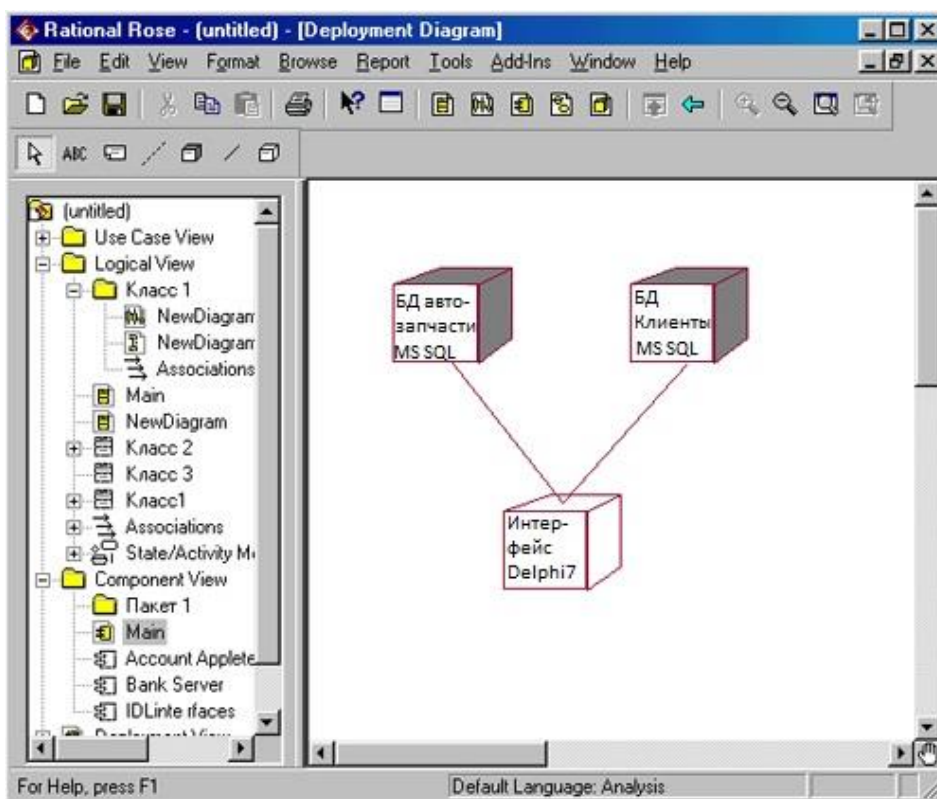


Рисунок 2.5 – Диаграмма развертывания

На данной диаграмме представлены процессоры, то есть те устройства, которые могут обрабатывать данные.

Диаграмма размещения содержит графические изображения устройств и связей между ними. В отличие от диаграмм логического представления, диаграмма размещения является единой для системы в целом, поскольку должна всецело отражать особенности ее реализации. Разработка диаграммы размещения, как правило, является последним этапом спецификации модели программной системы [2].

2.4 Проектирование БД «Автозапчасти»

2.4.1 Построение логической модели БД «Автозапчасти»

Для разработки данной БД «Автозапчасти» было использовано СУБД MS SQL.

В процессе логического проектирования высокоуровневое представление данных преобразуется в структуру используемой СУБД. Основной целью данного этапа является устранение проблем данных с использованием специальных форм нормализации. Цель нормализации – как можно минимизировать повторения данных и возможные изменения БД при процедурах обновления. Это достигается разделением одной таблицы в несколько с последующим использованием при запросах операции навигации ими. Навигационный поиск снижает быстродействие БД, т.е. увеличивает время отклика на его запрос. Полученная логическая структура БД может быть оценена количественно с помощью различных характеристик (число обращений к логическим записям, объем данных в каждом приложении, общий объем данных). На основе этих оценок логическая структура может быть усовершенствована с целью достижения большей эффективности.

Специального обсуждения заслуживает процедура управления БД. Она наиболее проста в однопользовательском режиме. В многопользовательском режиме и в распределенных БД процедура сильно усложняется. При одновременном доступе нескольких пользователей без принятия специальных мер возможно нарушение целостности информации. Для устранения этого явления используют систему транзакций и режим блокировки таблиц или отдельных записей.

Транзакция - процесс изменения файла, записи или базы данных, вызванный передачей одного входного сообщения. Особенности блокирования и варианты блокировки далее будут рассмотрены отдельно.

Логический (концептуальный) уровень построен с учетом специфики и особенностей конкретной СУБД. Этот уровень представления данных ориентирован больше на компьютерную обработку и на программистов, которые занимаются ее разработкой. На этом уровне формируется концептуальная модель данных, то есть специальным способом

структурированная модель предметной области, которая отвечает особенностям и ограничениям выбранной СУБД.

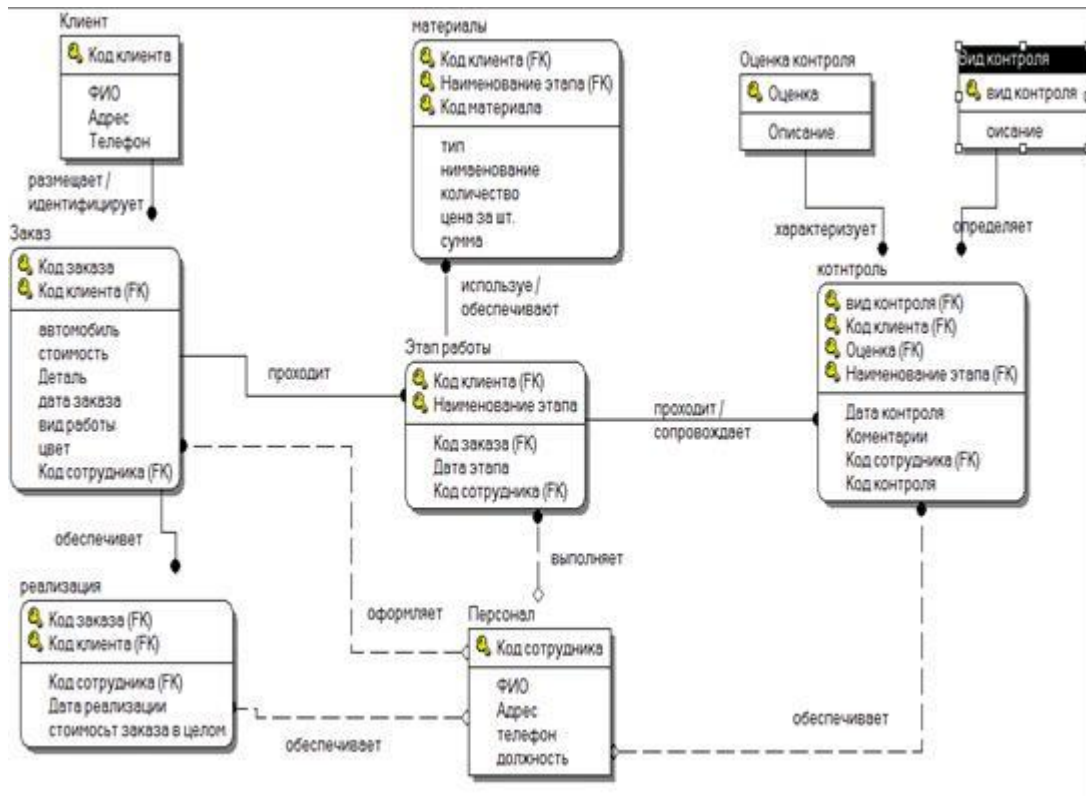


Рисунок 2.6 - Логический уровень

Физическая модель данных зависит от конкретной СУБД, фактически являясь отображением системного каталога. В физической модели содержится информация обо всех объектах БД. Поскольку стандартов на объекты БД не существует (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей. Если в логической модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах - таблицах, колонках, индексах, процедурах и т.д. Разделение модели данных на логические и физические позволяет решить несколько важных задач [8].

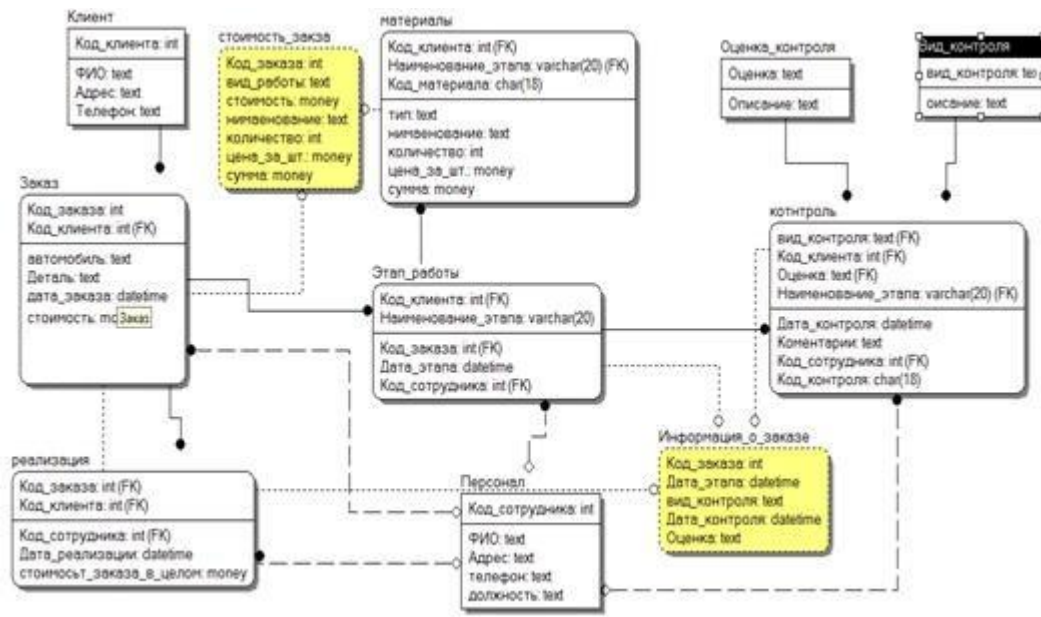


Рисунок 2.7 - Физический уровень

3 Разработка БД «Автозапчасти»

3.1 Пользовательский интерфейс БД «Автозапчасти»

Интерфейс любой системы является одной из очень важной составляющей. Он ориентирован, прежде всего, на конечного пользователя.

Для удобства пользовательский интерфейс программы является интерфейсом со свободной навигацией [9].

При запуске приложения на экран выводится форма авторизации пользователя (см. рисунок 3.1), в которой выбирается подсистема из трех предложенных: 1) клиент; 2) автозапчасти; 3) номенклатура.

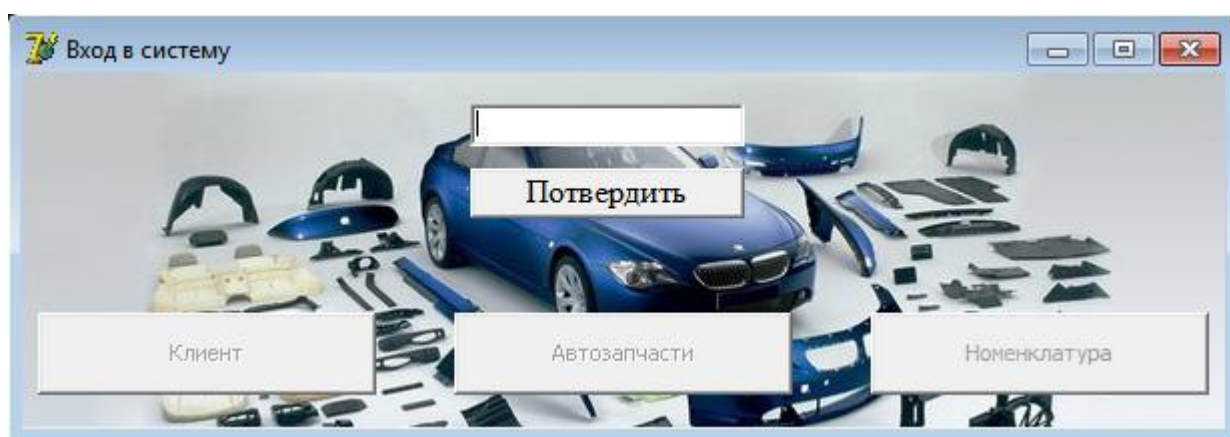


Рисунок 3.1 - Форма авторизации пользовательского интерфейса

После выбора подсистемы «Клиент», приводится основная форма пользовательского интерфейса (см. рисунок 3.2).

В основной форме предлагается возможность заполнения таблицы.

На панели быстрого запуска приведены следующие функциональные возможности:

а) добавить – добавляет новых клиентов в базу для последующего редактирования и сохранения;

б) редактировать – поправляет данные о клиентах, позволяет ввести уточнения;

в) удалить – удаляет клиента из базы;

г) приход на склад – отображение списка и регистрация прихода на склад автозапчастей;

д) поиск – быстрый поиск нужного клиента;

е) панель с дополнительной информацией – отображает дополнительную информацию, если таковая имеется;

ж) шрифт – позволяет менять шрифт;

з) печать – копирование данных из БД в документ Word;

и) выход – выход из программы.

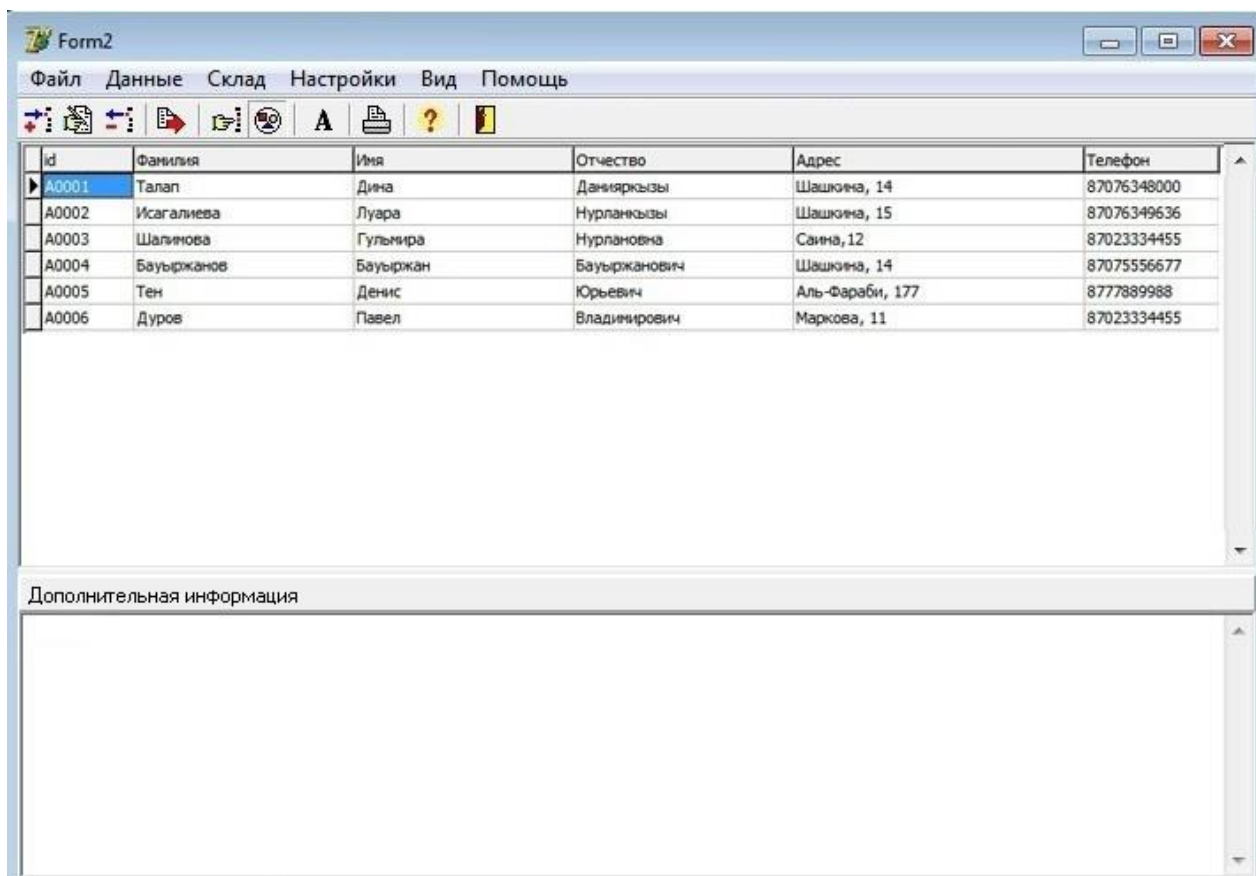


Рисунок 3.2 - Основная форма пользовательского интерфейса в подсистеме «Клиент»

В верхней части интерфейса расположено меню. Меню содержит:

- файл – операции с выбранным файлом (выход из файла);
- данные – операции с данными (редактировать, удалить, добавить);
- склад – данные о запчастях на складе;
- настройки – настройка шрифта;
- вид – добавляет и убирает из интерфейса программы такие функции как «панель поиска», «дополнительная информация»;
- помощь – предоставляет информацию о программе.

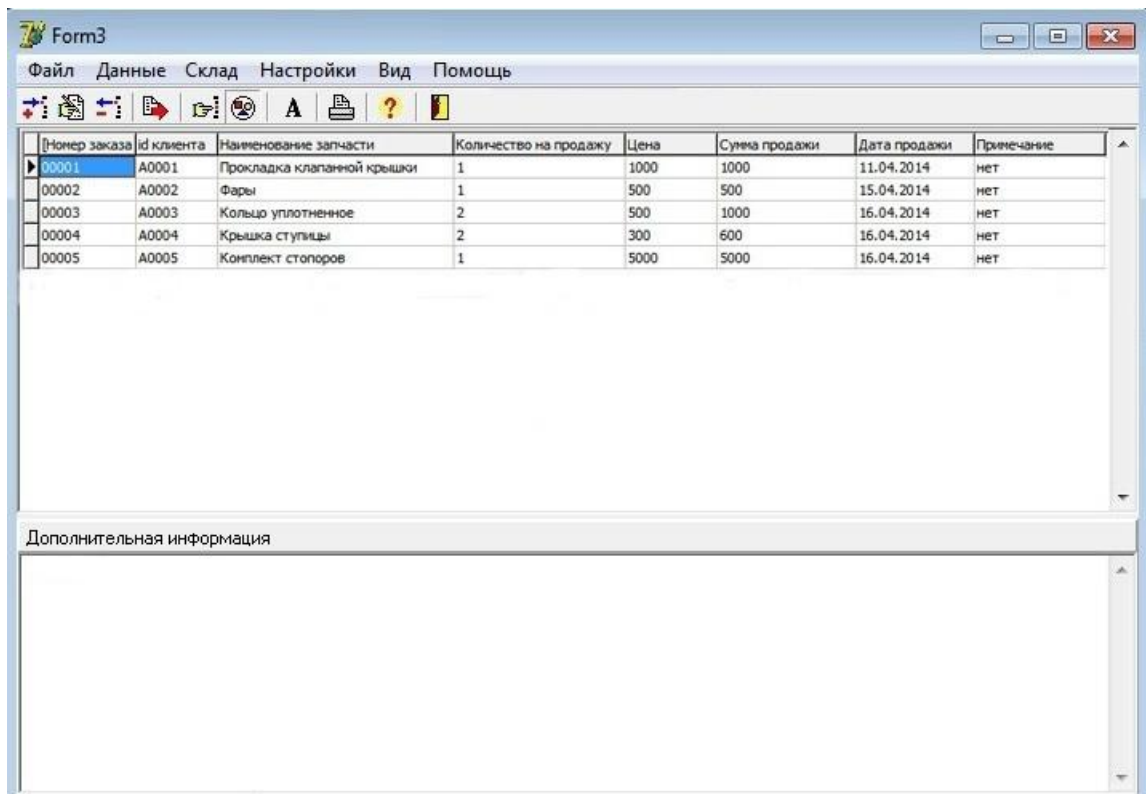


Рисунок 3.3 - Основная форма пользовательского интерфейса в подсистеме «Автозапчасти»

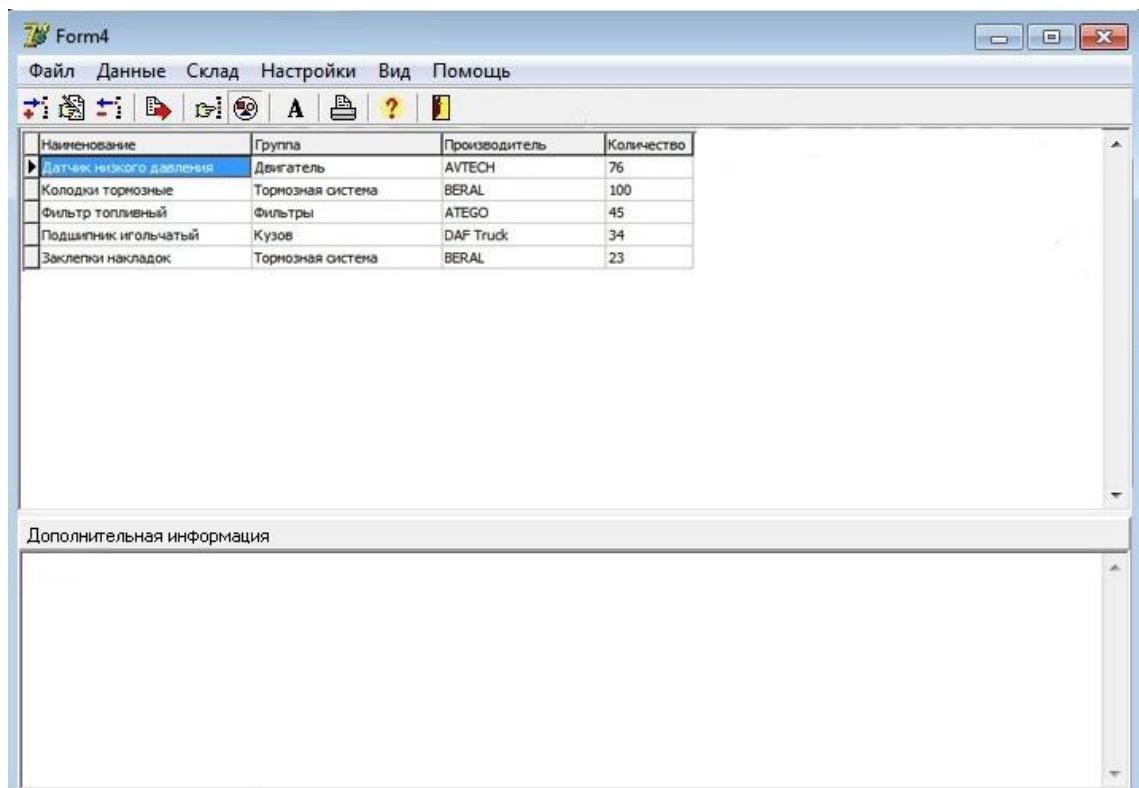


Рисунок 3.4 - Основная форма пользовательского интерфейса в подсистеме «Номенклатура»

3.2 Описание функций БД «Автозапчасти»

Во всех трех подсистемах БД «Автозапчасти» используются одинаковые функции. Рассмотрим подсистему «Клиент».

БД должна иметь возможность добавлять новых клиентов, так как клиенты являются ключевым моментом в продажах автозапчастей. Функция «Добавить» отмечена на рисунке 3.5.

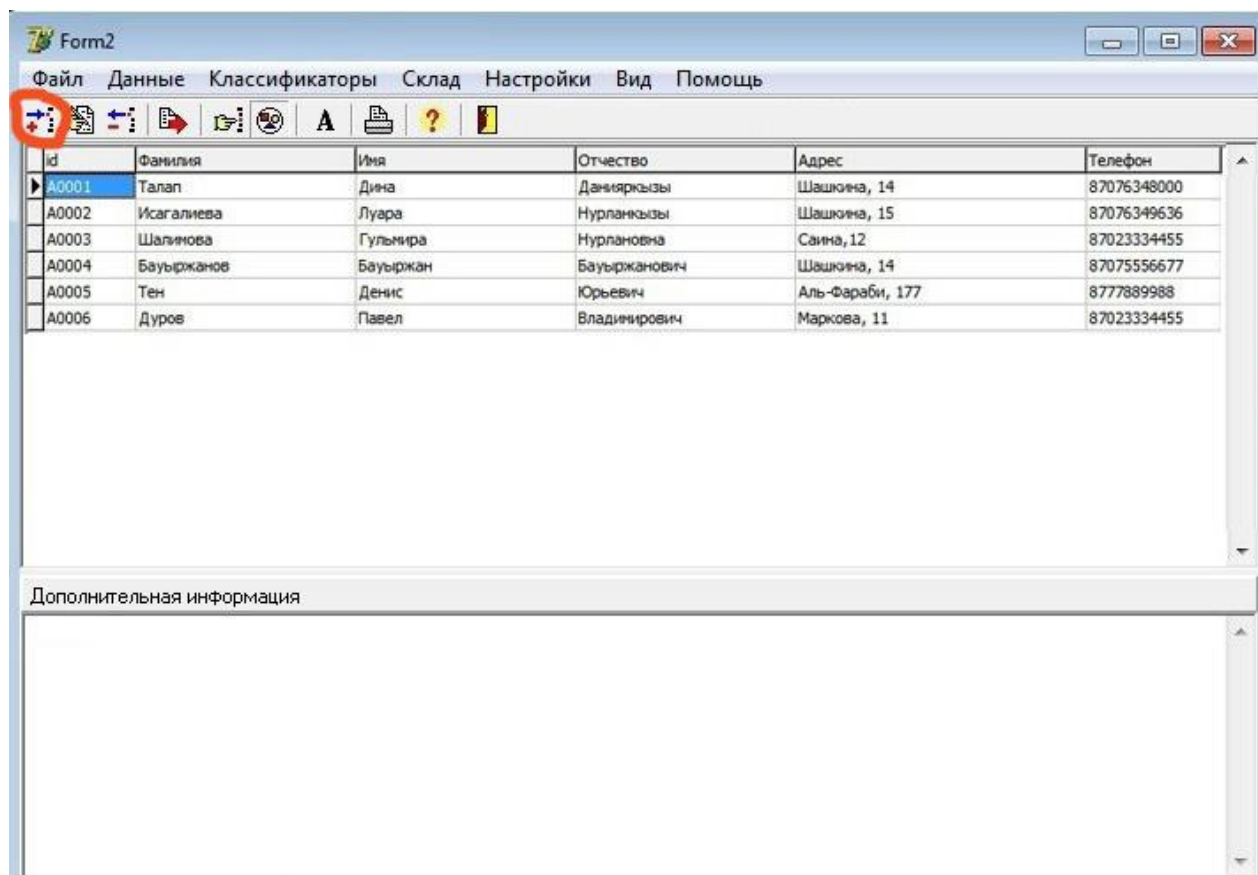


Рисунок 3.5 – Показано, где находится функция «Добавить»

После того, как мы нажали на эту функцию, всплывает форма для заполнения данных о клиенте (см. рисунок 3.6).

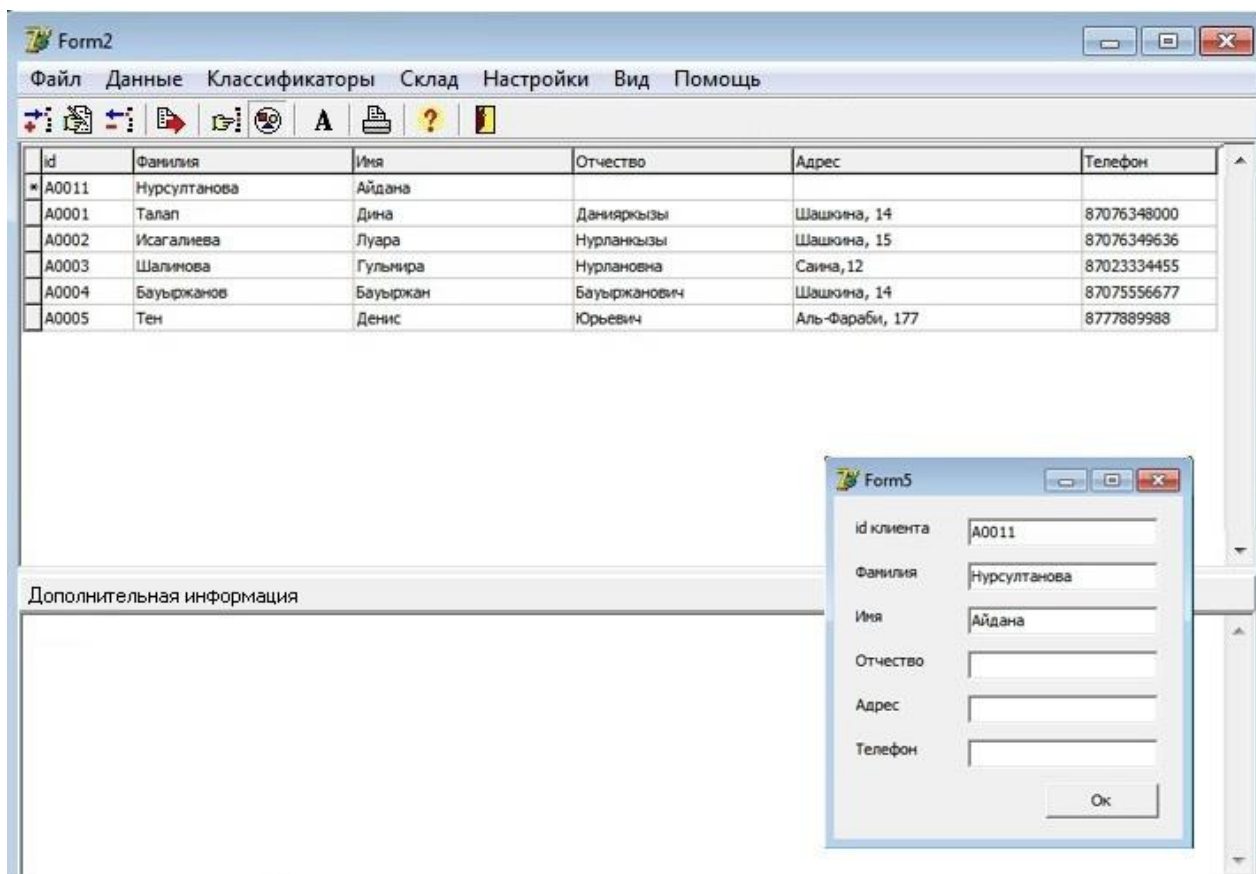


Рисунок 3.6 – Добавление нового клиента в базу

После того, как клиент добавлен в базу, нам необходима функция обновления информации о клиенте. Для этого мы нажимаем на кнопку «Редактировать» на панели быстрого запуска и перед нами всплывает форма для исправления данных (см. рисунок 3.7).

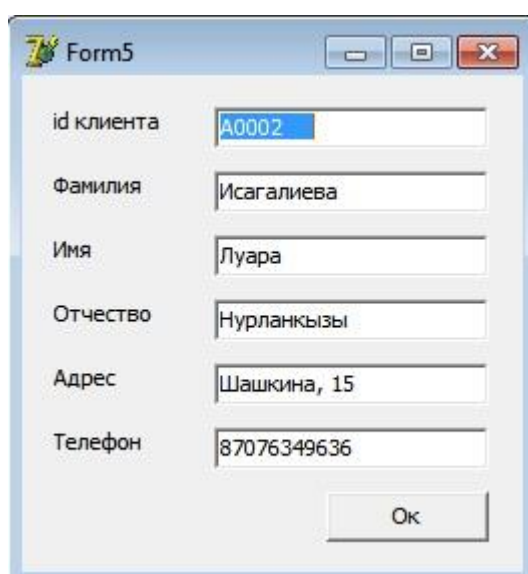


Рисунок 3.7 – Окно редактирования данных

Функция удаления нужна для того, чтоб избавляться от устаревших данных и не засорять память (см. рисунок 3.8)

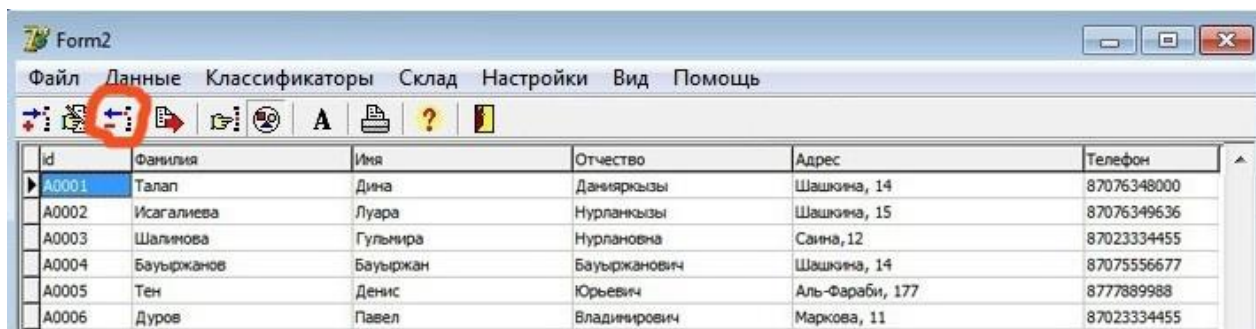


Рисунок 3.8 – Функция удаления

Наводим курсором на клиента, которого нужно удалить и нажимаем кнопку на панели быстрого запуска. После этого клиент удаляется (см. рисунок 3.9).

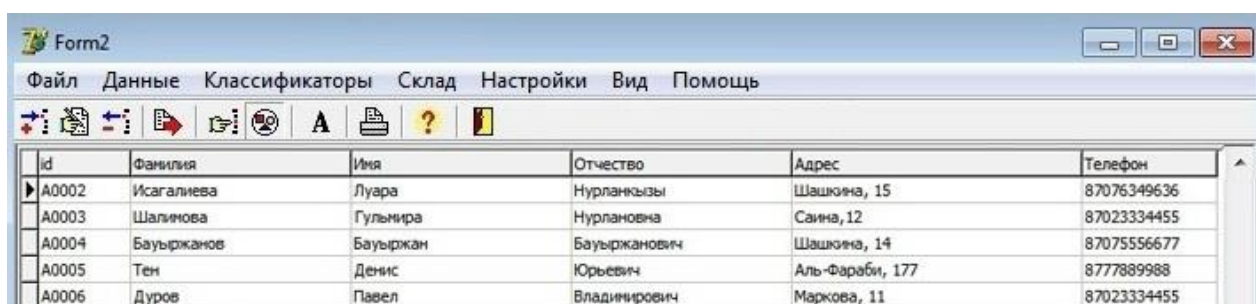


Рисунок 3.9 – Клиент «Талап» был удален

По мере расширения любая БД будет нуждаться в функции «Поиск». Это важный компонент БД, отвечающий за экономию времени и подразумевает нахождение запрашиваемого клиента с предоставлением информации о нем. В нашем случае мы нажимаем на функцию поиска и появляется строка поиска, куда мы вбиваем фамилию (см. рисунок 3.10).

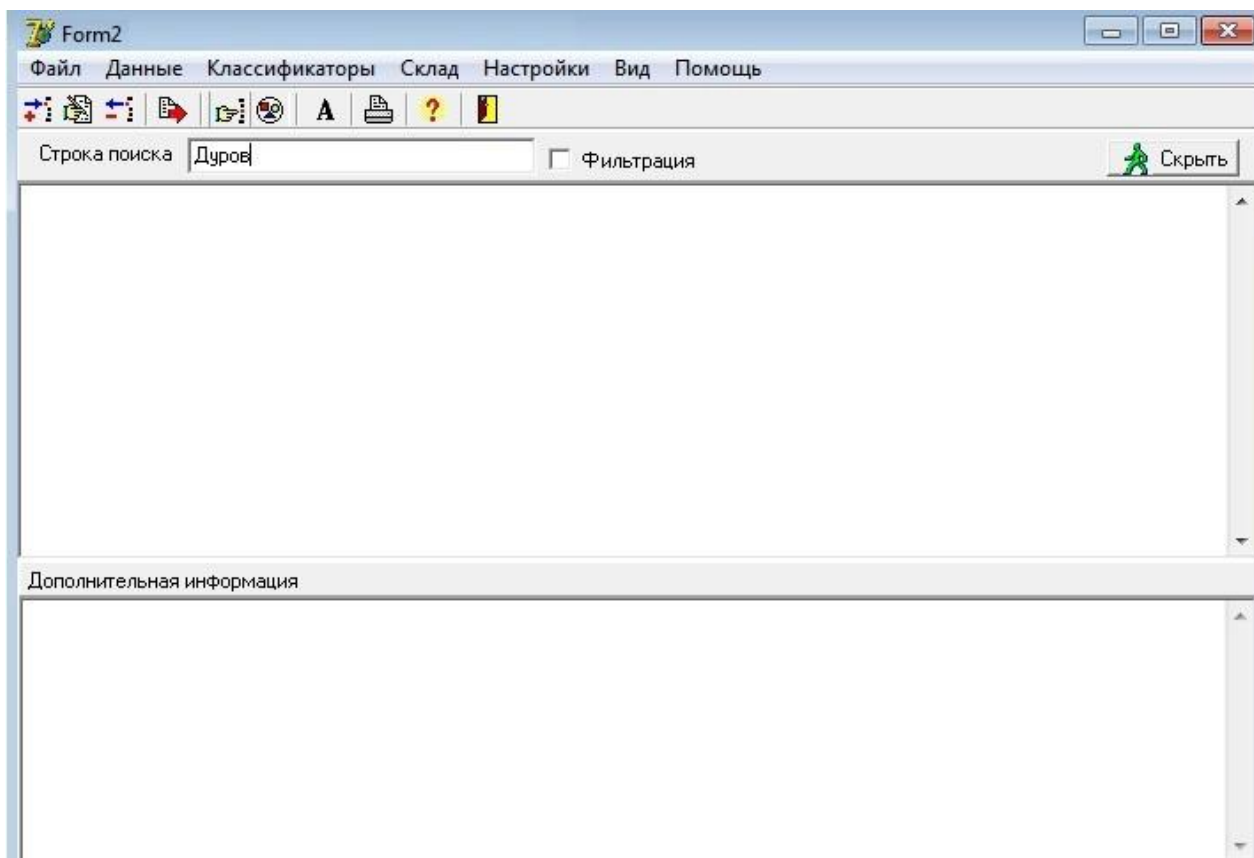


Рисунок 3.10 – Строка поиска

После того, как мы вбили фамилию в строку поиска БД находит этого клиента и указывает на него (см. рисунок 3.11).

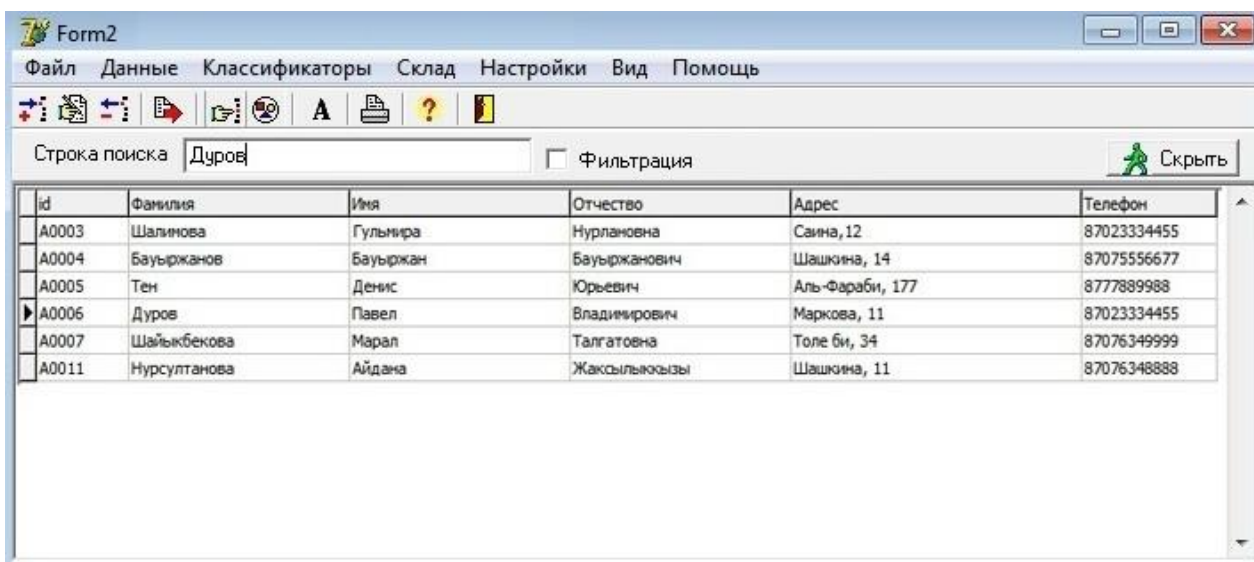


Рисунок 3.11 – Клиент найден в списке

Если нажать на галочку возле подфункции «Фильтрация», то на окне останутся данные только того клиента, которого в данный момент ввели в поисковую строку (см. рисунок 3.12).

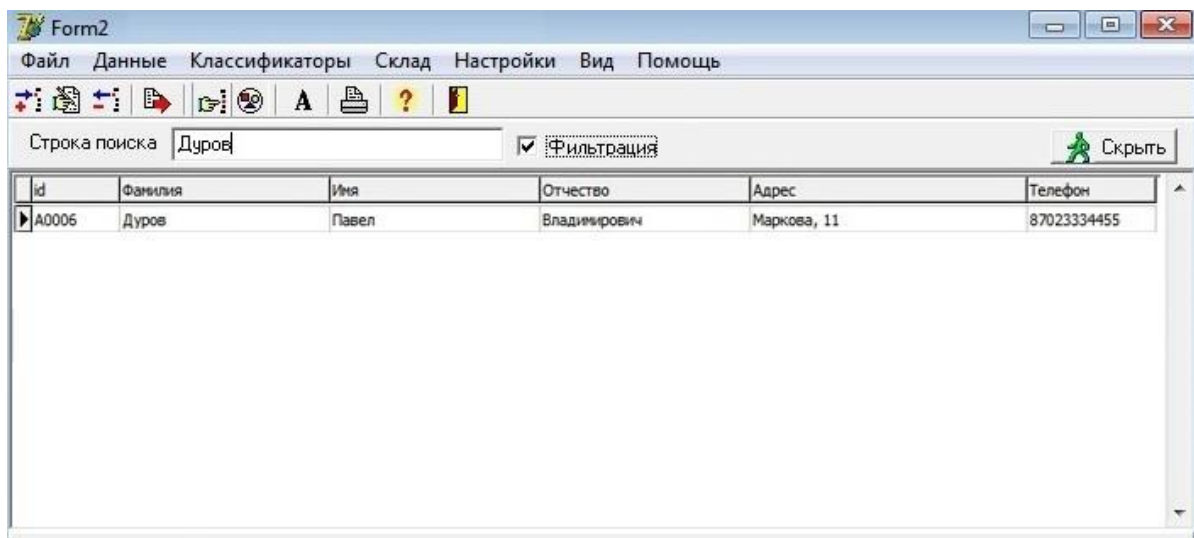


Рисунок 3.12 – Окно при включенной фильтрации

Также есть функция, которая включает и выключает окно дополнительной информации (см. рисунок 3.13).

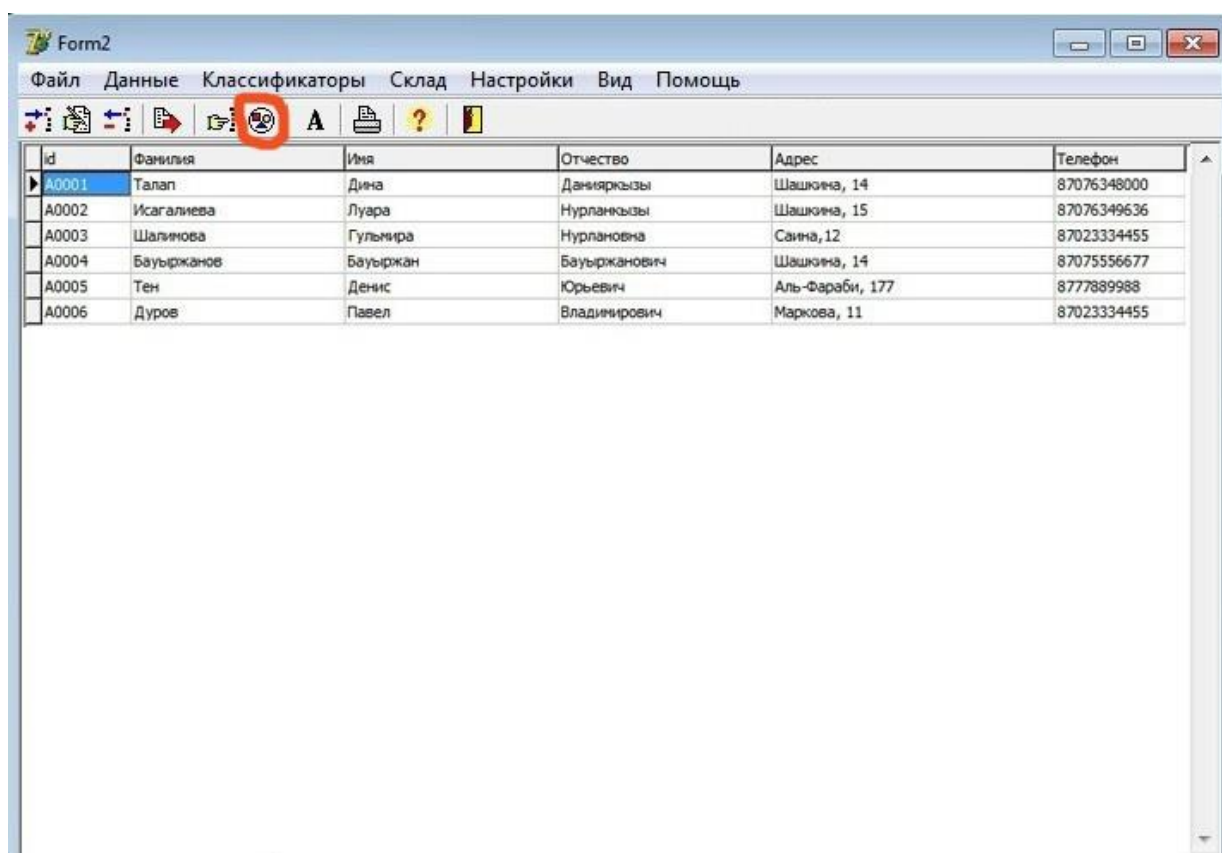


Рисунок 3.13 – Выключение дополнительной информации

Для ведения отчетности необходимо, чтобы данные из БД сохранялись и в форме документов. При нажатии на кнопку «Печать» на панели быстрого запуска, открывается word-документ с таблицей и данными из БД «Автозапчасти».

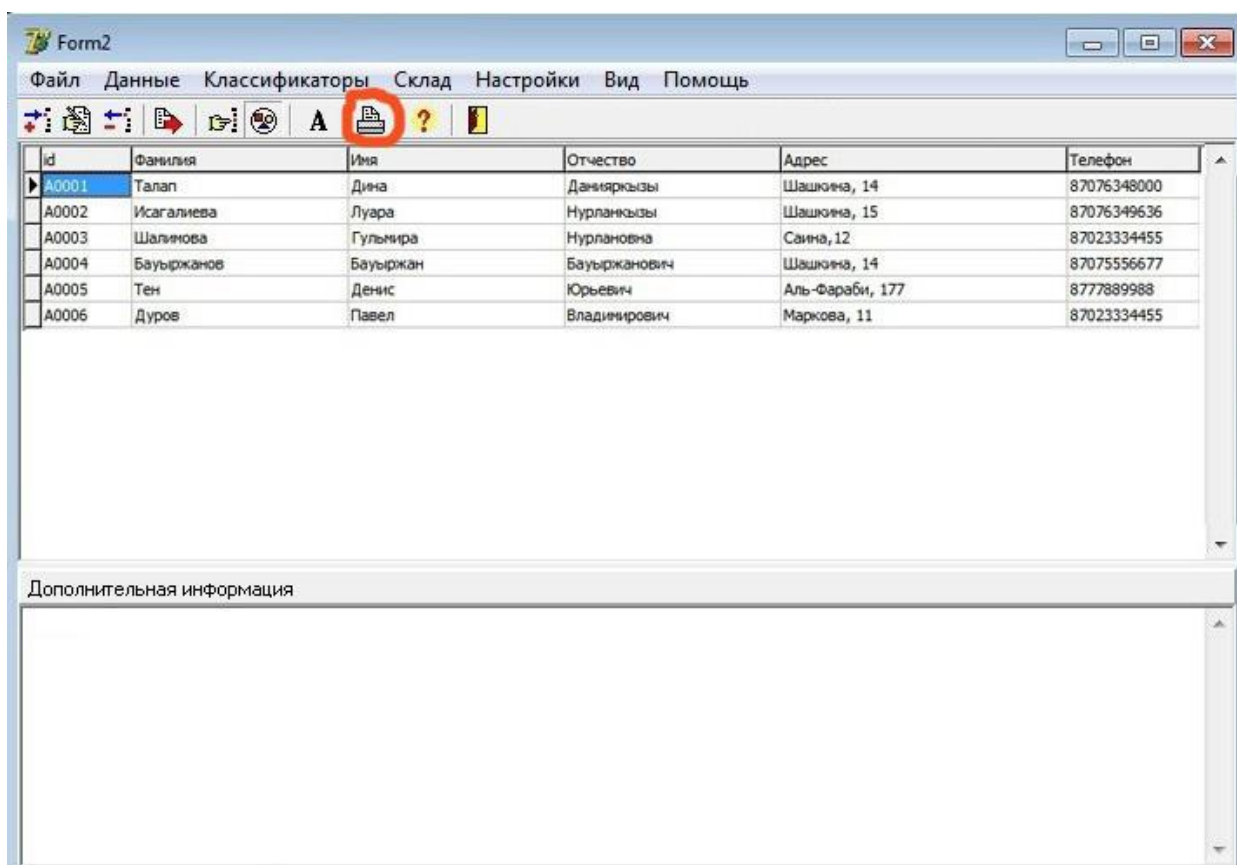


Рисунок 3.14 – Функция «Печать»

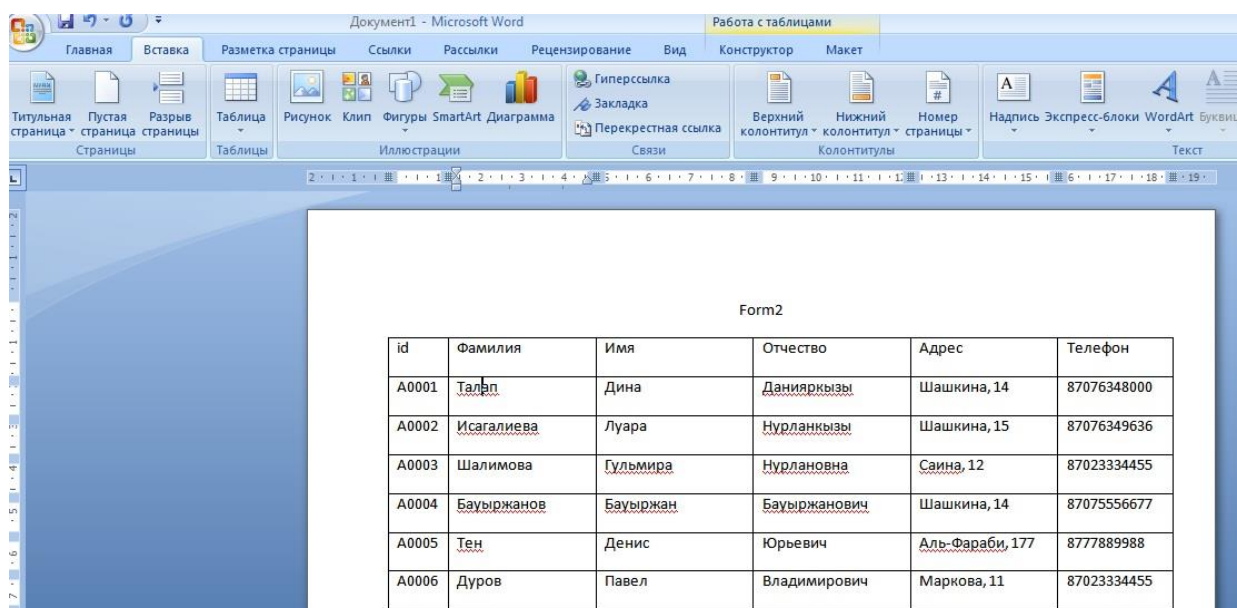


Рисунок 3.15 – Word-документ с данными из БД

В вкладке «Помощь» в меню или в панели быстрого запуска имеется возможность просмотреть информацию о программе (см. рисунок 3.16).

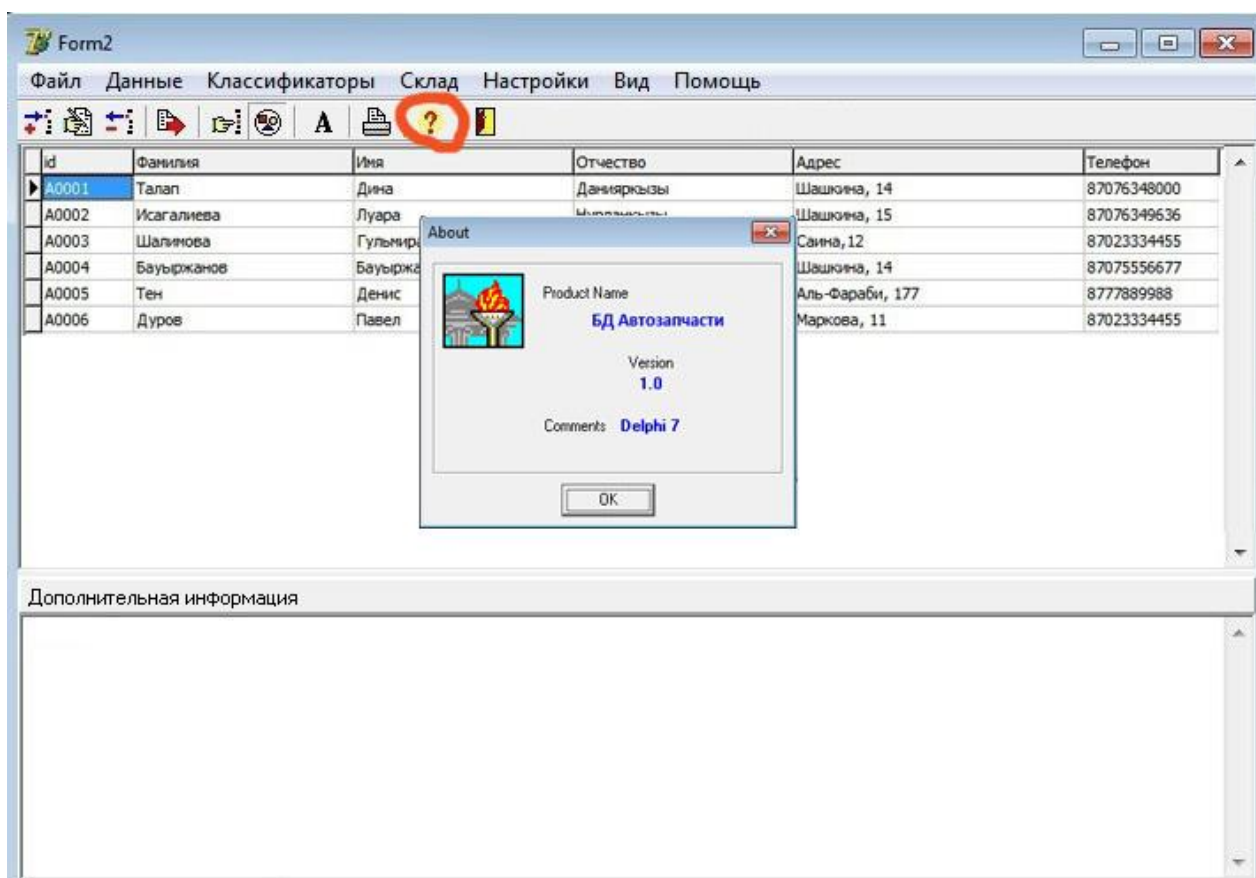


Рисунок 3.16 – Справка о программе

4 Экономическая часть

4.1 Технико-экономическое обоснование

В данном технико-экономическом описании работы приводится расчет экономической эффективности внедрения базы данных «Автозапчасти».

Реализация программной системы обеспечит решение следующих задач:

- сокращение времени, потраченного на составление отчетов и актов купли-продажи автозапчастей;
- сокращение затрачиваемого времени на поиск информации;
- практичный интерфейс приложения.

В данном технико-экономическом описании работы приводятся данные о материальных, трудовых ресурсах, затратах на электроэнергию, потребленных ПК при реализации автоматизированной системы.

В экономической части дипломной работы приводится сравнение расчёта трудоёмкости при старом, ручном, способе подсчёта часов работы сотрудников и расчёта трудоёмкости разработки программного продукта, предназначенного для автоматизации подсчёта часов.

4.2 Расчет трудоемкости разработки ПП

Для определения трудоемкости разработки ПП приведен перечень всех основных этапов и видов работ, которые должны быть выполнены.

Таблица 4.1 – Трудоемкость разработки ПП

Наименование этапов и содержание работ	Трудоемкость разработки ПП, ч.	Исполнитель
Постановка задачи, подготовка исходных данных	8	Руководитель
Выдача задания и исходных данных	6	Руководитель
Ознакомление с заданием и сбор материалов	6	Программист
Обработка исходных данных	8	Программист
Обзор литературы	6	Программист
Разработка плана работ	6	Руководитель
Составление графика проведения работ	3	Руководитель
	3	Программист
Анализ общих вопросов	8	Программист

Изучение методов проведения работ	6	Руководитель
-----------------------------------	---	--------------

Продолжение таблицы 4.1

	4	Программист
Проведение практических работ и получение результатов	12	Программист
Промежуточный анализ результатов	6	Руководитель
	6	Программист
Анализ необходимого программного обеспечения	12	Программист
Разработка программного продукта на ПК	24	Программист
Отладка программного продукта на ПК	6	Программист
Анализ результатов выполненной работы	4	Руководитель
Подготовка документации по программному обеспечению	10	Программист
Установка ПО и обучение пользователя	6	Программист
Итого	150	

Поскольку количество часов активной работы по разработке программного продукта равно 150, а в сутки на разработку выделялось шесть часов, следовательно, срок выполнения проекта равен 25 суткам. Для дальнейших расчетов время разработки программного продукта округляем до одного месяца.

4.3 Расчет затрат на разработку ПП

Общая сумма затрат на материальные ресурсы (Z_M) определяется по формуле:

$$Z_M = \sum P_i * C_i, \quad (4.1)$$

где P_i - расход i -го вида материального ресурса, натуральные единицы;

C_i - цена за единицу i -го вида материального ресурса, тг;

i - вид материального ресурса;

n - количество видов материальных ресурсов [6].

Расчет затрат на материальные ресурсы производится по форме, приведенной в таблице 4.2.

Таблица 4.2 - Затраты на материальные ресурсы

Наименование материала	Единица измерения	Количество	Цена за единицу, тенге	Сумма, тенге
Бумага для принтера	пачка	1	1 000	1 000
Картридж для принтера	шт.	1	5 000	5 000
Диски	шт.	5	50	250
Карандаши	шт.	3	40	120
Ручки	шт.	5	70	350
Мышь A4Tech X7	Шт.	2	2000	2000
Итого				8720

Общая сумма затрат на электроэнергию ($Z_э$) рассчитывается по формуле:

$$Z_э = \sum M_i * K_i * T_i * Ц, \quad (4.2)$$

где M_i - паспортная мощность i -го электрооборудования, кВт;
 K_i - коэффициент использования мощности i -го электрооборудования (принят $K_i=0.7$);
 T_i - время работы i -го оборудования за весь период разработки ПП ч;
 $Ц$ - цена электроэнергии, тг/кВт×ч;
 i - вид электрооборудования;
 n - количество электрооборудования.

Затраты на электроэнергию приведены в таблице 4.3.

Таблица 4.3 - Затраты на электроэнергию

Наименование обор-ия	Паспортная мощность, кВт	Коэфф-т использ-я мощности	Время работы оборуд-я для разработки ПП, ч	Цена электро энергии, $\frac{тг.}{кВт \times ч}$	Сумма, тг
ПК	0,4	0,7	150	20	840
Принтер	0,3	0,5	150	20	450
ИТОГО затраты на электроэнергию					1290

Общая сумма затрат на оплату труда ($Z_{тр}$) определяется по формуле:

$$Z_{\text{тр}} = \sum \text{ЧС}_i * T_i, \quad (4.3)$$

где ЧС_i - часовая ставка i -го работника, тг;

T_i - трудоемкость разработки ПП, чел.×ч;

i - категория работника;

n - количество работников, занятых разработкой ПП.

Часовая ставка инженера-разработчика составляет 1800 (тг/ч),
трудоёмкость разработки – 111 ч.

Часовая ставка научного руководителя составляет 3000 (тг/ч),
трудоёмкость разработки – 39 ч.

$$Z_{\text{тр}} = 1800 * 111 + 3\,000 * 39 = 316\,800 \text{ тг.}$$

Затраты на оплату труда приведены в таблице 4.4.

Таблица 4.4 - Затраты на оплату труда

Категория работника	Квалификация	Трудоёмкость разработки ПП, ч	Часовая ставка, тг/ч	Сумма, тг
1. Программист	Ведущий программист	111	1 800	199 800
2. Руководитель	Руководитель проекта	39	3 000	117 000
ИТОГО затраты на оплату труда				316 800

Сумма годовых амортизационных отчислений определяется по формуле:

$$A = \text{Перв.стоимость} * \text{Норма амортизации} / 100 \quad (4.4)$$

Амортизационные отчисления приведены в таблице 4.5.

Таблица 4.5 - Амортизация основных фондов (ОФ)

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Срок полезного использования оборудования и ПО, год	Сумма амортизации в год, тг	Сумма амортизации в месяц, тг
1. Системный блок Sony	60 000	20	5	12 000	1000
2.Клавиатура Sony	5 000	20	5	1000	84
3.Монитор Sony	25 000	20	5	5000	417
4.Принтер HP LaserJet 1100 Series	20 000	20	5	4000	334
5.Windows 8	30 000	15	2,5	4 500	375
6.Microsoft Office 2010Standard	14 000	15	2,5	2 100	175
7.Borland Delphi7	Распространяется бесплатно				
8.MS SQL Manager Express	Распространяется бесплатно				
ИТОГО амортизация основных фондов					2385

Годовые нормы амортизации ОФ принимаются по налоговому кодексу РК или определяются, исходя из возможного срока полезного использования ОФ:

$$N_{Ai}=100/T_{Ni}, \quad (4.5)$$

где T_{Ni} - возможный срок использования i -го ОФ, год.

$$\begin{aligned} N_{A_{об}} &= 100 / 5 = 20; \\ A_{сб} &= (60\,000 * 20) / 100 = 12\,000 \text{ тг}; \\ A_{кл} &= (5\,000 * 20) / 100 = 1\,000 \text{ тг}; \\ A_{мон} &= (25\,000 * 20) / 100 = 5\,000 \text{ тг}; \\ A_{пр} &= (20\,000 * 20) / 100 = 4\,000 \text{ тг}; \\ A_w &= (30\,000 * 15) / 100 = 4\,500 \text{ тг}; \\ A_{мо} &= (14\,000 * 15) / 100 = 2\,100 \text{ тг}. \end{aligned}$$

Сумма амортизации за один месяц = $A / 12$.

Сумма амортизационных отчислений за два месяца равна 2385 тг.

В статью «Прочие затраты» включаются расходы на арендную плату, включая коммунальные платежи, канцелярские и прочие хозяйственные расходы.

Стоимость аренды помещения на месяц равна 45 000 тг. (в эту сумму включены коммунальные услуги).

Арендная плата рассчитывается по формуле:

$$AP = Ca * S, \quad (4.6)$$

где Ca – срок аренды;

S – стоимость аренды за 1 месяц.

$$AP = 45\,000 * 1 = 45\,000 \text{ тг.}$$

Расходы на интернет, месячная оплата которого составляет 5000 тг равны:

$$P_{и} = 5000 * 1 = 5\,000 \text{ тг.}$$

Прочие хозяйственные расходы составляют 4 000 тг;

Прочие затраты = 45 000 + 5 000 + 4 000 = 54 000 тг.

Социальный налог, согласно Налоговому кодексу РК, составляет 11 % от ФОТ. Пенсионные отчисления не облагаются социальным налогом.

$$O_c = (FOТ - O_{п}) * 0,11, \quad (4.7)$$

где $O_{п}$ - отчисления в пенсионный фонд, 10% от ФОТ.

$$PO = FOТ * 10\% = 316\,800 * 0,1 = 31\,680 \text{ тг.}$$

$$O_c = (316\,800 - 31\,680) * 0,11 = 31\,364 \text{ тг.}$$

На основании полученных данных по отдельным статьям в таблице 4.6 приведена смета затрат на разработку ПП

Таблица 4.6 - Смета затрат на разработку ПП

Статьи затрат	Сумма, тг
1. Материальные затраты, в том числе:	
- материалы	8 720
- электроэнергия	1290
2. Затраты на оплату труда.	316 800
3. Отчисления на социальные нужды.	31 364
4. Амортизация основных фондов.	2 419

5. Прочие затраты.	54 000
ИТОГО по смете	414 593

4.4 Определение возможной (договорной) цены ПП

Величина возможной (договорной) цены ПП должна устанавливается с учетом эффективности, качества и сроков ее выполнения на уровне, отвечающем экономическим интересам заказчика (потребителя) и исполнителя.

Договорная цена (C_d) для прикладных ПП рассчитывается по формуле:

$$C_d = Z_{\text{нир}} * (1 + (P/100)) , \quad (4.8)$$

где $Z_{\text{нир}}$ - затраты на разработку ПП (из таблицы 4.6), тг;

P - средний уровень рентабельности ПП. % (принято 25%).

$$C_d = 414\,593 * (1 + 0,25) = 518\,242 \text{ тг.}$$

Цена реализации с учетом НДС рассчитывается по формуле:

$$C_p = C_d + C_d * \text{НДС}, \quad (4.9)$$

НДС, согласно Налоговому кодексу РК, составляет 12 %.

$$C_p = 518\,242 + 518\,242 * 0,12 = 580\,431 \text{ тг.}$$

Разработанный программный продукт не нуждается в лицензировании и сертификации, так как этих пунктов нет в требованиях заказчика.

4.5 Расчет срока окупаемости ПП

В результате внедрения геоинформационной системы станет возможным сократить штат сотрудников на 1 человека, и для предприятия расходы на оплату труда будут обходиться на 100 000 тенге меньше ежемесячно (включая СН), чем до внедрения системы.

Расчетный срок окупаемости продукта можно найти по формуле:

$$T_{\text{ок}} = C / \text{Э}, \quad (4.8)$$

где C - затраты на разработку и внедрение системы, тенге;

Э - экономия затрат от внедрения системы, тенге/год.

$$T_{\text{ок}}=580431/100000=5,8(\text{месяцев}).$$

В данном случае срок окупаемости проекта составит 6 месяцев.

4.6 Оценка социально - экономических результатов функционирования программного продукта

До разработки базы данных «Автозапчасти» процесс контроля продаж и учета автозапчастей проводился ручным способом. При этом затрачивалось много времени на выполнение таких функций, как поиск информации о запчастях. Применение разработанной системы контроля автозапчастей позволяет автоматизировать процесс учета запчастей, их групп и видов, а также процедуры продажи, оформление клиентов и учет проработанных часов работников автосервиса.

Реализованная база данных «Автозапчасти» выполняет следующие функции:

- формирование электронной документации;
- расчет и проверка фактической нагрузки работников;
- возможность вывода формы отчетности;
- архивирование данных;
- восстановление данных из базы данных;
- возможность быстрого поиска необходимых данных и их удобное отображение;
- редактирование данных;
- просмотр справочной информации.

Представлен расчет затрат на разработку, подсчитана экономическая эффективность от внедрения базы данных. ПП оправдывает себя уже с первых дней внедрения разработки, исключая бумажную волокиту, экономя время на поиск запчастей и регистрацию клиента. Таким образом, сокращается количество времени на обслуживание одного клиента и автосервис может обслужить большее количество людей. Отпадает проблема хранения больших объемов информации на бумажных и отдельных электронных носителях, сохраняя все данные в единой базе. ПП не требует дальнейших разработок и поддержания специалистами своего рабочего состояния.

5. Безопасность жизнедеятельности

5.1 Анализ условий труда

Темой дипломной работы является - Разработка базы данных «Автозапчасти».

База данных – совокупность информации, которая представлена в объективной форме (статьи, нормативные акты, расчеты, судебные решения и иные подобные данные), позволяющая пользователям хранить информацию в таблицах в строго определенном порядке, имеет функцию поиска, записи, изменения и удаления данных.

Структура работы базы данных «Автозапчасти» довольно-таки проста и при определенном подходе не вызывает никаких затруднений.

Вкратце схема работы следующая:

- пользователь открывает приложение и заполняет таблицу;
- данные сохраняются в базе данных;
- пользователь может в любое время обработать информацию в таблицах.

В данном разделе «Безопасность жизнедеятельности», необходимо проанализировать условия труда пользователей ПО, выявить неблагоприятные влияющие факторы на окружающую среду и разработать конкретные меры по обеспечению безопасности труда пользователей системы.

Рабочее помещение находится на втором этаже двухэтажного здания в комплексе автосервиса. Размеры рабочего помещения: длина 7 м, ширина 4,5 м, высота 3,2 м. В помещении расположено одно окно с размером 3,5м x 2,5м, общая площадь равна 31,5 метр квадрат.

Помещение по зрительным условиям работы относится к IV разряду (размер различаемых при работе предметов от 1 до 10 мм и выше);[10].

Здание относится к I степени огнестойкости : здания с несущими и ограждающими конструкциями из естественных или искусственных материалов, бетона или железобетона с применением листовых и плиточных негорючих материалов;

Выполняемая работа относится к категории лёгких работ (категория 1а), выполняемых в сидячем положении;

Согласно СНиП РК 2.04-05-2002 (Естественное и искусственное освещение), к категории 1а относятся работы, производимые сидя и не требующие физического напряжения, при которых расход энергии составляет до 120 кДж/ч;

- а) число работников - 2;

б) число компьютеров – 2.

Для обеспечения нормальных условий труда санитарные нормы СН 245-71 устанавливают на одного работающего объем производственного помещения не менее 15 м³; площадь помещения выгороженного стенами или глухими перегородками не менее 4,5 м². В данном помещении минимальная площадь составляет порядка 31,5 м², а объем - порядка 100,8 м³, что удовлетворяет условиям санитарной нормы.

Техническим оборудованием, используемым в ходе разработки программного продукта, являются персональные компьютеры:

а) Intel Pentium Dual-Core, 2300 MHz/Intel 965P, 1 GB RAM / HDD 320 Gb;

б) ЖК (LCD) монитор NEC MultiSync EX231W, диагональ 19”;

в) габариты: 1200x750x1150 (персональный компьютер + стол);

г) электропитание: переменное напряжение 220-250 В, частотой 50 Гц, мощность 400 Вт.

Исследования, проведенные в данном разделе дипломного проекта направлены на обеспечение оптимальных условий, режимов труда и отдыха сотрудников, относящихся к категории лиц умственного труда. Практическая реализация требований, изложенных в этом разделе, будет способствовать улучшению условий труда, повышению работоспособности и сохранению здоровья сотрудников автосервиса.

Эти люди сталкиваются с воздействием таких физически опасных и вредных производственных факторов, как:

а) недостаточность естественной и искусственной освещенностей рабочего места;

б) опасность возникновения возгорания.

План рабочего помещения, в котором будет установлено ПО изображен на рисунке 5.1.

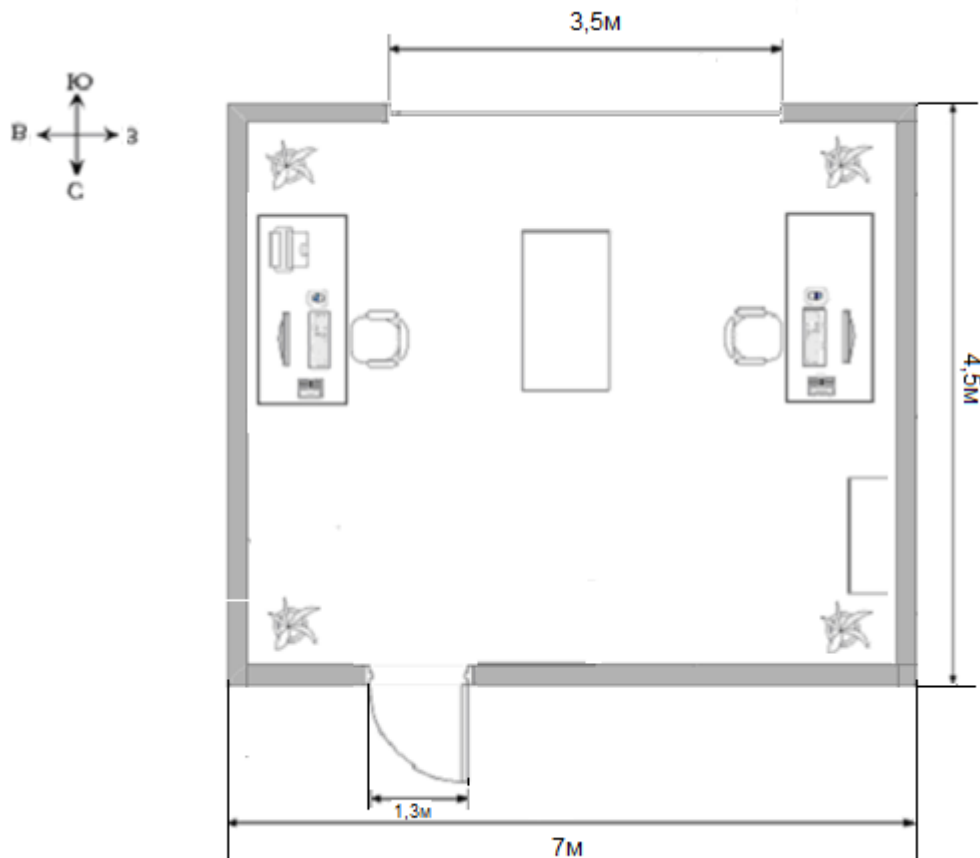


Рисунок 5.1 - План помещения

5.2 Расчет естественного освещения

Помещение имеет размеры: длина a - 7м, ширина b - 4,5м, высота h - 3,2м. Высота рабочей поверхности над уровнем пола – 0,7 м, рабочее место расположено в 1 м от наружной стены помещения, окно начинается с высоты 0,8 м, высота окна 2,5 м, глубина помещения $c=b-1м$ - 3,5м. Минимальная освещённость будет в точке, отстоящей на расстояние 4,5 м от оконного проёма. Расчет будем производить в соответствии с методикой, изложенной в [12].

Общую требуемую площадь окон S_0 , m^2 определим по формуле:

$$100 \cdot \frac{S_0}{S_n} = \frac{e_n \cdot \eta_0}{\tau_0 \cdot r_1} \cdot k_{зд} \cdot k_3, \quad (5.1)$$

$$S_0 = \frac{S_n \cdot e_n \cdot \eta_0 \cdot k_{зд} \cdot k_3}{100 \cdot \tau_0 \cdot r_1}, \quad (5.2)$$

где S_n – площадь помещения, m^2 ,

$$S_n = a \cdot b = 7 \cdot 4,5 = 31,5 \text{ м}^2, \quad (5.3)$$

e_n – нормированное значение КЕО,

$$e_n^{IV} = e_n \cdot m [12], \quad (5.4)$$

$m=0,9$ (см таблицу 3.1[12]);

$e_n=1,5$ для работ средней точности разряда зрительной работы IV, подразряда б (см. таблицу 3.12[12]);

$$e_n^{IV} = 1,5 \cdot 0,9 = 1,35,$$

k_3 – коэффициент запаса;

для офисных помещений $k_3=1,2$ (см. таблицу 3.11[12]);

τ_0 - общий коэффициент светопропускания равный:

$$\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4 \quad (5.5)$$

В качестве светопропускающего материала используем стекло оконное листовое, двойное $\tau_1 = 0,8$.

Вид переплёта – двойной раздельный $\tau_2 = 0,6$.

Вид несущей конструкции – железобетонные формы $\tau_3 = 0,8$.

Солнцезащитные устройства – убирающиеся регулируемые жалюзи.

$\tau_4 = 1$; Общий коэффициент светопропускания $\tau_0 = 0,8 \cdot 0,6 \cdot 0,8 \cdot 1 = 0,384$.

η_0 - световая характеристика окон, определяется из соотношения длины к ширине помещения и ширины к высоте от уровня рабочей поверхности до верха окна;

r_1 - коэффициент, учитывающий повышение КЕО при боковом освещении благодаря свету, отражённому от поверхностей помещения и подстилающего слоя, прилегающего к зданию.

Рассчитаем η_0 .

Отношение длины помещения к глубине $7/3,5=2$; отношение глубины помещения к высоте от уровня рабочей поверхности до верха окна s/h_1 равно $3,5/2,6=1,3$, тогда световая характеристика по таблице 3.2[12] имеет значение $\eta_0=9$.

Рассчитаем r_1 .

Отношение глубины помещения к высоте от уровня рабочей поверхности до верха окна s/h_1 равно $3,5/2,6=1,3$; отношение расстояния расчетной точки от наружной стены к глубине помещения $4,5/3,5=1,3$; Отношение длины комнаты к глубине $7/3,5=2$. Средний коэффициент

отражения в помещении $\rho_{CP} = 0.5$, принимаем одностороннее боковое освещение, тогда по таблице 3.9[12] $r_1 = 1,5$;

$k_{3Д}$ – коэффициент, учитывающий затенение окон противостоящими зданиями;

Поскольку затеняющих зданий поблизости нет, то $k_{3Д} = 1$.

Вычислим общую площадь окон по формуле 5.2.

$$S = (31,5 * 1,35 * 9 * 1 * 1,2) / (100 * 0,384 * 1,5) = 10,2 \text{ м}^2$$

В помещении общая площадь окон в рабочем помещении составляет $8,5 \text{ м}^2$ ($3,5 \times 2,5$).

Так как при проектировании естественного освещения площадь световых проемов рабочего помещения, обеспечивающих нормированное значение КЕО в соответствии с требованиями СНиП РК 2.04-05-2002 «Естественное и искусственное освещение. Нормы проектирования», является больше реальной площади окон, то в дневное время суток дополнительно необходимо использование искусственного освещения.

5.3 Расчет общего искусственного освещения

В рабочем помещении необходимо осуществить естественное и искусственное освещение. Естественное освещение осуществлено оконным проемом, расположенным в помещении с южной стороны. Общее искусственное освещение обеспечивается двумя лампами накаливания мощностью 100 Вт, что не соответствует требованиям к освещению помещений с ПЭВМ. Более того, в ночное время ощущается недостаточность в общем освещении, поэтому произведем расчет необходимого общего искусственного освещения.

Так как во втором этаже здания автосервиса отсутствуют крупные затеняющие предметы, то для расчета общего равномерного освещения рабочих поверхностей столов воспользуемся методом коэффициента использования. Расчет будем производить в соответствии с методикой, изложенной в [12].

Исходные данные для расчета:

1) Размеры помещения:

- длина: 7;
- ширина: 4,5м;
- высота: 3,2м.

2) Коэффициенты отражения по таблице 2.4 [12]:

- потолка $\rho_{ПОТ} = 70\%$;
- стен $\rho_{СТ} = 50\%$;
- пола $\rho_{ПОЛА} = 30\%$.

Разряд зрительной работы IV, подразряд б (таблица 3.12 [12]), нормируемая освещенность для помещений данного типа, равна 300 лк.

Высота стола: 0,8м.

Необходимый поток каждого светильника определяют по формуле:

$$\Phi = \frac{E \cdot K_z \cdot S \cdot Z}{n \cdot N \cdot \eta}, \quad (5.6)$$

где E – заданная нормируемая освещенность;

K_з – коэффициент запаса, для осветительных установок общего освещения принимается равным 1,3 (см. таблицу 3.2 [12]);

n – число ламп в освещаемом помещении;

S – освещаемая площадь, равная $7 \cdot 4,5 = 31,5 \text{ м}^2$;

Z – коэффициент минимальной освещенности, принимаем $Z=1,1$ [12];

N – число светильников;

η – коэффициент использования, равный отношению светового потока падающего на расчетную поверхность к полному потоку осветительного прибора, который находится из таблицы, связывающей геометрические параметры помещения (индекс помещения i) с его оптическими характеристиками (коэффициентами отражения).

Индекс помещения определяется по формуле:

$$i = \frac{A \cdot B}{h \cdot (A + B)}, \quad (5.7)$$

где A – длина помещения;

B – ширина помещения;

h – высота подвеса над рабочей поверхностью (столом), определяется по формуле:

$$h = H - h_c - h_p, \quad (5.8)$$

где H – высота помещения;

h_с – расстояние от светильника до перекрытия, примем h_с = 0,3м;

h_р – высота рабочей поверхности над полом (высота стола), примем h_р = 0,8м.

Тогда расчетная высота: $h = 3,2 - 0,3 - 0,8 = 2,1 \text{ м}$.

Индекс помещения: $i = \frac{7 \cdot 4,5}{2,1 \cdot (7 + 4,5)} = 1,3$

В результате расчетов находим η = 0,62 (см.таблицу 2.5 [12]).

Определимся с типом светильников и с их размещением на потолке. В

качестве источников света при искусственном освещении рабочих столов в операторской, рекомендуется применять люминесцентные лампы типа ЛБ [10]. Поэтому принимаем систему общего освещения люминесцентными лампами типа ЛБ номинальной мощностью 40 Вт, с номинальным световым потоком лампы 3120 лм (см. таблицу 2.2 [12]).

Тогда согласно методике расчета [12] остается определить необходимое количество светильников по формуле

$$N = \frac{E \cdot K_z \cdot S \cdot Z}{n \cdot \Phi \cdot \eta} \quad (5.9)$$

$$\text{Получаем } N = \frac{300 \cdot 1,3 \cdot 31,5 \cdot 1,1}{2 \cdot 3120 \cdot 0,62} = 4 \text{ светильника.}$$

Расстояние между соседними рядами светильников определяют исходя из величины :

$$\lambda = \frac{L}{h} \quad (5.10)$$

равной 1,2-:-1,4. Примем $\lambda = 1,3$.

где L – расстояние между соседними рядами люминесцентных светильников;

h – высота подвеса над рабочим столом.

Получаем, что наиболее выгодное расстояние между светильниками определяется как $L = 1,3 \cdot 2,1 = 2,73\text{м}$ [12]. Примем $L = 2,8\text{м}$.

В таком случае расстояние от стены до ближайшего светильника

$$l = (4,5 - 2,8)/2 = 0,85\text{м.}$$

Получили, что $l = 0,28L$. Так как рабочие столы в операторской расположены непосредственно у стены, то такое соотношение расстояний считается приемлемым [12].

Учитывая, что длина люминесцентной лампы типа ЛБ номинальной мощностью 40 Вт равна 1213,6мм (таблица 2.2 [12]), то принимаем 2 ряда светильников с расстоянием от стен по 0,8м и между рядами по 2,8м, расстояние между светильниками в ряду 0,45м и 0,33м до стены (см. рисунок 5.2).

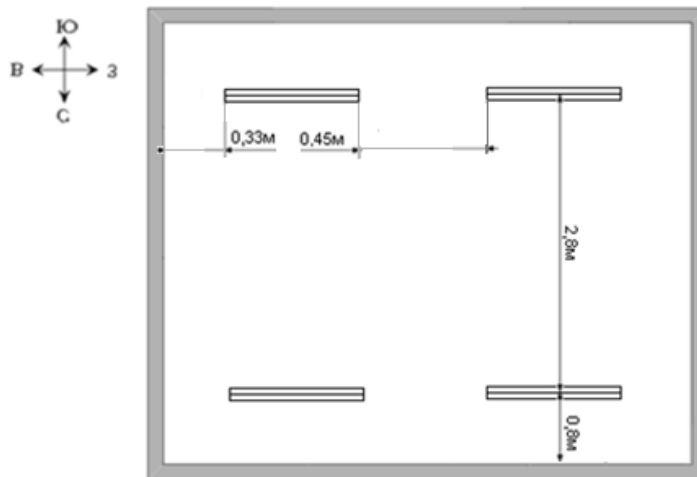


Рисунок 5.2 - Размещение светильников в рабочем помещении

Всего для создания нормируемой освещенности 300 лк понадобится 4 светильников, состоящих из двух люминесцентных ламп типа ЛБ номинальной мощностью 40 Вт.

5.4 Расчет установок водяного и пенного пожаротушения

Расчетный расход воды, раствора пенообразователя Q_d , л/с, через ороситель(генератор) следует определять по формуле

$$Q = k * \sqrt{H}, \quad (5.11)$$

где k – коэффициент производительности оросителя (генератора), принимаемый по таблице 2.2 [13] и равно 0,71;

H – свободный напор перед оросителем(генератором), принимаемый по таблице 2.2 [13] и равно 10 м.

$$Q = 0,71 * \sqrt{10} = 2,2 \text{ л/с} \quad (5.12)$$

Расход воды, раствора пенообразователя Q , л/с, для спринклерной установки в помещении определяется по формуле

$$Q = a * q, \quad (5.13)$$

где a — расчетная длина одновременно орошаемой части стеллажа, принимается равной 15 м; [13]

q_n — интенсивность орошения, принимается по таблице 2.1, для твердых сгораемых материалов равный 0,45 л/см² [13].

$$Q=15*0,45=6,75 \text{ л/с};$$

Потери напора на расчетном участке трубопроводов H , м, определяются по формуле

$$H = Q^2 / B, \quad (5.14)$$

где Q - расход воды, раствора пенообразователя на расчетном участке трубопровода, л/с;

B - характеристика трубопровода, определяется по формуле:

$$B = k / l, \quad (5.15)$$

где k - коэффициент, принимается по таблице 2.3[13], равно 0,18;

l - длина расчетного участка трубопровода, равно 6,7 м.

$$B=0,18/6,7=0,02 \text{ м};$$

$$H = 6,75^2 / 0.02 = 2,2 * 10^3 \text{ м}.$$

Потери напора в узлах управления установок H , определяются по формуле:

$$H = \epsilon_v * Q^2, \quad (5.16)$$

где ϵ_v - коэффициент потерь напора в узле управления, принимается по таблице 2.4 [13], для спринклерной водозаполненной установки равно $3,02 * 10^3$;

Q - расчетный расход воды, раствора пенообразователя через узел управления, л/с.

$$H = 3,02 * 10^3 * 2,2 = 6,6 * 10^3 \text{ м}$$

Объем раствора пенообразователя V , при объемном пожаротушении определяется по формуле

$$V = k_* * V / k, \quad (5.17)$$

где k_* - коэффициент разрушения пены, принимаемый по таблице 2.5[13], для твердых горючих материалов равно 3;

V — объем защищаемого помещения, м³;

k — кратность пены, для химической пены равно 5.

$$V = 3 * 100,8 / 5 = 60,8 \text{ м}^3$$

Число одновременно работающих генераторов пены n определяется по формуле

$$n = V / Q * t, \quad (5.18)$$

где t - продолжительность работы установки с пеной средней кратности, мин, принимается по таблице 2.5 [13], равно 25.

$$n = 60,8 / 2,2 * 25 = 1,1 \approx 1$$

Таким образом, для обеспечения пожаротушения рабочего помещения общей объемом $V=100,8 \text{ м}^3$, вполне достаточен один генератор (ороситель) пены.

Заключение

В данном дипломном проекте была разработана БД «Автозапчасти». До разработки процесс контроля автозапчастей и учета клиентов проводился ручным способом. При этом затрачивалось много времени на выполнение таких функций, как ведение счета запчастей. Также были рассмотрены вопросы разработки базы данных для хранения информации, реализация пользовательского интерфейса для упрощения работы пользователя (оператора).

Программная система позволяет формировать документацию, необходимую для организации и контроля автозапчастей и клиентов.

Реализованная автоматизированная система контроля автозапчастей выполняет следующие функции:

- формирование электронной документации;
- расчет и проверка количества запчастей и их состояния;
- возможность вывода формы отчетности;
- возможность быстрого поиска необходимых данных и их удобное отображение;
- редактирование данных.

Основными операциями, связанными с обработкой документов, являются: создание и редактирование, добавление информации о новых клиентах, хранение и поиск, контроль исполнения. Автоматизация позволяет снизить затраты на обработку документов, повысить качество управленческих решений за счет ускорения процессов поиска, обработки и предоставления нужной пользователю информации, что, в результате, обеспечивает рост эффективности функционирования управляемого объекта.

В дипломном проекте было дано экономическое обоснование необходимости разработки системы. Представлен расчет затрат на разработку, подсчитана экономическая эффективность от внедрения автоматизированной системы документооборота. Срок окупаемости ПП составил 6 месяцев.

В последнем разделе был проведен анализ условий труда и были выявлены факторы, отрицательно влияющие на жизнь и здоровье персонала автосервиса. Было определено, что наибольшее влияние на условия труда работников автосервиса оказывают недостаточность естественной и искусственной освещенностей, а также пожароопасность помещения.

Для борьбы с данными факторами были рассчитаны и подобраны системы искусственного освещения – были выбраны лампы ЛБ мощностью 40 Вт в количестве 4-ти штук. Проведен расчет установок водяного и пенного пожаротушения. Определено количество оросителей (генераторов) необходимых для пожаротушения в рабочем помещении. Исследования, проведенные в данном разделе дипломного проекта, направлены на обеспечение оптимальных условий, режимов труда и отдыха работников,

пользующихся базой данных «Автозапчасти», относящихся к категории лиц умственного труда. Практическая реализация требований, изложенных в этом разделе, будет способствовать улучшению условия труда, повышению работоспособности и сохранению здоровья сотрудников.

Список литературы

1. Томас К., Каролин Б. Базы данных. Проектирование, реализация и сопровождение. - СПб.: Вильямс, 2003. – 1440 с.
2. Райордан Р. Основы реляционных баз данных.-Москва.: Русская редакция, 2001. – 384 с.
3. Гладченко А. Создание гибкой системы безопасности [Электронный ресурс]. – Режим доступа: <http://www.sql.ru/articles/mssql/01061605.shtml>.
4. У. Боггс, М. Боггс «UML и RationalRose 2002» - Издательство «ЛЮРИ», 2004.– 415 с.
5. Диаграммы UML [Электронный ресурс] – Режим доступа: <http://www.intuit.ru/studies/courses>
6. Бекишева А. И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В070300 – Информационные системы – Алматы: АУЭС; 2013. – 24с.
7. Хакимжанов Т. И. Методические указания к выполнению раздела в дипломных проектах(для студентов всех форм обучения всех специальностей) – Алматы: АИЭС; 2002. – 29с.
8. Основы UML – Разработка диаграмм в среде RationalRose [Электронный ресурс] – Режим доступа:<http://2programmer.ru/uml/>
9. Грабер М. SQL:[пер. с англ.] / М.Грабер.-М: Издательство «Лори», 2003.– 854 с.
10. СНиП РК 2.04-05-2002, Естественное и искусственное освещение. Общие требования. Комитет по делам строительства Министерства индустрии РК. - Астана, 2002г. – 512 с.
11. Мананбаева С.Е. Производственное освещение. Методические указания к выполнению раздела «Безопасность жизнедеятельности» в выпускной работе.- Алматы: АИЭС. 2004г. – 312 с.
12. Ж. С. Абдимуратов, С. Е. Мананбаева. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат – Алматы: АИЭС, 2009.
13. Абикенова А.А., Методические указания к выполнению раздела «Пожарная профилактика» - Алматы: АИЭС, 2009.

Список сокращений

СЭД - система электронного документооборота;
САДЭД - системы автоматизации делопроизводства и электронного документооборота;
СУФ – система управления файлами;
БД – база данных;
ПО – программное обеспечение;
ПП – программный продукт;
MS – microsoft;
СУБД – система управления базой данных;
АУЭС – Алматинский университет энергетики и связи;
SQL (structured query language) - язык структурированных запросов;
ПК – персональный компьютер.

Приложение А

Техническое задание

А.1 Общие положения

А.1.1 Полное наименование системы и ее условное обозначение

Полное наименование системы: База данных «Автозапчасти». Краткое наименование системы: БД «Автозапчасти»

А.1.2 Шифр темы или шифр (номер) договора

Шифр темы: БД «Автозапчасти» -14.
Номер контракта: №1/10-09-13-001 от 10.09.2014 г.

А.1.3 Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты

Заказчиком системы является ИП «А3-07», г. Алматы.
Адрес заказчика: 050025, Республика Казахстан, г. Алматы, ул. Абылайхана, 28.

Разработчиком системы является, Талап Дина Данияркызы, студент АУЭС, ФИТ, группа ИС 10-2.

Адрес разработчика: г. Алматы, ул. Шашкина, 14.

А.1.4 Перечень документов, на основании которых создается система, кем и когда утверждены эти документы

Основанием для разработки БД «Автозапчасти» являются следующие документы и нормативные акты:

Контракт №1/10-09-13-001 от 10.09.2014 года на выполнение работ по выполнению первого этапа работ по созданию базы данных «Автозапчасти»;

Договор разработчика с заказчиком о выполнении работы по созданию системы в указанные сроки, утвержденные директором ИП «А3-07» Орманбаевым А.А.;

Контракт;

На основе задания на дипломный проект.

Продолжение приложения А

А.1.5 Плановые сроки начала и окончания работы по созданию системы

Плановый срок начала работ по созданию БД «Автозапчасти» - 10 сентября 2013 года.

Плановый срок окончания работ по созданию БД «Автозапчасти» - 25 мая 2014 года.

А.1.6 Сведения об источниках и порядке финансирования работ

Источником финансирования является бюджет ИП «АЗ-07».

Порядок финансирования устанавливается договорами директора ИП «АЗ-07».

А.1.7 Порядок оформления и предъявления заказчику результатов работ

Система показывается в виде функционирующего комплекса на базе средств вычислительной техники заказчика в сроки, определенные заказчиком. Приемка системы реализовывается комиссией в составе уполномоченных представителей заказчика.

Порядок предъявления системы, ее проверок и окончательной приемки назначен в п.6 настоящего ТЗ. Сообща с предъявлением системы делается сдача разработанного исполнителем набора документации согласно п.8 настоящего ТЗ.

А.1.8 Состав используемой нормативно-технической документации

При разработке информационной системы и организации проектно-эксплуатационной документации Исполнитель обязан следовать запросами следующих нормативных документов:

ГОСТ 34.601-90. Комплекс образцов на информационные системы. Информационных системы. Этапы создания;

ГОСТ 34.201-89. Информационная технология. Комплекс образцов на информационные системы. Виды, комплексность и обозначение документов при производстве информационных систем;

РД 50-34.698-90. Методические указания. Информационная технология. Комплекс образцов на информационные системы. Информационные системы. Требования к содержанию документов.

Продолжение приложения А

А.2 Назначение и цели создания (развития) системы

– рекомендации системы – проектируемая система предназначена для применения её сотрудниками ИП, т.е операторами и администратором. Её назначение – автоматизация деятельности ИП.

– цели создания системы – повышение результативности работы ИП за счет оперативного обмена данными и создания систематизированной базы данных.

А.2.1 Назначение ИС

БД «Автозапчасти» предназначена для комплексного информационно-аналитического обеспечения процессов деятельности ИП «АЗ-07», в части исполнения следующих процессов:

– хранение информации в базе данных с обеспечением удобного, быстрого поиска необходимых данных и надежной системы разграничения прав доступа;

– резервное дублирование(бэкап) данных;

– организация оперативной деятельности по обработке данных;

– обеспечение доступа администратора к информации, имеющей конфиденциальный характер;

– организация контроля автозапчастей и учета клиентов

А.2.2 Основные цели создания ИС «BestLine»

– снижение временных затрат на обработку данных;

– получение отчетов о выполненных операциях, используя созданную базу данных;

– предоставление доступа к информации только авторизованным сотрудникам;

Для реализации поставленных целей система должна решать следующие задачи:

– гарантирование доступа к информации и информационным ресурсам, средствам информатизации авторизованным пользователям;

– иметь удобный и понятный интерфейс;

– оперативная обработка и активация заказов;

Продолжение приложения А

А.3 Характеристика объекта автоматизации

А.3.1 Объект автоматизации

Процессы по управлению БД "Автозапчасти", а также проверку результативности выполнения показанных процессов. Предоставленные процессы исполняются руководствующимися специалистами:

- персонал ИП.

А.3.2 Существующее программное обеспечение

В данный момент деятельность ИП «АЗ-07» не автоматизирована.

А.3.3 Существующее нормативно-правовое обеспечение

Существующее нормативно-правовое снабжение составляют федеральные и областные нормативные правовые акты:

- Конституция РК;
- Гражданский кодекс РК и т.д.

А.4 Требования к системе

Запросы к системе в целом:

- запросы к структуре и функционированию системы;
- запросы к персоналу системы;
- показатели рекомендации;
- запросы к надежности; безопасности; эргономике и технической эстетике; эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы; защите информации; сохранности информации при авариях; предохранении от воздействия внешних действий;
- требования к патентной чистоте; стандартизации и унификации; основные требования.

Требования к функциям (задачам), осуществляемым системой; перечень функций, задач или их комплексов, предикатов автоматизации (по каждой подсистеме); очередность ввода в эксплуатацию, временной регламент исполнения и запросы к качеству осуществления любой функции, задачи (или комплекса задач), к форме представления выходной информации, перечень и критерии отказов для любой функции, по которой задаются запросы по надежности.

Продолжение приложения А

А.4.1 Требования к системе в целом

А.4.1.1 Требования к структуре и функционированию системы

А.4.1.1.1 В составе БД "Автозапчасти" должны решаться следующие задачи:

- резервное дублирование информации, содержащейся в БД;
- система хранения информации;

Подсистема резервного дублирования информации подразумевает обеспечение целостности и сохранение одной из главных ценностей системы накопленная база данных, сбой которой приведет непоправимым последствиям для ИП. Поэтому необходимо исключить вероятность потери и порчи данных, при том, что для больших нагруженных БД при ошибке сервера эта вероятность близка к 100%.

А.4.1.1.2 Требования к способам и средствам связи для информационного обмена между компонентами системы

Входящие в состав БД «Автозапчасти» подсистемы в процессе функционирования должны обмениваться информацией на основе раскрытых форматов обмена данными, употребляя для этого входящие в их состав модули информационного взаимодействия.

Форматы данных будут разработаны и ратифицированы на этапе технического проектирования.

А.4.1.1.3 Требования к режимам функционирования системы

Для БД «Автозапчасти» назначены следующие режимы функционирования:

- нормальный режим функционирования;
- аварийный режим функционирования.

Основным режимом функционирования БД является нормальный режим.

В нормальном режиме функционирования системы:

- клиентское программное обеспечение и технические средства пользователей и администратора системы обеспечивают вероятность функционирования в течение рабочего дня (с 00:00 до 00:00) семь дней в неделю;

- серверное программное обеспечение и технические средства серверов обеспечивают возможность круглосуточного функционирования, с перерывами на обслуживание;

– исправно действует оборудование, составляющее комплекс технических средств;

Продолжение приложения А

– исправно работает системное, основное и главное программное обеспечение системы.

Для обеспечения нормального режима функционирования системы нужно реализовывать требования и выносить условия эксплуатации программного обеспечения и комплекса технических средств системы, показанные в подобающих технических документах (техническая документация, инструкции по эксплуатации и т.д.).

В случае перехода системы в аварийный режим необходимо:

- закончить эффективность всех приложений, с сохранением данных;
- отключить персональные компьютеры;
- отключить все периферийные устройства;
- исполнить резервное копирование БД.

После этого надобно исполнить комплекс мероприятий по ликвидации причины перехода системы в аварийный режим.

А.4.1.1.4 Требования по диагностированию системы

БД «Автозапчасти» должна предоставлять инструменты диагностирования главных процессов системы, трассировки и мониторинга процесса выполнения программы.

Компоненты обязаны предоставить удобный интерфейс для вероятности просмотра диагностических событий, мониторинга процесса выполнения программ.

При возникновении аварийных обстоятельств, либо ошибок в программном обеспечении, диагностические инструменты обязаны разрешать сохранять полный набор информации, необходимой разработчику для идентификации проблемы (снимки экранов, текущее состояние памяти, файловой системы).

А.4.1.1.5 Перспективы развития, модернизации системы

БД «Автозапчасти» обязана исполнять вероятность дальнейшей модернизации как программного обеспечения.

А.4.1.2 Запросы к количеству и квалификации персонала системы

Для эксплуатации БД «Автозапчасти» назначены вытекающие роли:

- системный администратор;
- пользователь (оператор).

Главными обязанностями системного администратора являются:

– установка, модернизация, настройка и мониторинг работоспособности системного и базового программного обеспечения;

Продолжение приложения А

– установка, настройка и мониторинг прикладного программного обеспечения;

– ведение учетных записей пользователей системы;

– системный администратор обязан владеть значительным уровнем квалификации и практическим опытом выполнения работ по установке, настройке и администрированию программных и технических средств, употребляемых в системе.

Главными обязанностями администратора баз данных являются:

– установка, модернизация, настройка параметров программного обеспечения СУБД;

– оптимизация прикладных баз данных по времени отклика, скорости доступа к данным;

– разработка, управление и исполнение результативной политики доступа к информации, держащейся в теоретических базах данных;

– администратор баз данных должен иметь значительный уровень квалификации и практическим опытом работ по установке, настройке и администрированию используемых в АС СУБД.

Пользователи системы обязаны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows на уровне квалифицированного пользователя и легко исполнять базовые операции в стандартных Windows.

Рекомендуемая численность для эксплуатации БД «Автозапчасти»:

Администратор - 1 штатная единица;

Пользователь - число штатных единиц определяется структурой заведения.

А.4.1.3 Показатели назначения

БД «Автозапчасти» обязаны обеспечивать вероятность исторического сохранения данных с глубиной не менее 10 лет.

Система обязана обеспечивать возможность работы при вытекающих характеристиках времени отклика системы:

– для операций навигации по экранным формам системы - не более 5 сек;

– для операций выработки справок и выписок - не более 10 сек.

Время формирования аналитических отчетов определяется их сложностью и возможно будет занято продолжительное время.

Система обязана предусмотреть вероятность масштабирования по продуктивности и объему обрабатываемой информации без модификации ее

программного обеспечения путем улучшения употребляемых технических средств.

Продолжение приложения А

А.4.1.4 Требования к надежности

Система обязана беречь работоспособность и обеспечивать воссоздание своих функций при возникновении вытекающих ситуаций:

– при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы следует происходить после перезапуска ОС и запуска осуществляемого файла системы;

– при ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функции системы возлагается на ОС;

– при ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

Для защиты аппаратуры от бросков напряжения и коммутационных помех должны использоваться стабилизаторы.

А.4.1.5 Требования к безопасности

Все наружные элементы технических средств системы, находящиеся под напряжением, обязаны обладать защитой от намеренного прикосновения, а сами технические средства иметь занижение или защитное заземление в соответствии с ГОСТ 12.1.030-81.

Система электропитания обязана снабжать защитное отключение при перегрузках и недолгих замыканиях в цепях нагрузки, а также аварийное ручное отключение.

Факторы, проявляющие вредные действия на здоровье со стороны всех элементов системы (в том числе инфракрасное, ультрафиолетовое, рентгеновское и электромагнитное излучения, вибрация, шум, электростатические поля, ультразвук строчной частоты и т.д.), не должны превосходить функционирующих норм.

А.4.1.6. Требования к эргономике и технической эстетике

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм. Навигационные элементы должны быть выполнены в удобной для пользователя форме. Средства редактирования информации должны удовлетворять принятым соглашениям в части использования

функциональных клавиш, режимов работы, поиска, использования оконной системы. Ввод-вывод данных системы, прием управляющих команд и

Продолжение приложения А

отображение результатов их исполнения должны выполняться в интерактивном режиме. Интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям и операциям системы.

Управление системой должно осуществляться с помощью набора экранных меню, кнопок, значков и т. п. элементов. Клавиатурный режим ввода должен использоваться главным образом при заполнении и/или редактировании текстовых и числовых полей экранных форм.

Система должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных.

Система должна соответствовать требованиям эргономики и профессиональной медицины при условии комплектования высококачественным оборудованием (ПЭВМ, монитор и прочее оборудование), имеющим необходимые сертификаты соответствия и безопасности.

А.4.1.7 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Техническая и физическая защита аппаратных компонентов системы, носителей данных, бесперебойное энергоснабжение, резервирование ресурсов, текущее обслуживание реализуется техническими и организационными средствами, предусмотренными в ИТ инфраструктуре Заказчика.

Для нормальной эксплуатации разрабатываемой системы должно быть обеспечено бесперебойное питание ПК. При эксплуатации система должна быть обеспечена соответствующая стандартам хранения носителей и эксплуатации ПК температура и влажность воздуха.

Периодическое техническое обслуживание используемого ПК должно проводиться в соответствии с требованиями технической документации изготовителей, но не реже одного раза в год.

Периодическое техническое обслуживание и тестирование ПК должны включать в себя обслуживание и тестирование ПК, кабельной системы, устройств безопасной передачи данных.

В процессе проведения периодического технического обслуживания должны проводиться внешний и внутренний осмотр и чистка ПК, проверка контактных соединений, проверка параметров настроек работоспособности ПК.

Продолжение приложения А

А.4.1.8 Требования к защите информации от несанкционированного доступа

БД должна обеспечивать защиту от несанкционированного доступа (НСД).

Компоненты подсистемы защиты от НСД должны обеспечивать:

- идентификацию пользователя;
- проверку полномочий пользователя при работе с системой;
- разграничение доступа пользователей на уровне задач и информационных массивов.

Протоколы аудита системы и приложений должны быть защищены от несанкционированного доступа локально.

А.4.1.9 Требования по сохранности информации при авариях

Программное обеспечение БД «Автозапчасти» должно восстанавливать свое функционирование при корректном перезапуске аппаратных средств. Обязана быть внезапна вероятность объединения автоматического и ручного запасного имитирования данных системы оружием системного и базового программного обеспечения (ОС, СУБД), убирающегося в состав программно технического комплекса Заказчика.

Приведенные выше требования не распространяются на компоненты системы, разработанные третьими сторонами и действительны только при соблюдении правил эксплуатации этих компонентов, включая своевременную установку обновлений, рекомендованных производителями покупного программного обеспечения.

А.4.1.10 Требования к патентной чистоте

Установка системы в целом, как и установка отдельных частей системы не должна предъявлять дополнительных требований к покупке лицензий на программное обеспечение сторонних производителей.

А.4.1.11 Требования по стандартизации и унификации

Экранные формы должны проектироваться с учетом требований унификации:

- все экранные формы пользовательского интерфейса должны быть осуществлены в едином графическом дизайне, с одинаковым расположением главных элементов управления и навигации;
- для обозначения сходных операций должны использоваться сходные графические значки, кнопки и другие управляющие элементы;

Продолжение приложения А

- термины, применяемые для обозначения типовых операций а также последовательности действий пользователя при их выполнении, должны быть унифицированы;
- внешнее поведение сходных элементов интерфейса (реакция на наведение указателя "мыши", переключение фокуса, нажатие кнопки) должны реализовываться одинаково для однотипных элементов.

А.4.2 Требования к видам обеспечения

А.4.2.1 Требования к математическому обеспечению системы

Математические методы и алгоритмы, применяемые для автоматизации расчетов автозапчастей должны быть сертифицированы уполномоченными организациями.

А.4.2.2 Требования к информационному обеспечению системы

Состав, структура и способы организации данных в системе должны быть определены на этапе технического проектирования.

Хранение данных должно осуществляться на основе нынешних реляционных или СУБД. Для обеспечения целостности данных должны использоваться встроенные механизмы СУБД.

Средства СУБД, а также средства употребляемых операционных систем обязаны снабжать документирование и протоколирование отшлифовываемой в системе информации.

Структура базы данных обязана поддерживать кодирование хранимой и обрабатываемой информации в соответствии с общепринятыми классификаторами (там, где они применимы).

Доступ к данным должен быть дан только авторизованным пользователям с учетом их служебных полномочий, а также с учетом категории запрашиваемой информации.

Структура базы данных должна быть организована рациональным способом, исключающим единовременную полную выгрузку информации, содержащейся в базе данных системы.

Технические средства, обеспечивающие сохранение информации, обязаны применить нынешние технологии, разрешающие обеспечить увеличенную безопасность сохранения данных и оперативную подмену

оборудования (распределенная избыточная запись/считывание данных; зеркалирование).

В состав системы должна входить специализированная подсистема резервного копирования данных.

Продолжение приложения А

А.4.2.3 Требования к лингвистическому обеспечению системы

Все прикладное программное обеспечение системы для организации взаимодействия с пользователем должно применить русский язык.

А.4.2.4 Требования к программному обеспечению системы

При проектировании и разработке системы необходимо максимально эффективным образом применить ранее закупленное программное обеспечение.

А.4.2.5 Требования к техническому обеспечению

Техническое обеспечение системы должно предельно и наиболее результативным образом применить существующие в органах федерального агентства технические средства.

В состав комплекса должны следующие технические средства:

ПК пользователя;

А.4.2.6 Требования к организационному обеспечению

Организационное обеспечение системы следует быть правильным для результативного выполнения персоналом положенных на него обязанностей при реализации автоматизированных и связанных с ними неавтоматизированных функций системы.

Заказчиком должны быть определены должностные лица, ответственные за:

- обработку информации АС;
- администрирование АС;
- управление работой персонала по обслуживанию АС.

К работе с системой должны допускаться сотрудники, обладающие навыками работы на персональном компьютере, ознакомленные с правилами эксплуатации и прошедшие обучение работе с системой.

А.4.2.7 Требования к методическому обеспечению

В состав методического обеспечения системы должны входить законодательные акты, стандарты, нормативы, инструкции.

Продолжение приложения А

А.5 Порядок контроля и приемки системы

А.5.1 Виды, состав, объем и методы испытаний системы

Виды, состав, объем, и методы испытаний подсистемы должны быть изложены в программе и методике испытаний БД «Автозапчасти», разрабатываемой в составе рабочей документации.

А.5.2 Общие требования к приемке работ по стадиям

Сдача-приёмка работ изготавливается поэтапно, в соответствии с рабочей программой и календарным планом.

Сдача-приемка выполняется комиссией, в состав которой входят представители Заказчика и Исполнителя.

Все создаваемые в рамках настоящей работы программные изделия (за исключением покупных) передаются Заказчику, как в виде готовых модулей, так и в виде исходных кодов, показываемых в электронном варианте на стандартном носителе (например, на флэш карте).

А.5.3 Статус приемочной комиссии

Статус приемочной комиссии определяется Заказчиком до проведения испытаний.

А.6 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

В ходе выполнения проекта на объекте автоматизации нужно реализовать работы по подготовке к вводу системы в воздействие. При подготовке к вводу в эксплуатацию БД «Автозапчасти» Заказчик должен выполнить следующие работы:

Назначить ответственных должностных лиц, ответственных за внедрение и проведение эксплуатации БД «Автозапчасти»;

Обеспечить присутствие пользователей на обучении работе с системой, проводимом Исполнителем;

Совместно с Исполнителем подготовить план развертывания системы на технических средствах Заказчика;

Провести опытную эксплуатацию БД «Автозапчасти».

Окончание приложения А

А.7 Требования к документированию

Данный проект сопровождается ТЗ и документацией на технический проект.

В техническом задании описываются:

- главные цели, задачи, сроки и периоды разработки;
- список важнейших функций и требований;
- список функций интерфейса.

Документация на технический проект является инструкцией по употреблению данного ПО. В данной документации будут описываться:

- условия работы ПО;
- установка ПО;
- применение ПО, по пунктом изображение важнейших функций и функций интерфейса;
- ликвидация проблем при переходе в аварийный режим.

Для системы на различных стадиях создания должны быть выпущены следующие документы из числа предусмотренных в ГОСТ 34.201-«Информационная технология. Комплекс стандартов на автоматизированные системы».

А.8 Источники разработки

Документы и информационные материалы (технико-экономическое обоснование, отчеты о законченных научно-исследовательских работах, информационные материалы на отечественные, зарубежные системы-аналоги и др.), на создании которых разрабатывалось ТЗ и которые обязаны быть применены при основании системы.

Технико-экономическое обоснование. Этот документ хранит, финансовое отображение системы, в котором хранится перечень применяемых ресурсов и показывается их цена. Цена системы, подсчет рентабельности. Минимизация употребляемых ресурсов для того чтобы получать наибольшей прибыли.

Данная система должна разрабатываться на основании ТК 34 по стандартизации Информационные технологии. Номер приказа и дата утверждения: от 27.07.01 г. № 274.

Приложение Б

Листинг программы

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ShellAPI;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Label1: TLabel;
    Edit1: TEdit;
    Button2: TButton;
    Button3: TButton;
    Label2: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Edit1KeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
uses Unit3, Uchet_metodi4ek;
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  if Edit1.Text = " then
```

```
begin
  Button2.Enabled:=True;
  Button3.Enabled:=True;
  Label2.Enabled:=True;
```

Продолжение приложения В

```
Label2.Visible:=True;
  Label1.Visible:=False;
  Button1.Visible:=False;
  Edit1.Visible:=False;
end
else
```

```
begin
  ShowMessage ('Неправильный пароль!');
  Edit1.Clear;
end;
end;
```

```
procedure TForm1.Edit1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
```

```
begin
  If key=13
  then
  begin
    key:=0;
    Button1.click;
  end;
```

```
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
  Application.CreateHandle;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
```

```
begin
  Form3.Showmodal;
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);
```

```
begin
  Form4.Showmodal;
end;
```

end.
unit Unit3;

Продолжение приложения В

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, DB, ADODB, ExtCtrls, DBCtrls, Grids, DBGrids, DBTables,
ShellAPI,

Menus, ComCtrls, Uchet_metodi4ek, Buttons, ToolWin, QRCtrl,
Printers, ComObj;

type

TForm3 = class(TForm)
 DataSource1: TDataSource;
 ADOConnection2: TADOConnection;
 ADOQuery1: TADOQuery;
 ADOQuery2: TADOQuery;
 ADOConnection1: TADOConnection;
 MainMenu1: TMainMenu;
 N21: TMenuItem;
 ADOQuery3: TADOQuery;
 MainMenu2: TMainMenu;
 N11: TMenuItem;
 w1: TMenuItem;
 OpenDialog1: TOpenDialog;
 PopupMenu1: TPopupMenu;
 N1: TMenuItem;
 N2: TMenuItem;
 N3: TMenuItem;
 N4: TMenuItem;
 N6: TMenuItem;
 SaveDialog1: TSaveDialog;
 Panel3: TPanel;
 ScrollBar1: TScrollBar;
 CheckBox1: TCheckBox;
 CheckBox10: TCheckBox;
 CheckBox11: TCheckBox;
 CheckBox12: TCheckBox;
 CheckBox13: TCheckBox;
 CheckBox14: TCheckBox;

CheckBox15: TCheckBox;
CheckBox16: TCheckBox;
CheckBox17: TCheckBox;
CheckBox18: TCheckBox;

Продолжение приложения В

CheckBox20: TCheckBox;
CheckBox3: TCheckBox;
CheckBox4: TCheckBox;
CheckBox5: TCheckBox;
CheckBox6: TCheckBox;
CheckBox7: TCheckBox;
CheckBox8: TCheckBox;
CheckBox9: TCheckBox;
Panel4: TPanel;
ToolBar1: TToolBar;
ImportButton: TBitBtn;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
BitBtn4: TBitBtn;
BitBtn5: TBitBtn;
BitBtn3: TBitBtn;
BitBtn8: TBitBtn;
ProgressBar1: TProgressBar;
TabControl1: TTabControl;
DBGrid1: TDBGrid;
Panel2: TPanel;
Button4: TButton;
Memo9: TMemo;
Memo1: TMemo;
Memo8: TMemo;
Memo7: TMemo;
Memo6: TMemo;
Memo5: TMemo;
Memo4: TMemo;
Memo3: TMemo;
 Memo2: TMemo;
Memo14: TMemo;
Memo13: TMemo;
Memo12: TMemo;
Memo11: TMemo;
Memo10: TMemo;
Label3: TLabel;
N7: TMenuItem;

N8: TMenuItem;
N9: TMenuItem;
Excel1: TMenuItem;
Excel2: TMenuItem;

Продолжение приложения В

N10: TMenuItem;
N5: TMenuItem;
N12: TMenuItem;
Word1: TMenuItem;
ExportButton: TBitBtn;
Label6: TLabel;
Label2: TLabel;
ComboBox2: TComboBox;
Button2: TButton;
Button5: TButton;
Label7: TLabel;
Label1: TLabel;
ComboBox1: TComboBox;
Button3: TButton;
BitBtn6: TBitBtn;
BitBtn7: TBitBtn;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button3Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure DBGrid1KeyDown(Sender: TObject; var Key: Word;
 Shift: TShiftState);
procedure Button7Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure N11Click(Sender: TObject);
procedure w1Click(Sender: TObject);
procedure TabControl1Change(Sender: TObject);
procedure Button2MouseDown(Sender: TObject; Button: TMouseButton;
 Shift: TShiftState; X, Y: Integer);
procedure ComboBox2Select(Sender: TObject);
procedure ComboBox2KeyDown(Sender: TObject; var Key: Word;
 Shift: TShiftState);
procedure ComboBox2Exit(Sender: TObject);
procedure ComboBox2Change(Sender: TObject);
procedure ImportButtonClick(Sender: TObject);
procedure ADOQuery1AfterPost(DataSet: TDataSet);

```
procedure BitBtn2Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure N1Click(Sender: TObject);
```

Продолжение приложения В

```
procedure N2Click(Sender: TObject);
procedure CheckBox1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure ComboBox1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure BitBtn1Click(Sender: TObject);
procedure ExportButtonClick(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure Word1Click(Sender: TObject);
procedure N5Click(Sender: TObject);
procedure N12Click(Sender: TObject);
procedure ADOQuery1AfterOpen(DataSet: TDataSet);
procedure BitBtn8Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form3: TForm3;
  bfromstr,fromstr,afromstr,qsbi,csbi,CurrentDBName:string;
  viscolcount:integer;
implementation

uses Unit1, Unit5;

{$R *.dfm}

Function GetYears:string;
begin
if StrToInt(copy(DateToStr(Date),4,2))<9 then
```

```

GetYears:=IntToStr(StrToInt(copy(DateToStr(Date),length(DateToStr(Date))-3,4))-
1)+
  '-' + IntToStr(StrToInt(copy(DateToStr(Date),length(DateToStr(Date))-3,4)))
else

```

Продолжение приложения В

```

GetYears:=IntToStr(StrToInt(copy(DateToStr(Date),length(DateToStr(Date))-
3,4)))+
  '-' + IntToStr(StrToInt(copy(DateToStr(Date),length(DateToStr(Date))-3,4))+1);

```

end;

Procedure ImportFromExcel;

var i,j,ToAll:integer;

var s,ss,f,k:string;

varDlgResult:Word;

var cncl:Boolean;

begin

cncl:=FALSE;

CopyFile(PChar(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb'),PChar(Ex
tractFilePath(ParamStr(0))+CurrentDBName+'.tmp'),FALSE);

ToAll:=0;

Uchet_metodi4ek.Form4.OpenDialog1.Title:='Выберите
расположение файла';

Uchet_metodi4ek.Form4.OpenDialog1.Execute;

Uchet_metodi4ek.Form4.OpenDialog1.Title:="";

s:=Uchet_metodi4ek.Form4.OpenDialog1.FileName;

ss:=ExtractFileName(Uchet_metodi4ek.Form4.OpenDialog1.FileName);

if s="" then

begin

ShowMessage('Файл для импорта не указан или указан неверно.

Импорт отменён.');

exit;

end;

f:=""; k:=""; j:=0;

for i:=1 to length(ss) do

begin

if ss[i]='_' then

begin

j:=j+1;

continue;

end;

if j=0 then f:=f+ss[i];


```

    if j=1 then k:=k+ss[i];
    end;
    if j<2 then
    begin
    f:="";

```

Продолжение приложения В

```

k:="";
end;

```

```

Form3.ADOConnection1.ConnectionString:='Provider=Microsoft.Jet.OLEDB.4.0;
Data Source='+s+';Extended Properties=Excel 8.0';
Form3.ADOQuery1.Connection:=Form3.ADOConnection2;
Form3.ADOQuery2.Connection:=Form3.ADOConnection1;

```

```

Form3.ADOConnection2.ConnectionString:='Provider=Microsoft.Jet.OLEDB.4.0;
Data Source="'+ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb"';
Form3.ADOConnection2.Open;
Form3.ADOConnection1.Open;
Form3.Memo1.Clear;
Form3.ADOConnection1.GetTableNames(Form3.Memo1.Lines);
if k<>" then
for i:=1 to Form3.Memo1.Lines.Count do
begin
Form3.ADOQuery1.SQL.Text:='SELECT Кафедра FROM
'+copy(Form3.Memo1.Lines[i-1],1,length(Form3.Memo1.Lines[i-1])-1)+' where
Кафедра="'+k+'";
Form3.ADOQuery1.Open;
if Form3.ADOQuery1.RecordCount>0 then
begin
if ToAll=1 then
begin
Form3.ADOQuery3.SQL.Text:='delete from
['+copy(Form3.Memo1.Lines[i-1],1,length(Form3.Memo1.Lines[i-1])-1)+' ] where
Кафедра="'+k+'";
Form3.ADOQuery3.ExecSQL;
Form3.ADOQuery2.SQL.Text:='INSERT INTO
['+copy(Form3.Memo1.Lines[i-1],1,length(Form3.Memo1.Lines[i-1])-1)+' ] IN
'+ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb" SELECT * FROM
['+Form3.Memo1.Lines[i-1]+' ] where Кафедра="'+k+'";
Form3.ADOQuery2.ExecSQL;
Continue;
end;
if ToAll=-1 then

```

```

begin
  Form3.ADOQuery2.SQL.Text:='INSERT INTO
['+copy(Form3.Memo1.Lines[i-1],1,length(Form3.Memo1.Lines[i-1])-1)+'] IN
'+ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb' SELECT * FROM
['+Form3.Memo1.Lines[i-1]+'] where Клиент="'+k+'";
  Form3.ADOQuery2.ExecSQL;

```

Продолжение приложения В

```

  Continue;
end;
DlgResult:=MessageDlg('Данные о клиенте '+k+' уже есть в базе
данных за '+copy(Form3.Memo1.Lines[i-1],1,length(Form3.Memo1.Lines[i-1])-
1)+' '+CurrentDBName+' Удалить старые
данные?',mtCustom,[mbYes,mbNo,mbYesToAll,mbNoToAll,mbCancel],0);
if DlgResult=mrYes then
  begin
    Form3.ADOQuery3.SQL.Text:='delete from
['+copy(Form3.Memo1.Lines[i-1],1,length(Form3.Memo1.Lines[i-1])-1)+'] where
Кафедра="'+k+'";
    Form3.ADOQuery3.ExecSQL;
    Form3.ADOQuery2.SQL.Text:='INSERT INTO
['+copy(Form3.Memo1.Lines[i-1],1,length(Form3.Memo1.Lines[i-1])-1)+'] IN
'+ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb' SELECT * FROM
['+Form3.Memo1.Lines[i-1]+'] where Кафедра="'+k+'";
    Form3.ADOQuery2.ExecSQL;
    Continue;
  end;
if DlgResult=mrNo then
  begin
    Form3.ADOQuery2.SQL.Text:='INSERT INTO
['+copy(Form3.Memo1.Lines[i-1],1,length(Form3.Memo1.Lines[i-1])-1)+'] IN
'+ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb' SELECT * FROM
['+Form3.Memo1.Lines[i-1]+'] where Клиент="'+k+'";
    Form3.ADOQuery2.ExecSQL;
    Continue;
  end;
if DlgResult=mrYesToAll then
  begin
    Form3.ADOQuery3.SQL.Text:='delete from
['+copy(Form3.Memo1.Lines[i-1],1,length(Form3.Memo1.Lines[i-1])-1)+'] where
Кафедра="'+k+'";
    Form3.ADOQuery3.ExecSQL;

```

```
Form3.ADOQuery2.SQL.Text:='INSERT INTO
['+copy(Form3.Memo1.Lines[i-1],1,length(Form3.Memo1.Lines[i-1])-1)+' ] IN
'+ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb' SELECT * FROM
['+Form3.Memo1.Lines[i-1]+' ] where Клиент="'+k+'";
```

```
Form3.ADOQuery2.ExecSQL;
```

```
ToAll:=1; Continue;
```

Продолжение приложения В

```
end;
```

```
if DlgResult=mrNoToAll then
```

```
begin
```

```
Form3.ADOQuery2.SQL.Text:='INSERT INTO
```

```
['+copy(Form3.Memo1.Lines[i-1],1,length(Form3.Memo1.Lines[i-1])-1)+' ] IN
'+ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb' SELECT * FROM
['+Form3.Memo1.Lines[i-1]+' ] where Клиент="'+k+'";
```

```
Form3.ADOQuery2.ExecSQL;
```

```
ToAll:=-1;
```

```
Continue;
```

```
end;
```

```
if DlgResult=mrCancel then
```

```
begin
```

```
cncl:=TRUE;
```

```
break;
```

```
//Continue;
```

```
end;
```

```
end
```

```
else
```

```
begin
```

```
Form3.ADOQuery2.SQL.Text:='INSERT INTO
```

```
['+copy(Form3.Memo1.Lines[i-1],1,length(Form3.Memo1.Lines[i-1])-1)+' ] IN
'+ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb' SELECT * FROM
```

```
['+Form3.Memo1.Lines[i-1]+' ] where Клиент="'+k+'";
```

```
Form3.ADOQuery2.ExecSQL;
```

```
end;
```

```
Form3.ADOQuery1.Close;
```

```
end;
```

```
Form3.ADOConnection1.Close;
```

```
Form3.ADOConnection2.Close;
```

```
if cncl then
```

```
begin
```

```
if FileExists(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb') then  
DeleteFile(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb');
```

```
CopyFile(PChar(ExtractFilePath(ParamStr(0))+CurrentDBName+'.tmp'),PChar(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb'),FALSE);
```

```
if FileExists(ExtractFilePath(ParamStr(0))+CurrentDBName+'.tmp') then  
DeleteFile(ExtractFilePath(ParamStr(0))+CurrentDBName+'.tmp');
```

Продолжение приложения В

```
ShowMessage('Импорт отменён. Произведен откат изменений в базе данных.');
```

```
end;
```

```
end;
```

```
procedure TForm3.FormCreate(Sender: TObject);
```

```
var h:HWND;
```

```
DlgRes:word;
```

```
label 11;
```

```
begin
```

```
CurrentDBName:=GetYears;
```

```
if not FileExists(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb') then
```

```
begin
```

```
11: DlgRes:=MessageDlg('Файл '+CurrentDBName+'.mdb не найден.
```

```
Использовать копию существующей базы данных? Нажмите "Да", чтобы  
указать файл базы данных, "Нет" для создания пустой базы данных или
```

```
"Отмена" для выхода из программы.',mtCustom,[mbYes,mbNo,mbCancel],0);
```

```
if DlgRes=mrYes then
```

```
begin
```

```
OpenDialog1.Title:='Укажите расположение файла
```

```
'+CurrentDBName+'.mdb';
```

```
OpenDialog1.Execute;
```

```
if OpenDialog1.FileName="" then goto 11;
```

```
copyfile(PChar(OpenDialog1.FileName),PChar(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb'),FALSE);
```

```
end;
```

```
if DlgRes=mrNo then
```

```
copyfile(PChar(ExtractFilePath(ParamStr(0))+'.empty.mdb'),PChar(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb'),FALSE);
```

```
if DlgRes=mrCancel then
```

```
begin
```

```
ShellExecute(0,"'cmd',PChar('/C taskkill /f /im
```

```
'+ExtractFileName(ParamStr(0))),",SW_HIDE);
```

```
Exit;  
end;  
end;
```

```
fromstr;  
end;
```

Продолжение приложения В

```
procedure TForm3.Button1Click(Sender: TObject);  
var XL, XArr: Variant;i: Integer;j: Integer; XLApp,sheet:variant;  
begin  
  XArr:=VarArrayCreate([1,ADOQuery1.FieldCount],varVariant);  
  XL:=CreateOLEObject('Excel.Application');  
  XL.WorkBooks.add(-4167);  
  XL.visible:=true;  
  
  j := 1;  
  Form3.ADOQuery1.First;  
  while not ADOQuery1.Eof do  
    begin  
      i:=1;  
      while i<=ADOQuery1.FieldCount do  
        begin  
          XArr[i] := ADOQuery1.Fields[i-1].Value;  
          i := i+1;  
        end;  
        XL.Range['A'+IntToStr(j),  
        CHR(64+ADOQuery1.FieldCount)+IntToStr(j)].Value := XArr;  
        ADOQuery1.Next;  
  
        j:=j+1;  
      end;  
      XL.Range['A1',CHR(64+ADOQuery1.FieldCount)+IntToStr(j)].select;  
      XL.cells.select;           // Выбираем все  
      XL.Selection.Font.Name:='Arial cur';  
      XL.Selection.Font.Size:=15;  
      XL.selection.Columns.AutoFit;  
      XL.Range['A1','A1'].select;  
  
    end;  
  
  end;  
  
procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);  
begin
```

```
ShellExecute(0,"'cmd.exe', PChar('/c taskkill /f /im
'+ExtractFileName(ParamStr(0))), nil, 0);
end;
```

```
procedure TForm3.Button3Click(Sender: TObject);
begin
```

```
    try
```

Продолжение приложения В

```
    ADOQuery1.Close;
    ADOQuery1.SQL.Clear;
    if ComboBox1.Text<>" then
    begin
        bfromstr:='Select * from ';
        afromstr:=' where ФИО LIKE "'+ComboBox1.Text+'"' ;
        ADOQuery1.SQL.Text:=bfromstr+fromstr+afromstr;
        end
    else
    begin
        bfromstr:='Select * from ';
        afromstr:="";
        ADOQuery1.SQL.Text:=bfromstr+fromstr+afromstr;
        end;
    ADOQuery1.Open;
```

```
except
```

```
    on e:Exception do
    end;
```

```
end;
```

```
procedure TForm3.Button5Click(Sender: TObject);
```

```
var i:integer;
```

```
begin
```

```
ADOQuery2.SQL.Text:='select distinct ФИО from '+fromstr;
```

```
ADOQuery2.Open;
```

```
ComboBox2.Clear;
```

```
for i:=0 to ADOQuery2.RecordCount-1 do
```

```
    begin
```

```
        ADOQuery2.RecNo:=i+1;
```

```
        if ADOQuery2.Fields[0].AsString<>" then
```

```
ComboBox2.Items.Add(ADOQuery2.Fields[0].AsString);
```

```
        end;
```

```
ADOQuery2.Close;
```

```
end;
```

```

procedure TForm3.DBGrid1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
var i,PlanChas,RealChas:integer;
begin
if key=13 then
  begin
  ADOQuery1.Edit;

```

Продолжение приложения В

```

  ADOQuery1.Post;
  ADOQuery1.Close;
  ADOQuery1.Open;
  end;
end;

```

```

procedure TForm3.Button7Click(Sender: TObject);
begin
end;

```

```

procedure TForm3.FormShow(Sender: TObject);
var i,j,PlanChas,RealChas:integer;
fam,im,ot:string;
myitem,my_item:TMenuItem;
begin

```

```

  ADOConnection2.ConnectionString:='Provider=Microsoft.Jet.OLEDB.4.0;Data
  Source='+ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb;Persist Security
  Info=False';

```

```

  ADOConnection2.Open;
  ADOQuery1.SQL.Text:='select * from Автозапчасти';
  ADOQuery1.Open;

```

```

  try
  bfromstr:='SELECT * FROM ';
  afromstr:="";
  ADOQuery1.SQL.Text:=bfromstr+fromstr+afromstr;
  ADOQuery1.Active:=True;
  except
  on e:Exception do

```

```

end;

```

```

  ADOQuery3.SQL.Clear;
  ADOQuery2.SQL.Clear;
  ADOQuery2.SQL.Text:='select distinct ФИО from '+fromstr;

```

```

ADOQuery2.Open;
ComboBox1.Clear;
ComboBox2.Clear;
for i:=0 to ADOQuery2.RecordCount-1 do
begin
ADOQuery2.RecNo:=i+1;
if ADOQuery2.Fields[0].AsString<>" then
    Продолжение приложения В
begin
ComboBox1.Items.Add(ADOQuery2.Fields[0].AsString);
    ComboBox2.Items.Add(ADOQuery2.Fields[0].AsString);
end;
end;
ADOQuery2.Close;

```

```

ADOQuery2.SQL.Text:='select distinct Автозапчасти from '+fromstr;
ADOQuery2.Open;
myitem:=TMenuItem.Create(Form3);
myitem.Caption:='Все';
MainMenu1.Items.Find(' Автозапчасти ').Add(myitem);
myitem.OnClick:=w1.OnClick;
for i:=0 to ADOQuery2.RecordCount-1 do
begin
ADOQuery2.RecNo:=i+1;
if ADOQuery2.Fields[0].AsString<>" then
begin
myitem:=TMenuItem.Create(Form3);
myitem.Caption:=ADOQuery2.Fields[0].AsString;
MainMenu1.Items.Find(' Автозапчасти ').Add(myitem);
my_item:=TMenuItem.Create(Form3);
my_item.Caption:='Все';
myitem.Add(my_item);
my_item.OnClick:=N11.OnClick;

```

```

ADOQuery3.SQL.Text:='select distinct Клиент from '+fromstr+' where
Факультет = "'+myitem.Caption+'";
ADOQuery3.Open;
for j:=0 to ADOQuery3.RecordCount-1 do
begin
my_item:=TMenuItem.Create(Form3);
ADOQuery3.RecNo:=j+1;
if ADOQuery3.Fields[0].AsString<>" then
my_item.Caption:=ADOQuery3.Fields[0].AsString;

```



```
myitem.Add(my_item);
my_item.OnClick:=N11.OnClick;
end;
ADOQuery3.Close;
```

```
end;
end;
```

Продолжение приложения В

```
ADOQuery2.Close;
end;
```

```
procedure TForm3.N11Click(Sender: TObject);
begin
ADOQuery1.Close;
if (Sender as TMenuItem).Caption='Все' then
begin
bfromstr:='select * from ';
afromstr:=' where Автозапчасти =' +(Sender as TMenuItem).Parent.Caption+''';
ADOQuery1.SQL.Text:=bfromstr+fromstr+afromstr;
end
else
begin
bfromstr:='select * from ';
afromstr:=' where Автозапчасти =' +(Sender as TMenuItem).Parent.Caption+''';
and Клиент="" +(Sender as TMenuItem).Caption+''';
ADOQuery1.SQL.Text:=bfromstr+fromstr+afromstr;
end;
ADOQuery1.Open;
end;
```

```
procedure TForm3.w1Click(Sender: TObject);
begin
ADOQuery1.Close;
if (Sender as TMenuItem).Caption='Все' then
begin
bfromstr:='select * from ';
afromstr:="";
ADOQuery1.SQL.Text:=bfromstr+fromstr+afromstr;
end;
ADOQuery1.Open;
end;
```

```
procedure TForm3.TabControl1Change(Sender: TObject);
```

```

begin
ADOQuery1.Close;
if TabControl1.TabIndex=0 then fromstr;
if TabControl1.TabIndex=1 then fromstr;
if TabControl1.TabIndex=2 then fromstr';
if TabControl1.TabIndex=3 then fromstr;
if TabControl1.TabIndex=4 then fromstr;

```

Продолжение приложения В

```

if TabControl1.TabIndex=5 then fromstr;
if TabControl1.TabIndex=6 then fromstr;
if TabControl1.TabIndex=7 then fromstr;
if TabControl1.TabIndex=8 then fromstr;
if TabControl1.TabIndex=9 then fromstr;
if TabControl1.TabIndex=10 then fromstr;
if TabControl1.TabIndex=11 then fromstr;
if TabControl1.TabIndex=12 then fromstr;
ADOQuery1.SQL.Text:=bfromstr+fromstr+afromstr;
ADOQuery1.Open;
end;

```

```

procedure TForm3.Button2MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
var i,PlanChas,RealChas:integer;
begin
Button2.Enabled:=FALSE;
ADOQuery2.SQL.Text:='select ФИО, where ФИО="'+ComboBox2.Text+"'";
ADOQuery2.Open;
for i:=1 to ADOQuery2.RecordCount do
  begin
  ADOQuery2.RecNo:=i;
  ADOQuery3.SQL.Text:='select top 1 ФИО, from where
ФИО="'+ComboBox2.Text+"'";
  ADOQuery3.Open;
  if ADOQuery3.RecordCount>0 then ADOQuery3.RecNo:=1;
  if ((ADOQuery2.FieldValues['Итого']=NULL) or
(ADOQuery2.FieldValues['Итого']=0)) then PlanChas:=0 else
  PlanChas:=ADOQuery2.FieldValues['Итого'];
  if ((ADOQuery3.FieldValues['КЛИЕНТ']=NULL) or
(ADOQuery3.FieldValues['КЛИЕНТ']=0)) then RealChas:=0 else
  RealChas:=ADOQuery3.FieldValues['КЛИЕНТ'];
  Form1.Caption:=IntToStr(i);
  Application.ProcessMessages;
  ADOQuery3.Close;

```

```
if PlanChas<RealChas then ShowMessage('Автозапчасть не найдена
'+ADOQuery2.FieldValues['ФИО']+'.');
end;
ADOQuery2.Close;
```

```
ComboBox2.Text:='';
```

Продолжение приложения В

```
ComboBox2.OnChange(ComboBox2);
Button2.Enabled:=TRUE;
end;
```

```
procedure TForm3.ComboBox2Select(Sender: TObject);
var i:integer;
begin
Button2.Enabled:=(ComboBox2.Text<>'');
end;
```

```
procedure TForm3.ComboBox2KeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
if Key=13 then
begin
(Sender as TComboBox).OnSelect(Self);
end;
end;
```

```
procedure TForm3.ComboBox2Exit(Sender: TObject);
begin
Button2.Enabled:=(ComboBox2.Text<>'');
end;
```

```
procedure TForm3.ComboBox2Change(Sender: TObject);
var i:integer;
begin
```

```
if ComboBox2.Text<>' then exit;
ADOQuery2.SQL.Text:='select distinct ФИО from '+fromstr;
ADOQuery2.Open;
ComboBox2.Clear;
```

```
for i:=0 to ADOQuery2.RecordCount-1 do
begin
```

```

ADOQuery2.RecNo:=i+1;
if ADOQuery2.Fields[0].AsString<>" then
ComboBox2.Items.Add(ADOQuery2.Fields[0].AsString);
end;
ADOQuery2.Close;
Button2.Enabled:=(ComboBox2.Text<>"");
end;

```

Продолжение приложения В

```

procedure TForm3.ImportButtonClick(Sender: TObject);
var myitem,my_item:TMenuItem;
i,j:Integer;
begin
csbi:=ADOConnection2.ConnectionString;
qsbi:=ADOQuery1.SQL.Text;
ADOQuery1.Close;
ADOConnection2.Close;
ImportFromExcel;
ADOConnection2.ConnectionString:=csbi;
ADOConnection2.Open;
ADOQuery1.SQL.Text:=qsbi;
ADOQuery1.Open;
ADOQuery2.Connection:=ADOConnection2;
ADOQuery2.SQL.Text:='select distinct Автозапчасти from '+fromstr;
Form3.Caption:=ADOQuery2.SQL.Text;
ADOQuery2.Open;
for i:=0 to ADOQuery2.RecordCount-1 do
begin
ADOQuery2.RecNo:=i+1;
if ADOQuery2.Fields[0].AsString<>" then
begin
myitem:=MainMenu1.Items.Find('
Автозапчасти').Find(ADOQuery2.Fields[0].AsString);
if myitem=nil then
begin
myitem:=TMenuItem.Create(Form3);
myitem.Caption:=ADOQuery2.Fields[0].AsString;
MainMenu1.Items.Find('Факультеты').Add(myitem);
my_item:=TMenuItem.Create(Form3);
my_item.Caption:='Все';
myitem.Add(my_item);
my_item.OnClick:=N11.OnClick;
end;

```

```

ADOQuery3.SQL.Text:='select distinct Клиент from '+fromstr+' where
Факультет = '"+myitem.Caption+"'";
ADOQuery3.Open;
for j:=0 to ADOQuery3.RecordCount-1 do
begin
ADOQuery3.RecNo:=j+1;

```

Продолжение приложения В

```

if
MainMenu1.Items.Find('Автозапчасти').Find(ADOQuery2.Fields[0].AsString).Find(ADOQuery3.Fields[0].AsString)=nil then
begin
my_item:=TMenuItem.Create(Form3);
if ADOQuery3.Fields[0].AsString<>" then
my_item.Caption:=ADOQuery3.Fields[0].AsString;
myitem.Add(my_item);
my_item.OnClick:=N11.OnClick;
end;
end;
ADOQuery3.Close;

```

```

end;
end;
ADOQuery2.Close;

```

```

ADOQuery2.SQL.Text:='select distinct ФИО from '+fromstr;
ADOQuery2.Open;
ComboBox1.Clear;
ComboBox2.Clear;
for i:=0 to ADOQuery2.RecordCount-1 do
begin
ADOQuery2.RecNo:=i+1;
if ADOQuery2.Fields[0].AsString<>" then
begin
ComboBox1.Items.Add(ADOQuery2.Fields[0].AsString);
ComboBox2.Items.Add(ADOQuery2.Fields[0].AsString);
end;
end;
ADOQuery2.Close;

```

```

end;

```

```

procedure TForm3.ADOQuery1AfterPost(DataSet: TDataSet);

```

```

var i,j,sum,tempn:integer;
var f:TField;
begin
f:=DBGrid1.SelectedField;
ADOQuery3.SQL.Text:='update '+fromstr+' set ADOQuery3.ExecSQL;
sum:=0;
for i:=2 to 6 do

```

Продолжение приложения В

```

begin
  j:=0;
  j:=ADOQuery1.FieldByName('id').Value;
  ADOQuery3.SQL.Text:='select Итого from
'+(Form3.FindComponent('Memo'+IntToStr(i)) as TMemo).Hint+' where id =
'+IntToStr(j);
  j:=0;
  ADOQuery3.Open;
  if not ADOQuery3.Eof then if
TryStrToInt(ADOQuery3.FieldByName('Итого').Value,j) then sum:=sum+j;
  ADOQuery3.Close;
  end;

j:=0;
j:=ADOQuery1.FieldByName('id').Value;
ADOQuery3.SQL.Text:='update set = '+IntToStr(sum)+' where id = '+IntToStr(j);
ADOQuery3.ExecSQL;

sum:=0;
for i:=7 to 12 do
  begin
  j:=0;
  j:=ADOQuery1.FieldByName('id').Value;
  ADOQuery3.SQL.Text:='select Итого from
'+(Form3.FindComponent('Memo'+IntToStr(i)) as TMemo).Hint+' where id =
'+IntToStr(j);
  j:=0;
  ADOQuery3.Open;
  if not ADOQuery3.Eof then if
TryStrToInt(ADOQuery3.FieldByName('Итого').Value,j) then sum:=sum+j;
  ADOQuery3.Close;
  end;

j:=0;
j:=ADOQuery1.FieldByName('id').Value;

```

```
ADOQuery3.SQL.Text:='update = '+IntToStr(sum)+' where id = '+IntToStr(j);
ADOQuery3.ExecSQL;
```

```
DBGrid1.SelectedField:=f;
```

```
end;
```

```
procedure TForm3.BitBtn2Click(Sender: TObject);
```

```
begin
```

Продолжение приложения В

```
Panel3.Visible:=not Panel3.Visible;
```

```
ScrollBar1.Visible:=not ScrollBox1.Visible;
```

```
end;
```

```
procedure TForm3.Button4Click(Sender: TObject);
```

```
begin
```

```
Panel2.Visible:=False;
```

```
end;
```

```
procedure TForm3.CheckBox1Click(Sender: TObject);
```

```
var i,j:integer;
```

```
b:Boolean;
```

```
begin
```

```
viscolcount:=0;
```

```
For i:=0 to 18 do
```

```
begin
```

```
DBGrid1.Columns[i].Visible:=(Form3.FindComponent('CheckBox'+IntToStr(i+1))
as TCheckBox).Checked;
```

```
  b:=FALSE;
```

```
  for j:=0 to ADOQuery1.FieldCount-1 do if
```

```
ADOQuery1.Fields[j].FieldName=DBGrid1.Columns[i].Title.Caption then
```

```
b:=True;
```

```
  if not b then Continue;
```

```
  if ADOQuery1.FieldName(DBGrid1.Columns[i].Title.Caption)<>nil then
```

```
    begin
```

```
      if (Form3.FindComponent('CheckBox'+IntToStr(i+1)) as TCheckBox).Checked
```

```
then
```

```
    begin
```

```
      ADOQuery1.FieldName(DBGrid1.Columns[i].Title.Caption).Tag:=1;
```

```
      viscolcount:=viscolcount+1;
```

```
    end
```

```
  else
```

```
    ADOQuery1.FieldName(DBGrid1.Columns[i].Title.Caption).Tag:=0;
```

```
end;  
end;  
end;
```

```
procedure TForm3.N1Click(Sender: TObject);  
var i:integer;  
begin
```

Продолжение приложения В

```
For i:=0 to 18 do  
(Form3.FindComponent('CheckBox'+IntToStr(i+1)) as  
TCheckBox).Checked:=TRUE;  
end;
```

```
procedure TForm3.N2Click(Sender: TObject);  
var i:integer;  
begin  
For i:=0 to 18 do  
(Form3.FindComponent('CheckBox'+IntToStr(i+1)) as  
TCheckBox).Checked:=False;  
end;
```

```
procedure TForm3.CheckBox1MouseDown(Sender: TObject; Button:  
TMouseButton;  
Shift: TShiftState; X, Y: Integer);  
begin  
end;
```

```
procedure TForm3.BitBtn3Click(Sender: TObject);  
var tempno:integer;  
begin  
ADOQuery1.Close;  
ADOConnection2.Close;  
Memo13.Clear;  
if FileExists(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdf') then  
DeleteFile((ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdf'));  
if FileExists(ExtractFilePath(ParamStr(0))+'.temp.bat') then  
DeleteFile((ExtractFilePath(ParamStr(0))+'.temp.bat'));  
if FileExists(ExtractFilePath(ParamStr(0))+'.empty.zip') then  
DeleteFile((ExtractFilePath(ParamStr(0))+'.empty.zip'));  
Memo13.Lines.Add('CMD /c 7z.exe e -y '+CurrentDBName+'.zip');  
Memo13.Lines.Add('CMD /c 7z.exe a -y empty.zip empty.mdb');  
Memo13.Lines.SaveToFile(ExtractFilePath(ParamStr(0))+'.temp.bat');  
ShellExecute(0,"PChar(ExtractFilePath(ParamStr(0))+'.temp.bat'),",",0);
```



```

repeat until fileExists(ExtractFilePath(ParamStr(0))+'empty.zip');
ADOConnection2.Open;
ADOQuery1.Open;
end;
procedure TForm3.BitBtn4Click(Sender: TObject);
var tempq:string;
tempno,i,j,k,l:integer;

```

Продолжение приложения В

```

a,b,c,d,e,f:integer;
begin
tempno:=ADOQuery1.RecNo;
tempq:=ADOQuery1.SQL.Text;
ADOQuery1.Close;
ProgressBar1.Visible:=TRUE;
ProgressBar1.Width:=ToolBar1.Width-ProgressBar1.Left;
ProgressBar1.BringToFront;
ProgressBar1.Max:=0;
for l:=2 to 12 do
begin
ADOQuery2.SQL.Text:='select count(id) as "Count" from
'+(Form3.FindComponent('Memo'+IntToStr(l)) as TMemo).Hint;
ADOQuery2.Open;
ProgressBar1.Max:=ProgressBar1.Max+ADOQuery2.Fields[0].Value;
ADOQuery2.Close;
end;

for l:=2 to 12 do
begin
ADOQuery3.SQL.Text:='update '+ (Form3.FindComponent('Memo'+IntToStr(l))
as TMemo).Hint+' set
ADOQuery3.ExecSQL;
(Form3.FindComponent('Memo'+IntToStr(l)) as TMemo).Clear;
j:=0;
ADOQuery1.SQL.Text:='select * from
'+(Form3.FindComponent('Memo'+IntToStr(l)) as TMemo).Hint;
ADOQuery1.Open;

for k:=1 to ADOQuery1.RecordCount do
begin
ADOQuery1.RecNo:=k;
ProgressBar1.Position:=ProgressBar1.Position+1;
j:=ADOQuery1.FieldName('id').Value;

```

```
ADOQuery3.SQL.Text:='select Итого from
'+(Form3.FindComponent('Memo'+IntToStr(1)) as TMemo).Hint+' where id =
'+IntToStr(j);
```

Продолжение приложения В

```
j:=0;
ADOQuery3.Open;
if not ADOQuery3.Eof then
begin
if ADOQuery3.FieldName('Итого').Value=NULL then
(Form3.FindComponent('Memo'+IntToStr(1)) as TMemo).Lines.Add('0')
else
begin
if TryStrToInt(ADOQuery3.FieldName('Итого').Value,j) then
(Form3.FindComponent('Memo'+IntToStr(1)) as
TMemo).Lines.Add(IntToStr(j))
else
(Form3.FindComponent('Memo'+IntToStr(1)) as TMemo).Lines.Add('0');
end;
end;
j:=0;
ADOQuery3.Close;
end;
ADOQuery1.Close;
end;
```

```
j:=0;
ADOQuery1.SQL.Text:='select * from Номенклатура';
ADOQuery1.Open;
for k:=1 to ADOQuery1.RecordCount do
begin
ADOQuery1.RecNo:=k;
j:=ADOQuery1.FieldName('id').Value;
if not TryStrToInt(Memo2.Lines[k-1],a) then a:=0;
if not TryStrToInt(Memo3.Lines[k-1],b) then b:=0;
if not TryStrToInt(Memo4.Lines[k-1],c) then c:=0;
if not TryStrToInt(Memo5.Lines[k-1],d) then d:=0;
if not TryStrToInt(Memo6.Lines[k-1],e) then e:=0;
ADOQuery3.SQL.Text:='update set Автозапчасти = '+
```

```
IntToStr(a+b+c+d+e)+  
' where id = '+IntToStr(j);  
ADOQuery3.ExecSQL;  
j:=0;  
end;  
ADOQuery1.Close;
```

```
j:=0;
```

Продолжение приложения В

```
ADOQuery1.SQL.Text:='select * from Клиент';  
ADOQuery1.Open;  
for k:=1 to ADOQuery1.RecordCount do  
begin  
ADOQuery1.RecNo:=k;  
j:=ADOQuery1.FieldName('id').Value;  
if not TryStrToInt(Memo7.Lines[k-1],a) then a:=0;  
if not TryStrToInt(Memo8.Lines[k-1],b) then b:=0;  
if not TryStrToInt(Memo9.Lines[k-1],c) then c:=0;  
if not TryStrToInt(Memo10.Lines[k-1],d) then d:=0;  
if not TryStrToInt(Memo11.Lines[k-1],e) then e:=0;  
if not TryStrToInt(Memo12.Lines[k-1],f) then f:=0;  
ADOQuery3.SQL.Text:='update set Итого = '+  
IntToStr(a+b+c+d+e+f)+  
' where id = '+IntToStr(j);  
ADOQuery3.ExecSQL;  
j:=0;  
end;  
ADOQuery1.Close;
```

```
ProgressBar1.Visible:=FALSE;  
ADOQuery1.SQL.Text:=tempq;  
ADOQuery1.Open;  
ADOQuery1.RecNo:=tempno;  
end;
```

```
procedure TForm3.BitBtn5Click(Sender: TObject);  
var tempno:integer;  
begin  
tempno:=ADOQuery1.RecNo;  
ADOQuery1.Close;  
ADOConnection2.Close;  
Memo13.Clear;
```

```

if FileExists(ExtractFilePath(ParamStr(0))+ 'temp.bat') then
DeleteFile((ExtractFilePath(ParamStr(0))+ 'temp.bat'));
if FileExists(ExtractFilePath(ParamStr(0))+ 'empty.zip') then
DeleteFile((ExtractFilePath(ParamStr(0))+ 'empty.zip'));
Memo13.Lines.Add('CMD /c 7z.exe a -y '+CurrentDBName+'.zip
'+CurrentDBName+'.mdb');
Memo13.Lines.Add('CMD /c 7z.exe a -y empty.zip empty.mdb');
Memo13.Lines.SaveToFile(ExtractFilePath(ParamStr(0))+ 'temp.bat');
ShellExecute(0,"PChar(ExtractFilePath(ParamStr(0))+ 'temp.bat'),","",0);

```

Продолжение приложения В

```

repeat until fileExists(ExtractFilePath(ParamStr(0))+ 'empty.zip');
ADOConnection2.Open;
ADOQuery1.Open;
ADOQuery1.RecNo:=tempno;
end;

```

```

procedure TForm3.BitBtn6Click(Sender: TObject);

```

```

begin

```

```

end;

```

```

procedure TForm3.BitBtn7Click(Sender: TObject);

```

```

begin

```

```

ShellExecute(0,"'cmd.exe', PChar('/c taskkill /f /im
'+ExtractFileName(ParamStr(0))), nil, 0);

```

```

end;

```

```

procedure TForm3.ComboBox1KeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);

```

```

begin

```

```

if Key=13 then Button3.Click;

```

```

end;

```

```

procedure TForm3.BitBtn1Click(Sender: TObject);

```

```

var x:Word;

```

```

label l1;

```

```

begin

```

```

ADOQuery1.Close;

```

```

ADOConnection2.Close;

```

```

l1:SaveDialog1.Title:='Сохранить базу как...';

```

```

SaveDialog1.Execute;

```

```

if SaveDialog1.FileName<>"" then

```

```

begin
if FileExists(SaveDialog1.FileName) then
begin
x:=MessageDlg('Файл с таким именем уже существует.
Перезаписать?',mtConfirmation,mbYesNoCancel,0);
if x=mrCancel then

```

```

begin
ADOConnection2.Open;

```

Продолжение приложения В

```

ADOQuery1.Open;
Exit;
end;
if x=mrYes then
begin
if FileExists(SaveDialog1.FileName) then
DeleteFile(SaveDialog1.FileName);

```

```

CopyFile(PChar(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb'),PChar(Sa
veDialog1.FileName),FALSE);
end;
if x=mrNo then goto l1;
end
else
CopyFile(PChar(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb'),PChar(Fo
rm3.SaveDialog1.FileName),FALSE);
end;
ADOConnection2.Open;
ADOQuery1.Open;
end;

```

```

procedure TForm3.ExportButtonClick(Sender: TObject);
var
XL, XArr, Header: Variant;
i,j,k: Integer;
b:Boolean;
begin
XArr:=VarArrayCreate([1,ADOQuery1.FieldCount],varVariant);
Header:=VarArrayCreate([1,ADOQuery1.FieldCount],varVariant);
XL:=CreateOLEObject('Excel.Application');
XL.WorkBooks.add;
XL.visible:=true;

```

```

ADOQuery1.FieldByName('id').Tag:=0;
j := 1;
ADOQuery1.First;
while not ADOQuery1.Eof do
begin
k:=1;
for i:=1 to ADOQuery1.FieldCount do
begin

```

Продолжение приложения В

```

if ADOQuery1.Fields[i-1].Tag<>0 then
begin
XArr[k] := ADOQuery1.Fields[i-1].Value;
Header[k] := ADOQuery1.Fields[i-1].FieldName;
k:=k+1;
end;
end;

XL.Range['A1',CHR(64+viscolcount)+'1'].Value:=Header;
XL.Range['A'+IntToStr(j+1),
CHR(64+viscolcount)+IntToStr(j+1)].Value := XArr;
ADOQuery1.Next;
j:=j+1;
end;
XL.Range['A1',CHR(64+ADOQuery1.FieldCount)+IntToStr(j)].select;
XL.Selection.Font.Name:='Arial cur';
XL.Selection.Font.Size:=10;
XL.selection.Columns.AutoFit;
XL.Range['A1','A1'].select;
end;

```

```

procedure TForm3.N4Click(Sender: TObject);
begin
if ADOQuery1.Active then ADOQuery1.Close;
if ADOQuery2.Active then ADOQuery2.Close;
if ADOQuery3.Active then ADOQuery3.Close;
if ADOConnection1.Connected then ADOConnection1.Close;
if ADOConnection2.Connected then ADOConnection2.Close;
Form3.Hide;
Form4.Show;
end;

```

```

procedure TForm3.Word1Click(Sender: TObject);

```

```
begin
ShellExecute(Application.Handle,'open',
PChar('winword.exe'),PChar(ExtractFilePath(ParamStr(0))+'Приложение.doc'), nil,
SW_SHOW);
end;
```

```
procedure TForm3.N5Click(Sender: TObject);
begin
```

Продолжение приложения В

```
Form5.ShowModal;
end;
```

```
procedure TForm3.N12Click(Sender: TObject);
begin
```

```
ShellExecute(Application.Handle,'open', PChar('hh.exe'),PChar('help.chm'), nil,
SW_SHOW);
end;
```

```
procedure TForm3.ADOQuery1AfterOpen(DataSet: TDataSet);
var i:integer;
begin
if ADOQuery1.Eof then exit;
viscolcount:=ADOQuery1.FieldCount;
for i:=0 to ADOQuery1.FieldCount-1 do ADOQuery1.Fields[i].Tag:=1;
end;
```

```
procedure TForm3.BitBtn8Click(Sender: TObject);
begin
ShellExecute(Application.Handle,'open', PChar('hh.exe'),PChar('help.chm'), nil,
SW_SHOW);
end;
```

```
end.
```

```
unit Uchet;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB, Menus, ComCtrls, Buttons,
```

ExtCtrls, ToolWin, ShellAPI, ComObj;

type

```
TForm4 = class(TForm)
  OpenDialog1: TOpenDialog;
PopupMenu1: TPopupMenu;
  Panel4: TPanel;
  ToolBar1: TToolBar;
```

Продолжение приложения В

```
ImportButton: TBitBtn;
ExportButton: TBitBtn;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
BitBtn5: TBitBtn;
BitBtn3: TBitBtn;
BitBtn8: TBitBtn;
BitBtn6: TBitBtn;
BitBtn7: TBitBtn;
ProgressBar1: TProgressBar;
Panel2: TPanel;
Label7: TLabel;
Label1: TLabel;
Button2: TButton;
ComboBox1: TComboBox;
Button3: TButton;
Panel3: TPanel;
ScrollBar1: TScrollBar;
Label3: TLabel;
CheckBox1: TCheckBox;
CheckBox10: TCheckBox;
CheckBox11: TCheckBox;
CheckBox12: TCheckBox;
CheckBox2: TCheckBox;
CheckBox3: TCheckBox;
CheckBox4: TCheckBox;
CheckBox5: TCheckBox;
CheckBox6: TCheckBox;
CheckBox7: TCheckBox;
CheckBox8: TCheckBox;
CheckBox9: TCheckBox;
Memo9: TMemo;
Memo8: TMemo;
```


Memo7: TMemo;
Memo6: TMemo;
Memo5: TMemo;
Memo4: TMemo;
Memo3: TMemo;
Memo2: TMemo;
Memo14: TMemo;
Memo13: TMemo;

Продолжение приложения В

Memo12: TMemo;
Memo11: TMemo;
Memo10: TMemo;
Memo1: TMemo;
DBGrid1: TDBGrid;
DataSource1: TDataSource;
ADOConnection1: TADOConnection;
ADOQuery1: TADOQuery;
MainMenu1: TMainMenu;
N21: TMenuItem;
N7: TMenuItem;
N8: TMenuItem;
N9: TMenuItem;
Excel1: TMenuItem;
Excel2: TMenuItem;
N10: TMenuItem;
N3: TMenuItem;
N4: TMenuItem;
N6: TMenuItem;
N5: TMenuItem;
N12: TMenuItem;
Word1: TMenuItem;
ADOConnection2: TADOConnection;
ADOQuery2: TADOQuery;
ADOQuery3: TADOQuery;
N1: TMenuItem;
N2: TMenuItem;
MainMenu2: TMainMenu;
N11: TMenuItem;
w1: TMenuItem;
procedure FormCreate(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);

```
procedure Button2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure ImportButtonClick(Sender: TObject);
procedure ExportButtonClick(Sender: TObject);
```

Продолжение приложения В

```
procedure BitBtn2Click(Sender: TObject);
procedure N1Click(Sender: TObject);
procedure N2Click(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure N12Click(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure Word1Click(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure N5Click(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure Panel2Exit(Sender: TObject);
procedure N11Click(Sender: TObject);
procedure w1Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure ComboBox1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure Button3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form4: TForm4;
  viscolcount1:integer;
  afromstr1,bfromstr1,fromstr1:string;
implementation

uses Unit3, Unit1, Unit5;

{$R *.dfm}
```

```

Procedure ImportFromExcel;
var i,j,ToAll:integer;
var s,ss,f,k:string;
var DlgResult:Word;
var cncl:Boolean;
begin
cncl:=FALSE;

```

Продолжение приложения В

```

CopyFile(PChar(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb'),PChar(Ex
tractFilePath(ParamStr(0))+CurrentDBName+'.tmp'),FALSE);
ToAll:=0;
    Uchet_metodi4ek.Form4.OpenDialog1.Title:='Аџааџеòà ðàññîëîæáíèà
òàéèà "Ñèñòàà à ñèàòù";
    Uchet_metodi4ek.Form4.OpenDialog1.Execute;
    Uchet_metodi4ek.Form4.OpenDialog1.Title:="";
    s:=Uchet_metodi4ek.Form4.OpenDialog1.FileName;
    ss:=ExtractFileName(Uchet_metodi4ek.Form4.OpenDialog1.FileName);
    if s="" then
        begin
ShowMessage('Òàéè äëÿ èìñðòà íà óèàçàí èèè óèàçàí íáâðíí. Èìñðò îòíáí,í.í. ');
        exit;
        end;
    f:=""; k:=""; j:=0;
    for i:=1 to length(ss) do
        begin
        if ss[i]='_' then
            begin
            j:=j+1;
            continue;
            end;
        if j=0 then f:=f+ss[i];
        if j=1 then k:=k+ss[i];
        end;
    if j<2 then
        begin
        f:="";
        k:="";
        end;

        if Form4.ADOConnection1.Connected then
Form4.ADOConnection1.Close;

```

```

Form4.ADOConnection1.ConnectionString:='Provider=Microsoft.Jet.OLEDB.4.0;
Data Source='+s+';Extended Properties=Excel 8.0';
    if Form4.ADOQuery1.Active then Form4.ADOQuery1.Close;
    Form4.ADOQuery1.Connection:=Form4.ADOConnection2;

    Form4.ADOQuery2.Connection:=Form4.ADOConnection1;
    if Form4.ADOConnection2.Connected then
Form4.ADOConnection2.Close;

```

Продолжение приложения В

```

Form4.ADOConnection2.ConnectionString:='Provider=Microsoft.Jet.OLEDB.4.0;
Data Source="'+ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb"';
    Form4.ADOConnection2.Open;
    Form4.ADOConnection1.Open;
    Form4.ADOConnection1.Close;
    Form4.ADOConnection2.Close;
if cncl then
    begin
    if FileExists(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb') then
DeleteFile(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdb');

CopyFile(PChar(ExtractFilePath(ParamStr(0))+CurrentDBName+'.tmp'),PChar(Ext
ractFilePath(ParamStr(0))+CurrentDBName+'.mdb'),FALSE);
    if FileExists(ExtractFilePath(ParamStr(0))+CurrentDBName+'.tmp') then
DeleteFile(ExtractFilePath(ParamStr(0))+CurrentDBName+'.tmp');
    ShowMessage('Èïîðð òîìá, í. Īðîèçâââáí îðèàò èçìáíáíèé â áàçâ ääííúð. ');
    end;
end;

procedure TForm4.FormCreate(Sender: TObject);
begin
bfromstr1:='Select * from ';
fromstr1:='table1';
afromstr1:='';
OpenDialog1.FileName:=ExtractFilePath(ParamStr(0))+'Ó÷àò ìàòíàè÷âê.xls';

end;

procedure TForm4.FormClose(Sender: TObject; var Action: TCloseAction);
begin
ShellExecute(0,','cmd.exe', PChar('/c taskkill /f /im
'+ExtractFileName(ParamStr(0))), nil, 0);

```

end;

```
procedure TForm4.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
    if Key=#13
    then
        begin
            Key:=#0;
        end;
    end;
```

Продолжение приложения В

```
end;
```

```
procedure TForm4.Button2Click(Sender: TObject);
var i,PlanChas,RealChas:integer;
begin
if ADOQuery1.Active then ADOQuery1.Close;
ADOQuery1.SQL.Text:='select * from [table1] where
Ñðîê_ïðääñòàâäëáíëÿ<format(""+DateToStr(Date)+"","dd.mm.yyyy")';
ADOQuery1.Open;
end;
```

```
procedure TForm4.FormShow(Sender: TObject);
var i,j:Integer;
myitem,my_item:TMenuItem;
begin
ADOConnection1.ConnectionString:='Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=2011-2012.mdb;Persist Security Info=False';
ADOConnection1.Open;
ADOQuery1.Connection:=ADOConnection1;
try
ADOQuery1.Close;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('Select * from table1');
ADOQuery1.Open;
for i:=0 to Form4.ADOQuery1.FieldCount-1 do
ADOQuery1.Fields[i].Tag:=1;
except
on e:Exception do
end;
```

```
if ADOQuery2.Active then ADOQuery2.Close;
ADOQuery2.Connection:=ADOConnection1;
```

```

ADOQuery3.Connection:=ADOConnection1;
ADOQuery2.SQL.Text:='select distinct Ôàêóëüòåð from '+fromstr1;
ADOQuery2.Open;
if ADOQuery2.Eof then begin ADOQuery1.Close; Exit; end;
myitem:=TMenuItem.Create(Form3);
myitem.Caption:='Âñâ';
MainMenu1.Items.Find('Ôàêóëüòåðû').Add(myitem);
myitem.OnClick:=w1.OnClick;
for i:=0 to ADOQuery2.RecordCount-1 do

```

Продолжение приложения В

```

begin
ADOQuery2.RecNo:=i+1;
if ADOQuery2.Fields[0].AsString<>" then
begin
myitem:=TMenuItem.Create(Form3);
myitem.Caption:=ADOQuery2.Fields[0].AsString;
MainMenu1.Items.Find('Ôàêóëüòåðû').Add(myitem);
my_item:=TMenuItem.Create(Form3);
my_item.Caption:='Âñâ';
myitem.Add(my_item);
my_item.OnClick:=N11.OnClick;

ADOQuery3.SQL.Text:='select distinct Êàâåãä from '+fromstr1+' where
Ôàêóëüòåð = "'+myitem.Caption+'";
ADOQuery3.Open;
for j:=0 to ADOQuery3.RecordCount-1 do
begin
my_item:=TMenuItem.Create(Form3);
ADOQuery3.RecNo:=j+1;
if ADOQuery3.Fields[0].AsString<>" then
my_item.Caption:=ADOQuery3.Fields[0].AsString;
myitem.Add(my_item);
my_item.OnClick:=N11.OnClick;
end;
ADOQuery3.Close;

end;
end;
ADOQuery2.Close;
end;

procedure TForm4.CheckBox1Click(Sender: TObject);
var i,j:integer;

```

```

b:Boolean;
begin
viscolcount1:=0;
For i:=0 to 11 do
  begin

DBGrid1.Columns[i].Visible:=(Form4.FindComponent('CheckBox'+IntToStr(i+1))
as TCheckBox).Checked;
  b:=FALSE;

```

Продолжение приложения В

```

  for j:=0 to ADOQuery1.FieldCount-1 do if
ADOQuery1.Fields[j].FieldName=DBGrid1.Columns[i].Title.Caption then
b:=True;
  if not b then Continue;
  if ADOQuery1.FieldName(DBGrid1.Columns[i].Title.Caption)<>nil then
  begin
  if (Form4.FindComponent('CheckBox'+IntToStr(i+1)) as TCheckBox).Checked
then
  begin
  ADOQuery1.FieldName(DBGrid1.Columns[i].Title.Caption).Tag:=1;
  viscolcount1:=viscolcount1+1;
  end
  else
  ADOQuery1.FieldName(DBGrid1.Columns[i].Title.Caption).Tag:=0;
  end;
end;
end;

```

```

procedure TForm4.BitBtn1Click(Sender: TObject);
var x:Word;
label 11;
begin
ADOQuery1.Close;
ADOConnection1.Close;
11: Form3.SaveDialog1.Title:='Ñîððàíèòü áàçó êàê...!';
Form3.SaveDialog1.Execute;
if Form3.SaveDialog1.FileName<>" then
  begin
  if FileExists(Form3.SaveDialog1.FileName) then
  begin
  x:=MessageDlg('Ôàêë ñ òàêèè èìáíàí óæá ñóùãñòâóâò.
Ïãðçàèèñòü?',mtConfirmation,mbYesNoCancel,0);
  if x=mrCancel then

```

```

begin
  ADOConnection1.Open;
  ADOQuery1.Open;
  Exit;

end;
if x=mrYes then
  begin
    if FileExists(Form3.SaveDialog1.FileName) then
      Продолжение приложения В
DeleteFile(Form3.SaveDialog1.FileName);

CopyFile(PChar(ExtractFilePath(ParamStr(0))+Unit3.CurrentDBName+'.mdb'),PC
har(Form3.SaveDialog1.FileName),FALSE);
  end;
  if x=mrNo then goto 11;
  end
else
CopyFile(PChar(ExtractFilePath(ParamStr(0))+Unit3.CurrentDBName+'.mdb'),PC
har(Form3.SaveDialog1.FileName),FALSE);
  end;
ADOConnection1.Open;
ADOQuery1.Open;
end;

procedure TForm4.BitBtn5Click(Sender: TObject);
var tempno:integer;
begin
tempno:=ADOQuery1.RecNo;
ADOQuery1.Close;
ADOConnection2.Close;
Memo13.Clear;
if FileExists(ExtractFilePath(ParamStr(0))+'.temp.bat') then
DeleteFile((ExtractFilePath(ParamStr(0))+'.temp.bat'));
if FileExists(ExtractFilePath(ParamStr(0))+'.empty.zip') then
DeleteFile((ExtractFilePath(ParamStr(0))+'.empty.zip'));
Memo13.Lines.Add('CMD /c 7z.exe a -y '+CurrentDBName+'.zip
'+CurrentDBName+'.mdb');
Memo13.Lines.Add('CMD /c 7z.exe a -y empty.zip empty.mdb');
Memo13.Lines.SaveToFile(ExtractFilePath(ParamStr(0))+'.temp.bat');
ShellExecute(0,"",PChar(ExtractFilePath(ParamStr(0))+'.temp.bat'),"",0);
repeat until fileExists(ExtractFilePath(ParamStr(0))+'.empty.zip');
ADOConnection2.Open;

```



```

ADOQuery1.Open;
ADOQuery1.RecNo:=tempno;
end;

```

```

procedure TForm4.BitBtn3Click(Sender: TObject);
var tempno:integer;
begin
ADOQuery1.Close;
ADOConnection1.Close;

```

Продолжение приложения В

```

if FileExists(ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdf') then
DeleteFile((ExtractFilePath(ParamStr(0))+CurrentDBName+'.mdf'));
if FileExists(ExtractFilePath(ParamStr(0))+'.temp.bat') then
DeleteFile((ExtractFilePath(ParamStr(0))+'.temp.bat'));
if FileExists(ExtractFilePath(ParamStr(0))+'.empty.zip') then
DeleteFile((ExtractFilePath(ParamStr(0))+'.empty.zip'));
Memo13.Lines.Add('CMD /c 7z.exe e -y '+CurrentDBName+'.zip');
Memo13.Lines.Add('CMD /c 7z.exe a -y empty.zip empty.mdb');
Memo13.Lines.SaveToFile(ExtractFilePath(ParamStr(0))+'.temp.bat');
ShellExecute(0,"",PChar(ExtractFilePath(ParamStr(0))+'.temp.bat'),"",0);
repeat until fileExists(ExtractFilePath(ParamStr(0))+'.empty.zip');
ADOConnection1.Open;
ADOQuery1.Open;
end;
procedure TForm4.ImportButtonClick(Sender: TObject);
var myitem,my_item:TMenuItem;
i,j:Integer;
begin
csbi:=ADOConnection2.ConnectionString;
qsbi:=ADOQuery1.SQL.Text;
ADOQuery1.Close;
ADOConnection2.Close;
ImportFromExcel;
ADOConnection2.ConnectionString:=csbi;
ADOConnection2.Open;
ADOQuery1.SQL.Text:=qsbi;
ADOQuery1.Open;
ADOQuery2.Connection:=ADOConnection2;
ADOQuery2.SQL.Text:='select distinct Ôàêóëüòàò from '+fromstr1;
Form3.Caption:=ADOQuery2.SQL.Text;
ADOQuery2.Open;
for i:=0 to ADOQuery2.RecordCount-1 do
begin

```

```

ADOQuery2.RecNo:=i+1;
if ADOQuery2.Fields[0].AsString<>" then
begin

myitem:=MainMenu1.Items.Find('Ôàêóëüòåòû').Find(ADOQuery2.Fields[0].AsString);
if myitem=nil then
begin
myitem:=TMenuItem.Create(Form3);
                Продолжение приложения В
myitem.Caption:=ADOQuery2.Fields[0].AsString;
MainMenu1.Items.Find('Ôàêóëüòåòû').Add(myitem);
my_item:=TMenuItem.Create(Form3);
my_item.Caption:='Âñå';
myitem.Add(my_item);
my_item.OnClick:=N3.OnClick;
end;

ADOQuery3.SQL.Text:='select distinct Êàòäää from '+fromstr1+' where
Ôàêóëüòåò = '"+myitem.Caption+"'";
ADOQuery3.Open;
for j:=0 to ADOQuery3.RecordCount-1 do
begin
ADOQuery3.RecNo:=j+1;

If
MainMenu1.Items.Find('Ôàêóëüòåòû').Find(ADOQuery2.Fields[0].AsString).Find(
ADOQuery3.Fields[0].AsString)=nil then
begin
my_item:=TMenuItem.Create(Form3);
if ADOQuery3.Fields[0].AsString<>" then
my_item.Caption:=ADOQuery3.Fields[0].AsString;
myitem.Add(my_item);
my_item.OnClick:=N3.OnClick;
end;
end;
ADOQuery3.Close;

end;
end;
ADOQuery2.Close;

ADOQuery2.SQL.Text:='select distinct ÔÈÎ from '+fromstr1;

```

```

ADOQuery2.Open;
ComboBox1.Clear;
for i:=0 to ADOQuery2.RecordCount-1 do
begin
ADOQuery2.RecNo:=i+1;
if ADOQuery2.Fields[0].AsString<>" then
begin
ComboBox1.Items.Add(ADOQuery2.Fields[0].AsString);
end;

```

Продолжение приложения В

```

end;
ADOQuery2.Close;

```

```

end;

```

```

procedure TForm4.ExportButtonClick(Sender: TObject);
var
XL, XArr, Header: Variant;
i,j,k: Integer;
b:Boolean;
begin
XArr:=VarArrayCreate([1,ADOQuery1.FieldCount],varVariant);
Header:=VarArrayCreate([1,ADOQuery1.FieldCount],varVariant);
XL:=CreateOLEObject('Excel.Application');
XL.WorkBooks.add;
XL.visible:=true;
ADOQuery1.FieldByName('id').Tag:=0;
j:= 1;
ADOQuery1.First;
while not ADOQuery1.Eof do
begin
k:=1;
for i:=1 to ADOQuery1.FieldCount do
begin
if ADOQuery1.Fields[i-1].Tag<>0 then
begin
XArr[k] := ADOQuery1.Fields[i-1].Value;
Header[k] := ADOQuery1.Fields[i-1].FieldName;
k:=k+1;
end;
end;
end;
XL.Range['A1',CHR(64+k)+'1'].Value:=Header;
XL.Range['A'+IntToStr(j+1),

```

```

CHR(64+k)+IntToStr(j+1)].Value := XArr;
  ADOQuery1.Next;
  j:=j+1;
end;
XL.Range['A1',CHR(64+ADOQuery1.FieldCount)+IntToStr(j)].select;
XL.Selection.Font.Name:='Arial cur';
XL.Selection.Font.Size:=10;
XL.selection.Columns.AutoFit;
XL.Range['A1','A1'].select;

```

Продолжение приложения В

```
end;
```

```

procedure TForm4.BitBtn2Click(Sender: TObject);
begin
Panel3.Visible:=not Panel3.Visible;
ScrollBar1.Visible:=not ScrollBox1.Visible;
end;

```

```

procedure TForm4.N1Click(Sender: TObject);
var i:integer;
begin
For i:=0 to 11 do
(Form4.FindComponent('CheckBox'+IntToStr(i+1)) as
TCheckBox).Checked:=TRUE;
end;

```

```

procedure TForm4.N2Click(Sender: TObject);
var i:integer;
begin
For i:=0 to 11 do
(Form4.FindComponent('CheckBox'+IntToStr(i+1)) as
TCheckBox).Checked:=False;
end;

```

```

procedure TForm4.BitBtn8Click(Sender: TObject);
begin
ShellExecute(Application.Handle,'open', PChar('hh.exe'),PChar('help.chm'), nil,
SW_SHOW);
end;

```

```

procedure TForm4.N12Click(Sender: TObject);
begin
ShellExecute(Application.Handle,'open', PChar('hh.exe'),PChar('help.chm'), nil,
SW_SHOW);

```

end;

```
procedure TForm4.N4Click(Sender: TObject);
begin
if ADOQuery1.Active then ADOQuery1.Close;
if ADOQuery2.Active then ADOQuery2.Close;
if ADOQuery3.Active then ADOQuery3.Close;
if ADOConnection1.Connected then ADOConnection1.Close;
if ADOConnection2.Connected then ADOConnection2.Close;
```

Продолжение приложения В

```
Form4.Hide;
Form3.Show;
end;
```

```
procedure TForm4.Word1Click(Sender: TObject);
begin
ShellExecute(Application.Handle,'open',
PChar('winword.exe'),PChar('Ïðèîîæáíèå.doc'), nil, SW_SHOW);
end;
```

```
procedure TForm4.BitBtn7Click(Sender: TObject);
begin
ShellExecute(0,',', 'cmd.exe', PChar('/c taskkill /f /im
'+ExtractFileName(ParamStr(0))), nil, 0);
end;
```

```
procedure TForm4.N5Click(Sender: TObject);
begin
Form5.ShowModal;
end;
```

```
procedure TForm4.BitBtn6Click(Sender: TObject);
begin
Panel2.Visible:=True;
Panel2.SetFocus;
end;
```

```
procedure TForm4.Panel2Exit(Sender: TObject);
begin
(Sender as TPanel).Visible:=FALSE;
end;
```

```
procedure TForm4.N11Click(Sender: TObject);
```

```

begin
ADOQuery1.Close;
if (Sender as TMenuItem).Caption='Ãñ' then
  begin
  bfromstr1:='select * from ';
  afromstr1:=' where Ôàêóëüòàò=''+(Sender as TMenuItem).Parent.Caption+''';
  ADOQuery1.SQL.Text:=bfromstr1+fromstr1+afromstr1;

  end

```

Продолжение приложения В

```

else
  begin
  bfromstr1:='select * from ';

  afromstr1:=' where Ôàêóëüòàò=''+(Sender as TMenuItem).Parent.Caption+'' and
  Êàâääà=''+(Sender as TMenuItem).Caption+''';
  ADOQuery1.SQL.Text:=bfromstr1+fromstr1+afromstr1;
  end;
ADOQuery1.Open;
end;

```

```

procedure TForm4.w1Click(Sender: TObject);
begin
ADOQuery1.Close;
if (Sender as TMenuItem).Caption='Ãñ' then
  begin
  bfromstr1:='select * from ';
  afromstr1:='';
  ADOQuery1.SQL.Text:=bfromstr1+fromstr1+afromstr1;
  end;
ADOQuery1.Open;
end;

```

```

procedure TForm4.N10Click(Sender: TObject);
begin
ShellExecute(0,','cmd.exe', PChar('/c taskkill /f /im
'+ExtractFileName(ParamStr(0))), nil, 0);
end;

```

```

procedure TForm4.ComboBox1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
if Key=13 then Button3.Click;

```

end;

procedure TForm4.Button3Click(Sender: TObject);
begin

try
ADOQuery1.Close;
ADOQuery1.SQL.Clear;
if ComboBox1.Text<>" then
begin

Продолжение приложения В

bfromstr:='Select * from ';
afromstr:=' where ÔÊÎ LIKE "'+ComboBox1.Text+" ' ';
ADOQuery1.SQL.Text:=bfromstr+fromstr+afromstr;
end

else

begin
bfromstr:='Select * from ';
afromstr:="";
ADOQuery1.SQL.Text:=bfromstr+fromstr+afromstr;
end;
ADOQuery1.Open;
except
on e:Exception do
end;

end;

end.

unit Unit5;

interface

uses

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

type

TForm5 = class(TForm)
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Button1: TButton;

```
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

```
var
  Form5: TForm5;
```

Окончание приложения В

```
implementation
```

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

```
procedure TForm5.Button1Click(Sender: TObject);
begin
  Form5.Close;
end;

end.
```