ФЕДЕРАЛЬНОЕ АГЕНТСВО ПО ОБРАЗОВАНИЮ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «БелГУ»

**Факультет компьютерных наук и телекоммуникаций**

**Кафедра прикладной информатики**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

по дисциплине: *«Разработка и стандартизация программных средств и информационных технологий»*

на тему: *«Разработка и патентная защита программного средства «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники»»*

080801. 140606. 1406084.ПЗКР

Студент: А.В. Леонов

Руководитель курсовой работы: И.И. Чижов

Оценка курсовой работы:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Принял:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дата \_\_\_\_\_\_\_\_\_\_\_\_\_

Белгород 2010

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc277155595)

[1 ОБЩИЕ ВОПРОСЫ СТАНДАРТИЗАЦИИ 4](#_Toc277155596)

[2 ПРОГРАММНО-АЛГОРИТМИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРОГРАММНОГО СРЕДСТВА «АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО ИНЖЕНЕРА ПО СЕРВИСНОМУ ОБСЛУЖИВАНИЮ КОМПЬЮТЕРНОЙ ТЕХНИКИ» 5](#_Toc277155597)

[2.1 Назначение и основные возможности программного средства «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники». 5](#_Toc277155598)

[2.2 Создание бизнес логики 9](#_Toc277155599)

[2.3 Реализация программного средства «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники».](#_Toc277155600) 15

[3 ПАТЕНТОВАНИЕ ПРОГРАММНОГО СРЕДСТВА «АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО ИНЖЕНЕРА ПО СЕРВИСНОМУ ОБСЛУЖИВАНИЮ КОМПЬЮТЕРНОЙ ТЕХНИКИ» 24](#_Toc277155601)

[3.1 Общая информация о Роспатенте 24](#_Toc277155602)

[3.2 Оформление документов на регистрацию программного средства 25](#_Toc277155603)

[ЗАКЛЮЧЕНИЕ 31](#_Toc277155604)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 32](#_Toc277155605)

[Приложение 33](#_Toc277155606)

# ВВЕДЕНИЕ

В рамках курсовой работы мной было создано программное средство «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники». Курсовая работа будет включать в себя описание всех основных этапов работы с самой базой данных, Windows приложением и описание процесса патентования получившегося программного средства в Роспатент.

При разработке данного программного средства в курсовой работе была применена архитектура клиент-сервер со специализацией серверов. В простейшем случае типовое определение архитектуры клиент-сервер - приложение на клиенте, БД - на сервере - использует эту схему. Поэтому при создании информационных систем с клиент-серверной архитектурой необходимо использовать среду разработки, такую как Borland C++Builder 6. Возможность работы с серверами баз данных InterBase и Firebird, прозрачность подключения новых механизмов доступа к данным делает C++ Builder 6 наиболее естественным выбором для профессионалов индустрии информационных технологий.

К целям и задачам выполнения данной курсовой работы можно отнести:

* + создание программного средства «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники»;
  + закрепление знаний об общих принципах патентования, а также о функциях и задачах Роспатента;
  + получение навыков оформления документов для регистрации программного средства «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники» в Роспатенте.

Курсовая работа содержит 39 страниц, 30 рисунков, 2 таблицы, и приложение.

# 1 ОБЩИЕ ВОПРОСЫ СТАНДАРТИЗАЦИИ

Стандарт IEEE 1074-1995 охватывает полный жизненный цикл ПС, в котором выделяются шесть крупных базовых процессов. Эти процессы детализируются 16 частными процессами. В после­дних имеется еще более мелкая детализация в совокупности на 65 процессов-работ. Содержание каждого частного процесса начи­нается с описания общих его функций и задач и перечня дей­ствий - работ при последующей детализации. Для каждого про­цесса в стандарте представлены входная и результирующая ин­формация о его выполнении и краткое описание сущности процесса. Внимание сосредоточено преимущественно на непос­редственном создании ПС и на процессах предварительного проектирования. В приложении представлены четыре варианта адап­тации максимального состава компонентов ЖЦ ПС к конкрет­ным особенностям типовых проектов.

Хотя основные процессы близки к описанным в стандарте ISO 12207, общая архитектура и детализация частных процессов и работ в данном стандарте значительно отличаются. Процессы непосредственного создания ПС и его поддержка в стандарте представлены наибольшим числом частных процессов (около 70%), начинающихся с разработки требований к ПС и завершающихся приемо-сдаточными испытаниями, проводимыми заказчиком или пользователем.

# 2 ПРОГРАММНО-АЛГОРИТМИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРОГРАММНОГО СРЕДСТВА «АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО ИНЖЕНЕРА ПО СЕРВИСНОМУ ОБСЛУЖИВАНИЮ КОМПЬЮТЕРНОЙ ТЕХНИКИ»

## 2.1 Назначение и основные возможности программного средства «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники».

База данных абстрактно представляет совокупность организации, отношений, правил и процессов. Прежде чем подойти к началу проектирования структур и правил базы данных, необходимо провести системный анализ проблемы, работая с людьми, вовлеченными в определение структур, правил и требований реальной жизни, из которых будет соз­дан проект базы данных. Следует особенно подчеркнуть важность скрупулезного описания и анализа.

Анализ логических данных является итеративным процессом детализации и поиска сути во множестве входных данных, задач и выходных данных, которые должны быть реализованы в базе данных. В рамках данного проекта разрабатывается информационная система «Музыкальная коллекция».

Она основана на клиент-серверной технологии, разработана база данных, создано клиентское приложение.

Для хранения информации в базе данных, была разработана структура БД. Определено, какие таблицы (сущности) она в себя включает, какие поля (атрибуты) включают в себя таблицы.

Требования к функциональным характеристикам

1. Добавление данных о клиентах, заказах, инженерах, деталях.
2. Изменение данных о клиентах, заказах, инженерах, деталях.
3. Сортировка клиентов, заказов, деталей.
4. Удаление заказов, инженеров, клиентов, деталей
5. Поиск данных о клиентах, заказах, деталях.
6. Фильтрация о клиентах, заказах, деталях.
7. Просмотр сведений о клиентах, заказах, деталях, инженерах, видах расчета.
8. Формирование отчета о клиентах.

Структура базы данных

БД будет состоять из пяти таблиц: Клиенты, Заказы, Детали, Инженеры, Виды расчета.

На рисунке 1 изображена инфологическая модель БД «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники».

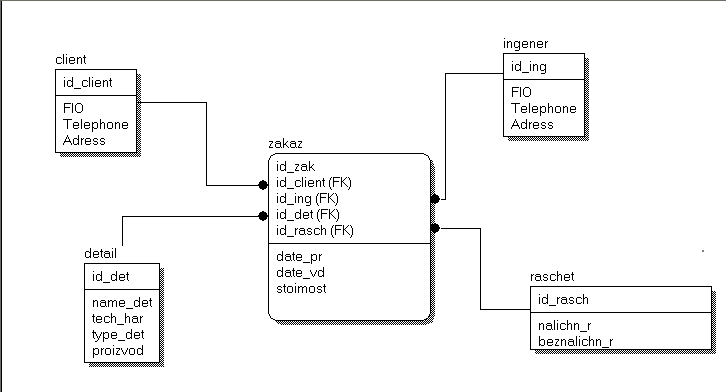


Рисунок 1 - Инфологическая модель БД

Разработано программное средство «Автоматизированное рабочее место инженера по обслуживанию компьютерной техники» с использованием клиент-серверной технологии.

Приложение «Автоматизированное рабочее место инженера по обслуживанию компьютерной техники» выполняет различные функции и соответствует определённым критериям. Программа легка и удобна в использовании, имеет простой интерфейс, содержит в себе информацию о произведениях, в ней присутствуют функции: удаления записей, сортировки, добавления записи, изменение записи, поиск записей, а также отчеты на основе информации в БД.

На рисунке 2 изображена контекстная диаграмма работы «Автоматизированное рабочее место инженера по обслуживанию компьютерной техники».

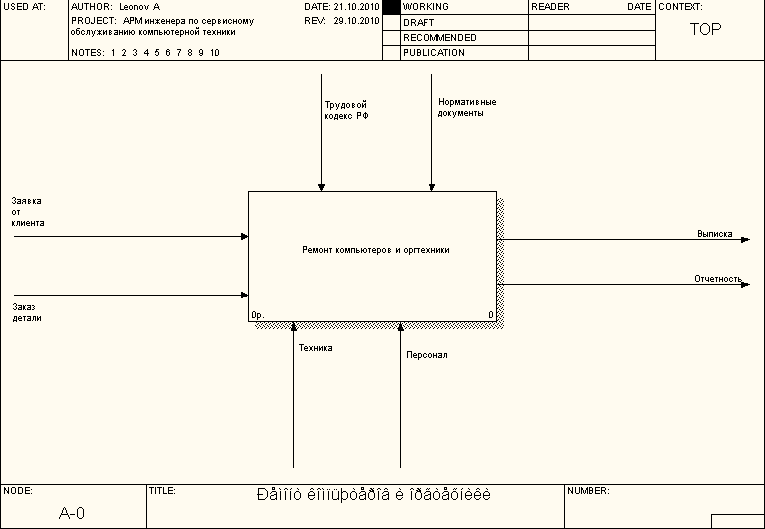


Рисунок 2 - Контекстная диаграмма работы программного средства «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники»»

На рисунке 3 изображена декомпозиция процесса работы программного средства «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники».

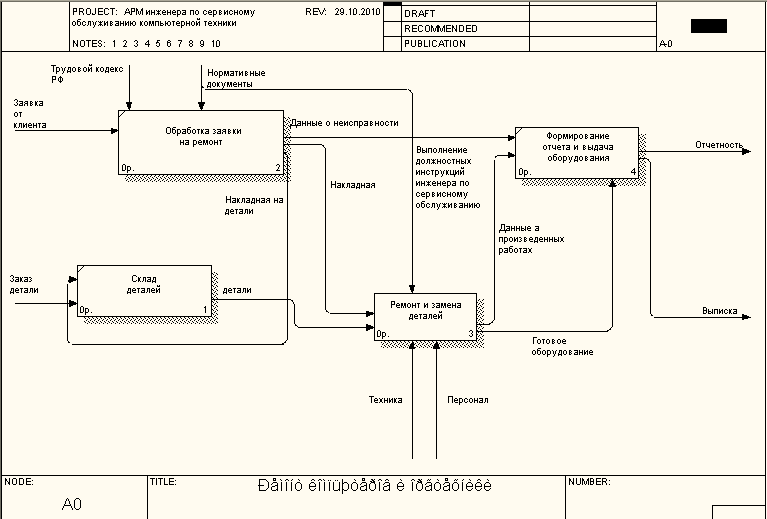


Рисунок 3 - Декомпозиция работы программного средства «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники»»

Для разработки программного средства используются СУБД Firebird 2.5 и утилита IBExpert. Для создания клиентского приложения будет использован Borland C++Builder 6.0.

Возможность работы с серверами баз данных InterBase и Firebird, прозрачность подключения новых механизмов доступа к данным делает C++ Builder 6 наиболее естественным выбором для профессионалов индустрии информационных технологий.

Предполагается предъявлять следующие требования к составу и параметрам технических средств:

* AMD Turion (tm) 64 Processor;
* ОЗУ 1,00 ГБ;
* Видеоадаптер – Intel 89245G Express Chipset Family;
* Звуковой адаптер – Creative SB X-Fi Xtreme Audio.

## 2.2 Создание бизнес логики

База данных состоит из таблиц, хранящих информацию о следующих объектах – клиенты, заказы, инженеры, детали, виды расчета.

Созданная база данных представляет собой структурированную информацию о работе инженера по сервисному обслуживанию (эта информация хранится в пяти связанных таблицах) и предоставляет возможность работы с ней с помощью соответствующего программного продукта.

Первая таблица «Клиенты» содержит следующую информацию: номер клиента, ФИО клиента, адрес клиента, телефон клиента.

Во второй таблице «Заказы» хранится информация о номере заказа, Дате приема оборудования, номере клиента, номере инженера, номере детали, номере вида расчета, дате выдачи оборудования и стоимости ремонта.

Третья таблица «Детали» содержит сведения о номере детали, о наименовании детали, о технических характеристиках, о производителе.

В четвертой таблице «Инженеры» хранится информация о номере инженера, ФИО инженера, содержит телефон и адрес.

Для полей таблиц нужны различные типы данных. Задаем их согласно предполагаемым записям (см. таблицу 1, таблицу 2).

Таблица №1 – Создание доменов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя домена | Тип | Длина | Not Null | Ограничения |
| D\_INDEX | smallint |  | + | >0 |
| D\_NAME | varchar | 50 |  |  |
| D\_STOIM | varchar | 10 |  |  |
| D\_DATE | DATE |  |  | <TODAY |
| D\_ADRES | varchar | 50 |  |  |
| D\_TELEPHONE | varchar | 15 |  |  |

Таблица 2 – Создание таблиц и определение их типов полей

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя таблицы | Поле | Тип | Длина | Not Null | PK | Имя домена |
| client | ID\_CLIENT | smallint |  | + | + | D\_INDEX |
| FIO\_CL | varchar | 50 |  |  | D\_NAME |
| ADRES | varchar | 50 |  |  | D\_ADRES |
| TELEPHONE | varchar | 15 |  |  | D\_TELEPHONE |
|  |  |  |  |  |  |
| detail | ID\_DETAIL | smallint |  | + | + | D\_INDEX |
| NAME\_DETAIL | varchar | 50 |  |  | D\_NAME |
| TECH\_HAR | varchar | 50 |  |  | D\_NAME |
| PROIZVOD | varchar | 50 |  |  | D\_NAME |
| ingener | ID\_ING | smallint |  | + | + | D\_INDEX |
| FIO\_ING | varchar | 50 | + |  | D\_NAME |
| TELEPHONE | varchar | 15 |  |  | D\_TELEPHONE |
| ADRES | varchar | 50 |  |  | D\_ADRES |
| raschet | Id\_rasch | smallint |  | + | + | D\_INDEX |
| Nalich\_R | varchar | 50 |  |  | D\_NAME |
| Beznalich\_R | varchar | 50 |  |  | D\_NAME |
| zakaz | Id\_zak | smallint |  | + | + | D\_INDEX |
| DATE\_PR | DATE |  |  |  | D\_DATE |
| ID\_CLIENT | smallint |  | + |  | D\_INDEX |
| ID\_ING | smallint |  | + |  | D\_INDEX |
| ID\_DET | smallint |  | + |  | D\_INDEX |
| ID\_RASCH | smallint |  | + |  | D\_INDEX |
| DATE\_V | DATE |  |  |  | D\_DATE |
| STOIMOST | VARCHAR | 10 |  |  | D\_STOIM |

Создали таблицы с помощью запроса SQL. Это можно увидеть по рисунку 4, рисунку 5, рисунку 6, рисунку 7, рисунку 8.

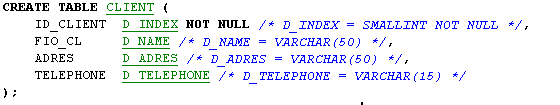


Рисунок 4 - Создание таблицы “Клиенты”

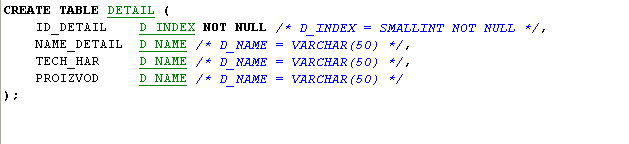


Рисунок 5 - Создание таблицы “Детали ”

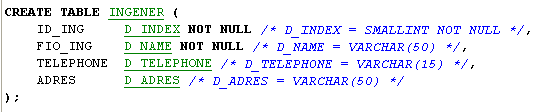


Рисунок 6 - Создание таблицы “Инженер”

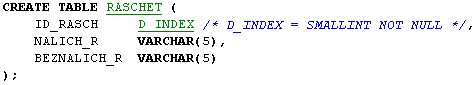


Рисунок 7 - Создание таблицы “Расчеты”

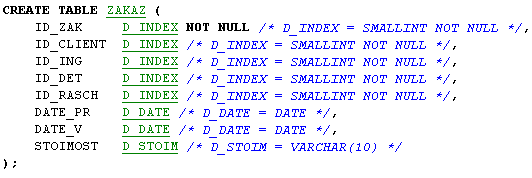


Рис. 8 – Создание таблицы «Заказы»

Далее были созданы генераторы и триггеры. Генератор представляет собой механизм, создающий уникальную последовательность чисел и автоматически заполняющий заданное поле при вставке или обновлении записей. Генераторы, как правило, используются в хранимых процедурах для автоматического заполнения поля (полей), входящих в первичный ключ.

Триггер является функцией, выполняющейся при вставке, изменении или удалении записи. Триггеры могут определяться как для таблиц, так и для обновляемых представлений.

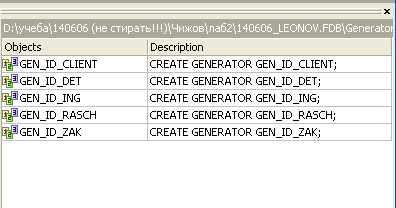


Рис. 9. Список генераторов

В результате было создано по 5 триггеров и генераторов:

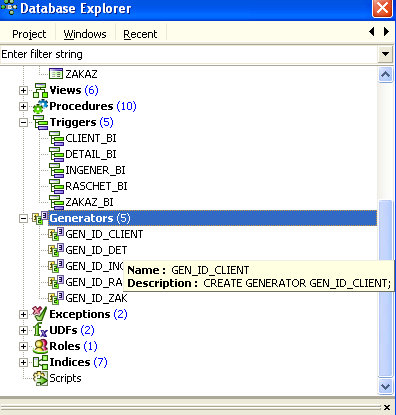


Рис. 10. Генераторы и триггеры базы данных

Представление «Список клиентов» (CLIENT\_INFO)

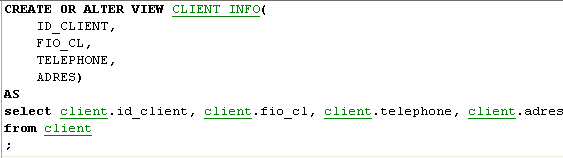


Рисунок 11 - Создание просмотра CLIENT\_INFO

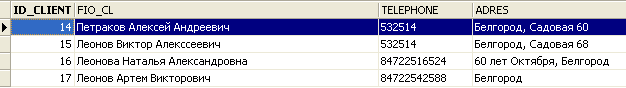


Рисунок 12 - Результат просмотра CLIENT\_INFO

Аналогичным образом были созданы представления остальных таблиц.

Примеры создания хранимых процедур:

Процедура «Добавить клиента» (ADD\_CLIENT)

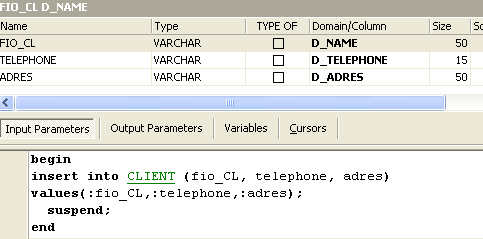


Рисунок 13 - Создание хранимой процедуры ADD\_ CLIENT

Процедура «Удалить заказ» (DEL\_ZAK)

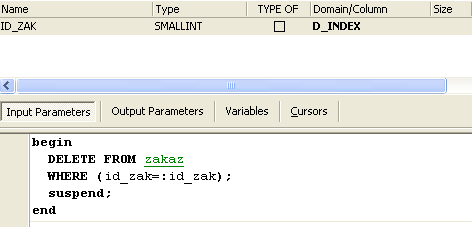


Рисунок 14 - Создание хранимой процедуры DEL\_ZAK

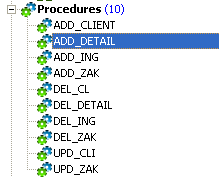


Рисунок 15 - Список созданных хранимых процедур

Было создано исключение. Исключения представляют собой именованное сообщение об ошибке.

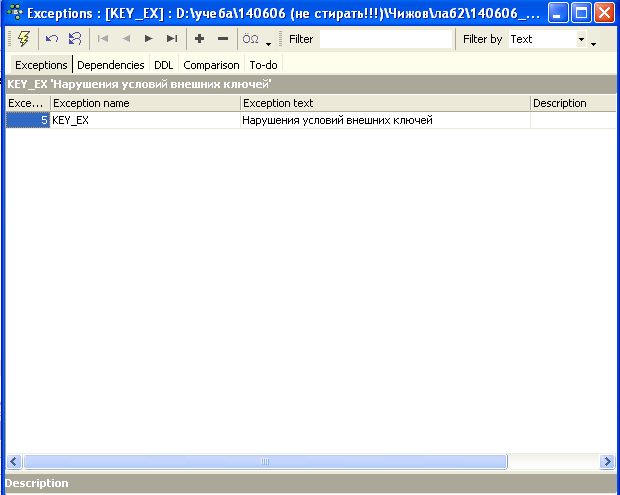


Рисунок 16 - Список созданных исключений

Исключение KEY\_EX внедряется следующим образом:

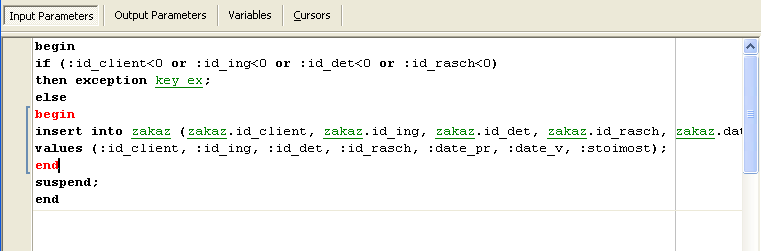


Рисунок 17 - Исключение KEY\_EX в хранимой процедуре. На примере процедуры ADD\_ZAK.

2.3 Реализация программного средства «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники»

C++Builder предоставляет разработчикам следующие компоненты для разроботки приложений:

* Компоненты управления данными Data Control, обеспечивающие отображение и редактирования записей на форме приложения.
* Компоненты вкладки Standart (Button, Label, Edit, RadioButton, CheckBox, RadioGroup, Panel)
* Компоненты доступа к данным Data Access - адресуют фактические данные, хранящиеся в файле базы данных.
* Компоненты вкладки QReport (QuickRep, QRSubDetail, QRLabel, QRDBText, QRBand, QRSysData), создание отчетов
* Компоненты Interbase (IBDatabase, IBTransaction, IBTable, IBStoredProc)

Наличие на форме большого количества невидимых компонентов в ряде случаев затрудняет проектирование пользовательского интерфейса. Кроме того, нередко бывает удобно отделить компоненты, отвечающие за доступ к данным и бизнес-логику информационной системы, от интерфейсных элементов, например, для обегчения ее дальнейшей модернизации. Для этой цели в C++ Builder имеется специальный тип, называемый модулем данных - TDataModule. На рис.18 представлен модуль данных разрабатываемого клиентского приложения.

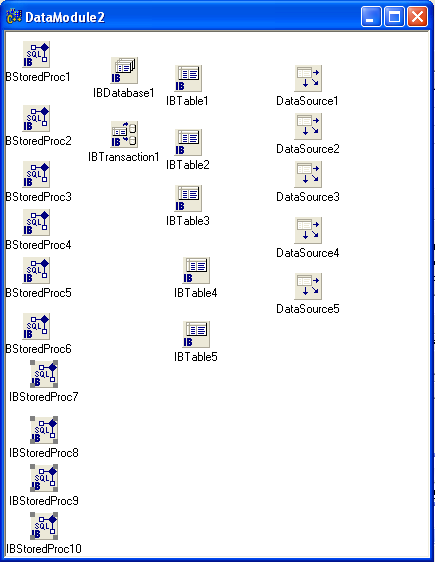


Рисунок 18 - Компонент DataModule2

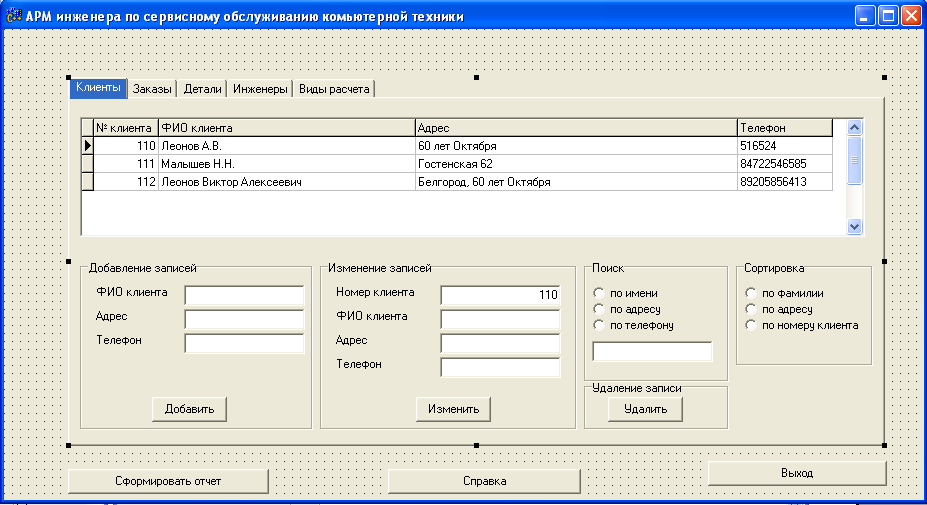


Рисунок 19 - Результат заполнения таблицы “Клиенты ”

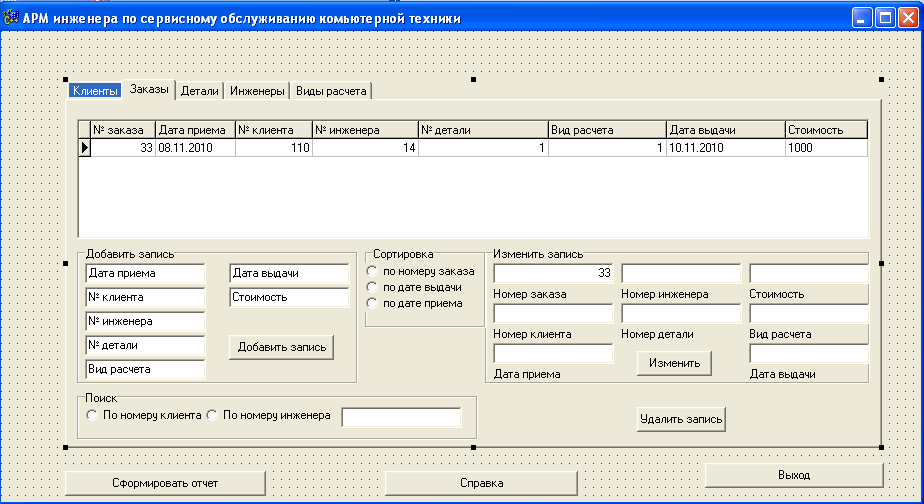


Рисунок 20 - Результат заполнения таблицы “Заказы”

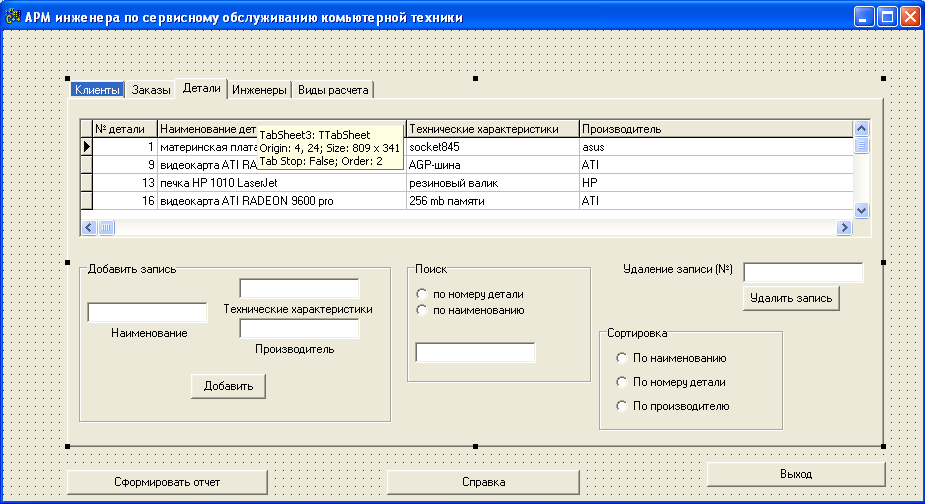


Рисунок 21 - Результат заполнения таблицы “Детали”

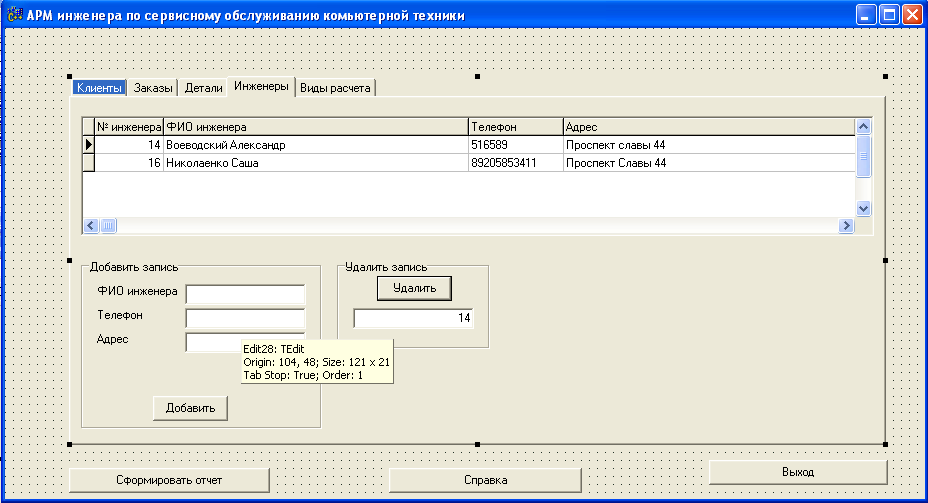


Рисунок 22 - Результат заполнения таблицы “Инженеры”

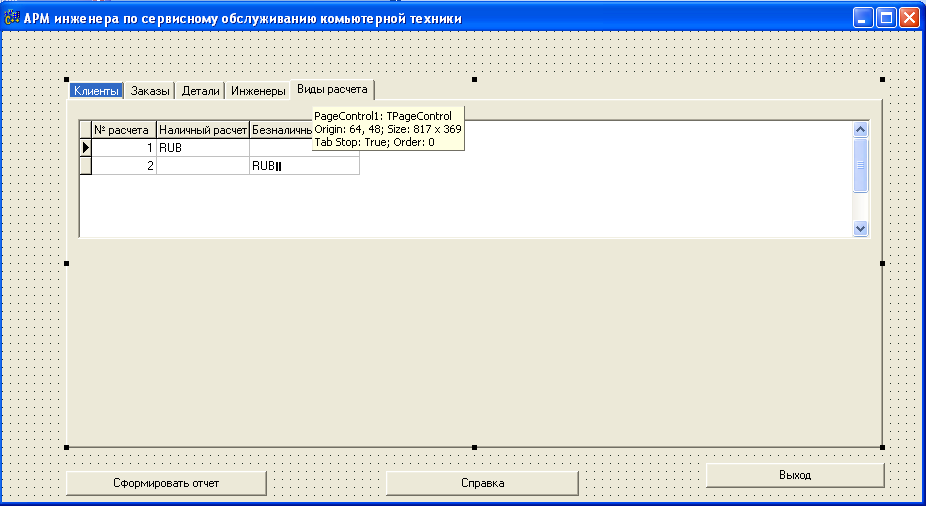


Рисунок 23 - Результат заполнения таблицы “Виды расчета”

## 2.3 Реализация программного средства «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники»

Рассмотрим подробнее реализацию программного средства «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники».

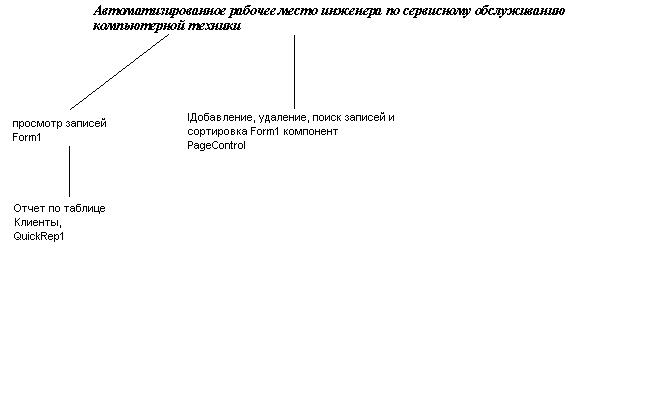


Рисунок 24 - Модульная структура программного обеспечения

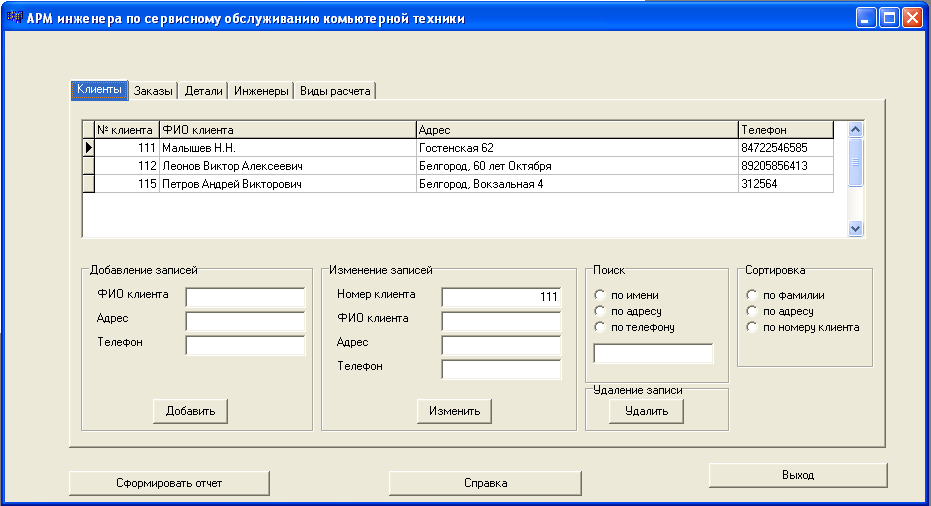


Рисунок 25 - Просмотр содержимого таблицы “Клиенты”

Имя таблицы задается в свойствах компонента IBTable. В свойствах компонента **DataSource** указываем свойство DataSet->IBTable.

Чтобы данные отображались в компоненте DBGrid, в свойстве DataSource указывается DataModule2->DataSource.

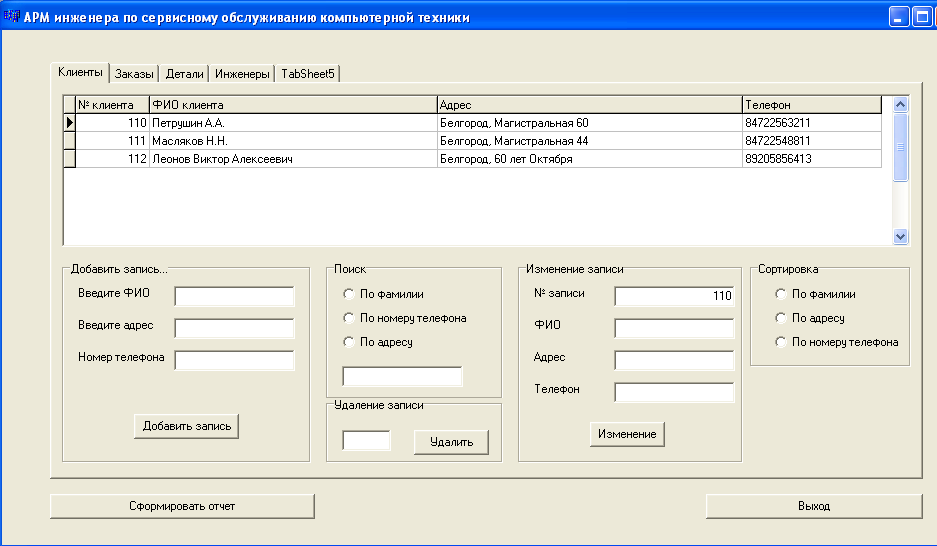


Рисунок 26 – Результат работы процедуры добавления данных

Для внесения данных во все таблицы была использована одна форма, чтобы сократить время на внесение пользователем данных в таблицу и упростить контроль над правильностью введенных данных.

Пользователь может заполнять данные по таблицам “Заказы”, “Клиенты”, “Детали ”, “Инженеры”.

Ввод значения поля в таблицу осуществляется, например, такой строкой как:

DataModule2->IBStoredProc1->ParamByName("FIO\_CL")->AsString=Form1->Edit1->Text;

Удаление записи из таблиц происходит посредством следующего кода:

DataModule2->IBStoredProc3->ParamByName("ID\_CLIENT")->AsString=DBEdit1->Text;

Осуществляется каскадное удаление – из таблицы «Клиенты».

При нажатии на кнопку “Удалить” запись удаляется.

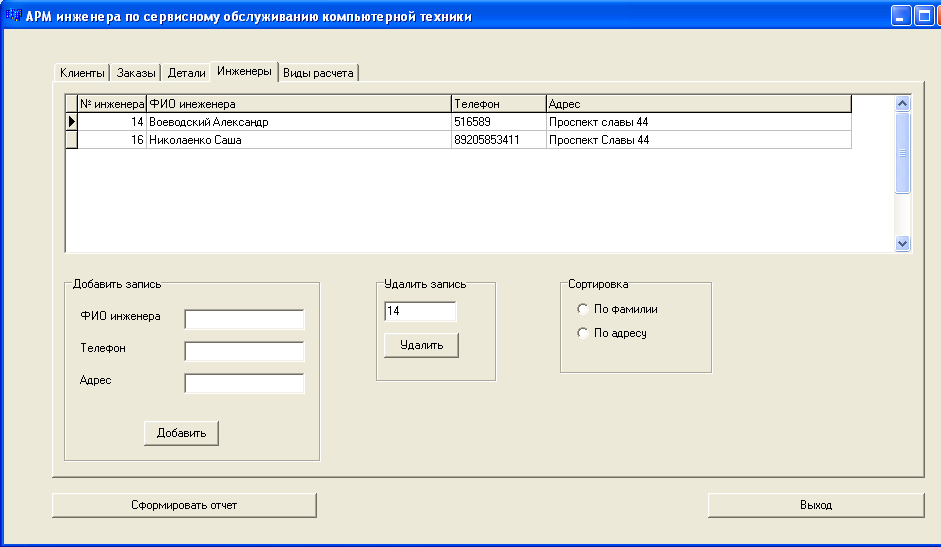


Рисунок 27 – Результат работы процедуры удаления данных

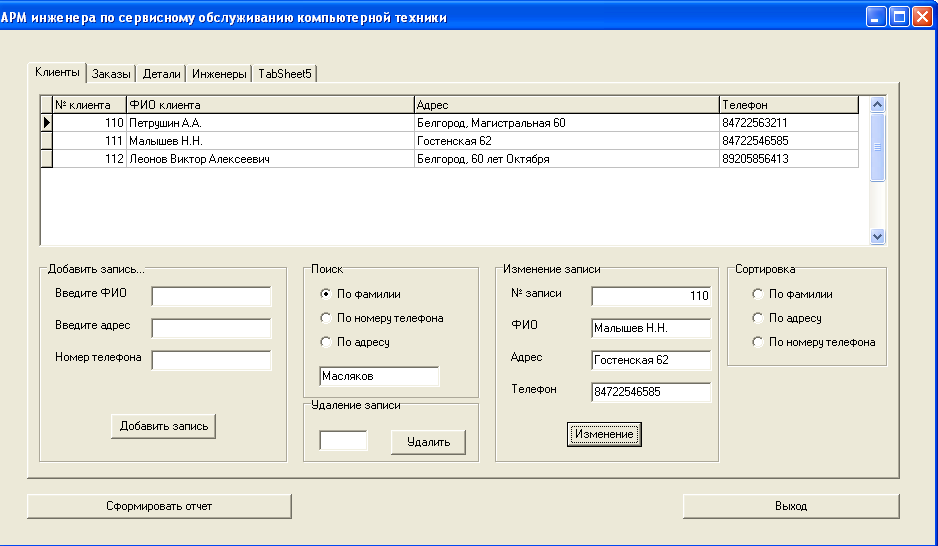


Рисунок 28 - Редактирование данных

Изменение записей в таблицах БД осуществляется программно:

DataModule2->IBStoredProc2->ParamByName("ID\_CLIENT")->AsString=DBEdit1->Text;

DataModule2->IBStoredProc2->ParamByName("FIO\_CL")->AsString=Edit4->Text;

Вывод данных в отчет также осуществляется программно с помощью компонента Button:

Form3->QuickRep1->Preview();

Form3->Visible=false;

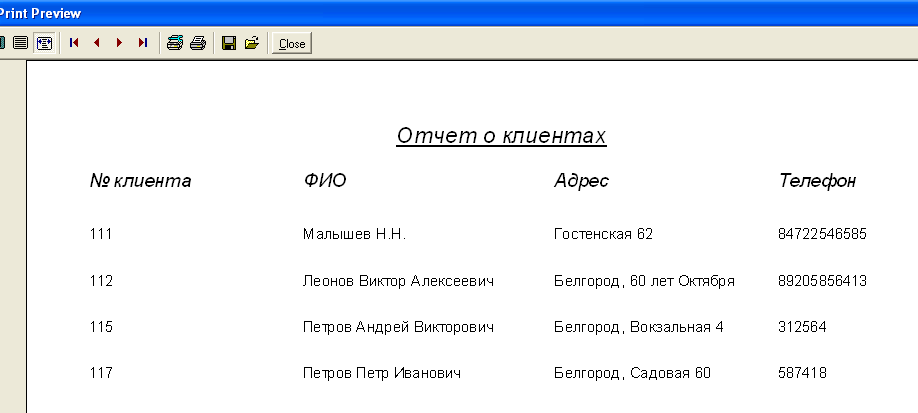


Рисунок 29 - Вывод данных в отчет

В зависимости от того, вкладка с какой таблицей выбрана в данный момент, при вводе нужных данных происходит поиск в таблице:

if (RadioButton1->Checked)

{TLocateOptions LO;

DataModule2->IBTable1->Locate("FIO\_CL", Edit7->Text, LO<<loPartialKey<< loCaseInsensitive);}

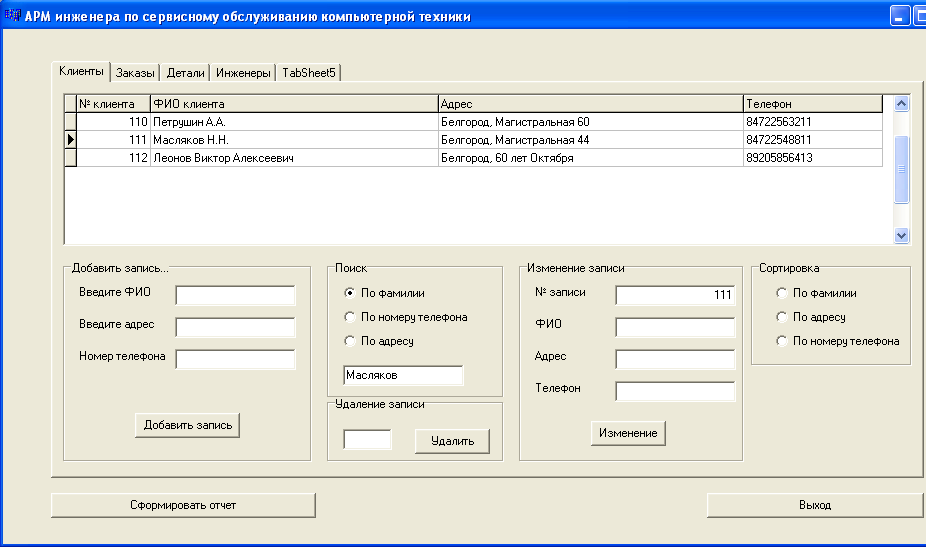


Рисунок 30 - Результат работы поиска в таблице «Клиенты»

# 3 ПАТЕНТОВАНИЕ ПРОГРАММНОГО СРЕДСТВА «АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО ИНЖЕНЕРА ПО СЕРВИСНОМУ ОБСЛУЖИВАНИЮ КОМПЬЮТЕРНОЙ ТЕХНИКИ»

## 3.1 Общая информация о Роспатенте

Федеральная служба по интеллектуальной собственности, патентам и товарным знакам (Роспатент) является федеральным органом исполнительной власти, осуществляющим функции по контролю и надзору в сфере правовой охраны и использования объектов интеллектуальной собственности, патентов и товарных знаков и результатов интеллектуальной деятельности, вовлекаемых в экономический и гражданско-правовой оборот, соблюдения интересов Российской Федерации, российских физических и юридических лиц при распределении прав на результаты интеллектуальной деятельности, в том числе создаваемые в рамках международного научно-технического сотрудничества. Роспатент находится в ведении Министерства образования и науки Российской Федерации.

Основными функциями Федеральной службы по интеллектуальной собственности, патентам и товарным знакам являются:

- обеспечение установленного Конституцией Российской Федерации, федеральными конституционными законами, федеральными законами и другими нормативными правовыми актами порядка предоставления в Российской Федерации правовой охраны объектам интеллектуальной собственности, а также порядка их использования;

- осуществление контроля и надзора за проведением экспертизы заявок на объекты интеллектуальной собственности и выдача охранных документов в установленном законодательством Российской Федерации порядке;

- регистрация прав на объекты интеллектуальной собственности, а также лицензионных договоров и договоров уступки прав в сфере интеллектуальной собственности и публикация сведений о зарегистрированных объектах интеллектуальной собственности;

- осуществление контроля и надзора за соблюдением порядка уплаты патентных пошлин и регистрационных сборов;

- проведение аттестации и регистрация патентных поверенных Российской Федерации и осуществление контроля за выполнением требований, предусмотренных законодательством Российской Федерации.

## 3.2 Оформление документов на регистрацию программного средства

Федеральная служба по интеллектуальной собственности, патентам и товарным знакам является федеральным органом исполнительной власти, осуществляющим функции по контролю и надзору в сфере правовой охраны и использования объектов интеллектуальной собственности, включая патенты и товарные знаки. Именно в Роспатент предоставляются документы на регистрацию программного средства «Информационная система «Музыкальная коллекция»».

Рассмотрим подробнее перечень документов, предоставляемых в Роспатент.

Первой для заполнения и по значимости является форма РП, в которой указываются данные:

- тип регистрируемой системы (собственно программа либо БД);

- имена и подписи владельца прав и исполнителей;

- название и альтернативное название ПС;

- дата заключения договора и выпуска программы;

- количество авторов ПС;

- сведения о других документах, направляемых вместе с данной формой;

- адреса всех перечисленных субъектов.

Заполнение формы РП представлено ниже.

Кроме формы РП в Роспатент необходимо направить еще ряд документов. В частности, так называемый, реферат, в котором отражается основная информация о программе (вес программного кода, язык написания, платформа, необходимые технические средства для ее использования). Так же стоит отметить, что реферат заполняется в двух экземплярах. Результат показан ниже.

|  |
| --- |
| РЕФЕРАТ  Автор: Леонов Артем Викторович  Правообладатель: Леонов Артем Викторович  Программа: Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники  Аннотация: Программное средство «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники» выполняет различные функции . Программа легка и удобна в использовании, имеет простой интерфейс, содержит в себе полную информацию о музыкальных произведениях, в ней присутствуют функции: удаления записей, сортировки, добавления записи, изменение записи, поиск записей, а также отчеты на основе информации в БД.  Тип ЭВМ: IBM PC  Язык: С++  ОС: Windows XP  Объём  программы: 12,01 Kб |

Завершающим этапом подготовки является написания результирующего заявления (сопроводительного письма), в котором указывается выбранный вид получения свидетельства и все документы, входящие в пакет, а так же ставится дата заполнения и подпись.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| В отдел регистрации программ для ЭВМ,  баз данных, топологий ИМС и передачи прав на них  Федерального государственного учреждения «Федеральный институт промышленной собственности Федеральной службы по интеллектуальной собственности, патентам и товарным знакам (ФГУ ФИПС)  Бережковская наб., 30, корп. 1, Москва,  Г-59, ГСП-5, 123995  Направляю Вам на регистрацию программу для ЭВМ «Информационная система «Музыкальная коллекция»», правообладателем исключительного права на которую (заявителем) являются Курочкин И.Н.  Комплектность заявки указана в приложении.  Приложение:   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  | Заявление (форма РП) | на | 1 | л. | в | 1 | экз. | |  | Распечатка исходного текста программы | на | 5 | л. | в | 1 | экз. | |  | Реферат | на | 1 | л. | в | 2 | экз. | |  | Платежный документ об уплате госпошлины | на | 1 | л. | в | 1 | экз. | |  | Материалы аудиовизуальных отображений | на | 5 | л. | в | 1 | экз. |   Всего на 13 листах + 1 платежный документ.  Свидетельство прошу выслать по почте  Представитель правообладателя *Подпись* *Расшифровка подписи*  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2010 г. |

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данного курсового проекта были достигнуты все поставленные цели и задачи:

* создали программное средство «Автоматизированное рабочее место инженера по сервисному обслуживанию компьютерной техники»;
* закрепили навыки проектирования моделей программных средств, а также их реализации;
* закрепили знания об общих принципах патентования, а также о функциях и задачах Роспатента;

Выполнение курсовой работы также помогло:

1. развить навыки самостоятельного планирования и выполнения научно-исследовательской работы;
2. получить опыт сбора и обработки материала, анализу научно-технической литературы;
3. углубить теоретические и практические знания в области патентования программных средств.

Разработанная информационная система обладает всей необходимой для работы функциональностью, интерфейс является очень удобным приложением для работы с БД.

Создание курсовой работы помогло закрепить пройденный материал по курсу «Разработка и стандартизация программных средств и информационных технологий», углубить знания по разработке удаленных баз данных и клиентских приложений, работающих с ними. Также были получены навыки патентной защиты программ для ЭВМ.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. «CASE-технологии. Современные методы и средства проектирования информационных систем» А.М. Вендров, http://[www.webresurs.ru](file:///C:\Documents%20and%20Settings\artem_leonov\Application%20Data\QIP\Profiles\artem_leonov\RcvdFiles\Y_647324060\www.webresurs.ru).
2. А.Я. Архангельский, «Программирование в C++ Builder 6». — М.:«БИНОМ», 2003 г.
3. Бори Х., «FIREBIRD. Руководство разработчика баз данных». –С-Пб.: «БХВ», 2006 г., 2-е издание.
4. Першиков В. И., Савинков В. М. Толковый словарь по информатике.: – М.: Финансы и статистика 1995г.; 2-е издание.
5. Тоу Д. Настройка SQL. Для профессионалов — СПб.: БХВ-Петербург 2004.
6. Кравец Л.Г. Патентно-информационное обеспечение конкурентной разведки; Информ-издат. центр. - М.: ИНИЦ Роспатента, 1999- 60 с.
7. Михелёв В.М. Базы данных и СУБД: учебное пособие – Белгород: Изд-во БелГУ, 2007 г. – 200 с.
8. Устинова Е.А. Патентный закон Российской Федерации и практика патентования изобретений в области информатики. - 2-е изд.. - М.: Российское агентство по патентам и товарным знакам, 2001 - 68 с.

# Приложение

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

#include "Unit2.h"

#include "Unit3.h"

#include "Unit4.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

DataModule2->IBStoredProc1->ParamByName("FIO\_CL")->AsString=Form1->Edit1->Text;

DataModule2->IBStoredProc1->ParamByName("ADRES")->AsString=Form1->Edit2->Text;

DataModule2->IBStoredProc1->ParamByName("TELEPHONE")->AsString=Form1->Edit3->Text;

DataModule2->IBStoredProc1->Prepare();

DataModule2->IBStoredProc1->ExecProc();

DataModule2->IBTransaction1->Commit();

DataModule2->IBTable1->Active=false;

DataModule2->IBTable1->Active=true;

DataModule2->IBTable2->Active=true;

DataModule2->IBTable3->Active=true;

DataModule2->IBTable4->Active=true;

DataModule2->IBTable5->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{DataModule2->IBStoredProc2->ParamByName("ID\_CLIENT")->AsString=DBEdit1->Text;

DataModule2->IBStoredProc2->ParamByName("FIO\_CL")->AsString=Edit4->Text;

DataModule2->IBStoredProc2->ParamByName("ADRES")->AsString=Edit5->Text;

DataModule2->IBStoredProc2->ParamByName("TELEPHONE")->AsString=Edit6->Text;

DataModule2->IBStoredProc2->Prepare();

DataModule2->IBStoredProc2->ExecProc();

DataModule2->IBTransaction1->Commit();

DataModule2->IBTable1->Active=false;

DataModule2->IBTable1->Active=true;}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button3Click(TObject \*Sender)

{

DataModule2->IBStoredProc3->ParamByName("ID\_CLIENT")->AsString=DBEdit1->Text;

DataModule2->IBStoredProc3->Prepare();

DataModule2->IBStoredProc3->ExecProc();

DataModule2->IBTransaction1->Commit();

DataModule2->IBTable1->Active=false;

DataModule2->IBTable1->Active=true;

DataModule2->IBTable2->Active=true;

DataModule2->IBTable3->Active=true;

DataModule2->IBTable4->Active=true;

DataModule2->IBTable5->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Edit7Change(TObject \*Sender)

{

if (RadioButton1->Checked)

{TLocateOptions LO;

DataModule2->IBTable1->Locate("FIO\_CL", Edit7->Text, LO<<loPartialKey<< loCaseInsensitive);}

else if (RadioButton2->Checked)

{TLocateOptions LO;

DataModule2->IBTable1->Locate("ADRES", Edit7->Text, LO<<loPartialKey<< loCaseInsensitive);}

else if (RadioButton3->Checked)

{TLocateOptions LO;

DataModule2->IBTable1->Locate("TELEPHONE", Edit7->Text, LO<<loPartialKey<< loCaseInsensitive);}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::RadioButton4Click(TObject \*Sender)

{

DataModule2->IBTable1->Active=false;

DataModule2->IBTable1->IndexFieldNames="FIO\_CL";

DataModule2->IBTable1->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::RadioButton5Click(TObject \*Sender)

{

DataModule2->IBTable1->Active=false;

DataModule2->IBTable1->IndexFieldNames="ADRES";

DataModule2->IBTable1->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::RadioButton6Click(TObject \*Sender)

{

DataModule2->IBTable1->Active=false;

DataModule2->IBTable1->IndexFieldNames="ID\_CLIENT";

DataModule2->IBTable1->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button4Click(TObject \*Sender)

{

DataModule2->IBStoredProc4->ParamByName("DATE\_PR")->AsString=Form1->Edit9->Text;

DataModule2->IBStoredProc4->ParamByName("ID\_CLIENT")->AsString=Form1->Edit10->Text;

DataModule2->IBStoredProc4->ParamByName("ID\_ING")->AsString=Form1->Edit11->Text;

DataModule2->IBStoredProc4->ParamByName("ID\_DET")->AsString=Form1->Edit12->Text;

DataModule2->IBStoredProc4->ParamByName("ID\_RASCH")->AsString=Form1->Edit13->Text;

DataModule2->IBStoredProc4->ParamByName("DATE\_V")->AsString=Form1->Edit14->Text;

DataModule2->IBStoredProc4->ParamByName("STOIMOST")->AsString=Form1->Edit15->Text;

DataModule2->IBStoredProc4->Prepare();

DataModule2->IBStoredProc4->ExecProc();

DataModule2->IBTransaction1->Commit();

DataModule2->IBTable2->Active=false;

DataModule2->IBTable2->Active=true;

DataModule2->IBTable1->Active=true;

DataModule2->IBTable3->Active=true;

DataModule2->IBTable4->Active=true;

DataModule2->IBTable5->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Edit22Change(TObject \*Sender)

{

if (RadioButton8->Checked)

{TLocateOptions LO;

DataModule2->IBTable2->Locate("ID\_CLIENT", Edit22->Text, LO<<loPartialKey<< loCaseInsensitive);}

else if (RadioButton9->Checked)

{TLocateOptions LO;

DataModule2->IBTable2->Locate("ID\_ING", Edit22->Text, LO<<loPartialKey<< loCaseInsensitive);}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button6Click(TObject \*Sender)

{

DataModule2->IBStoredProc5->ParamByName("ID\_CLIENT")->AsString=Form1->Edit9->Text;

DataModule2->IBStoredProc5->ParamByName("ID\_ING")->AsString=Form1->Edit10->Text;

DataModule2->IBStoredProc5->ParamByName("ID\_DET")->AsString=Form1->Edit11->Text;

DataModule2->IBStoredProc5->ParamByName("STOIMOST")->AsString=Form1->Edit12->Text;

DataModule2->IBStoredProc5->ParamByName("ID\_RASCH")->AsString=Form1->Edit13->Text;

DataModule2->IBStoredProc5->ParamByName("DATE\_PR")->AsString=Form1->Edit14->Text;

DataModule2->IBStoredProc5->ParamByName("DATE\_V")->AsString=Form1->Edit15->Text;

DataModule2->IBStoredProc5->Prepare();

DataModule2->IBStoredProc5->ExecProc();

DataModule2->IBTransaction1->Commit();

DataModule2->IBTable1->Active=false;

DataModule2->IBTable1->Active=true;

DataModule2->IBTable2->Active=true;

DataModule2->IBTable3->Active=true;

DataModule2->IBTable4->Active=true;

DataModule2->IBTable5->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button7Click(TObject \*Sender)

{

DataModule2->IBStoredProc7->ParamByName("NAME\_DETAIL")->AsString=Edit23->Text;

DataModule2->IBStoredProc7->ParamByName("TECH\_HAR")->AsString=Edit24->Text;

DataModule2->IBStoredProc7->ParamByName("PROIZVOD")->AsString=Edit25->Text;

DataModule2->IBStoredProc7->Prepare();

DataModule2->IBStoredProc7->ExecProc();

DataModule2->IBTransaction1->Commit();

DataModule2->IBTable1->Active=true;

DataModule2->IBTable2->Active=true;

DataModule2->IBTable3->Active=true;

DataModule2->IBTable4->Active=true;

DataModule2->IBTable5->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button5Click(TObject \*Sender)

{

DataModule2->IBStoredProc6->ParamByName("ID\_ZAK")->AsString=DBEdit2->Text;

DataModule2->IBStoredProc6->Prepare();

DataModule2->IBStoredProc6->ExecProc();

DataModule2->IBTransaction1->Commit();

DataModule2->IBTable1->Active=false;

DataModule2->IBTable1->Active=true;

DataModule2->IBTable2->Active=true;

DataModule2->IBTable3->Active=true;

DataModule2->IBTable4->Active=true;

DataModule2->IBTable5->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button8Click(TObject \*Sender)

{

DataModule2->IBStoredProc6->ParamByName("ID\_ING")->AsString=DBEdit3->Text;

DataModule2->IBStoredProc8->ParamByName("FIO\_ING")->AsString=Edit27->Text;

DataModule2->IBStoredProc8->ParamByName("TELEPHONE")->AsString=Edit28->Text;

DataModule2->IBStoredProc8->ParamByName("ADRES")->AsString=Edit29->Text;

DataModule2->IBStoredProc8->Prepare();

DataModule2->IBStoredProc8->ExecProc();

DataModule2->IBTransaction1->Commit();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button9Click(TObject \*Sender)

{

DataModule2->IBStoredProc9->ParamByName("ID\_ING")->AsString=DBEdit3->Text;

DataModule2->IBStoredProc9->Prepare();

DataModule2->IBStoredProc9->ExecProc();

DataModule2->IBTransaction1->Commit();

DataModule2->IBTable4->Active=false;

DataModule2->IBTable4->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button11Click(TObject \*Sender)

{

Form3->QuickRep1->Preview();

Form3->Visible=false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button10Click(TObject \*Sender)

{

Form4->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button12Click(TObject \*Sender)

{

Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button13Click(TObject \*Sender)

{

DataModule2->IBStoredProc10->ParamByName("ID\_DETAIL")->AsString=Edit30->Text;

DataModule2->IBStoredProc10->Prepare();

DataModule2->IBStoredProc10->ExecProc();

DataModule2->IBTransaction1->Commit();

DataModule2->IBTable3->Active=false;

DataModule2->IBTable3->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::RadioButton14Click(TObject \*Sender)

{

DataModule2->IBTable3->Active=false;

DataModule2->IBTable3->IndexFieldNames="NAME\_DETAIL";

DataModule2->IBTable3->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::RadioButton15Click(TObject \*Sender)

{

DataModule2->IBTable3->Active=false;

DataModule2->IBTable3->IndexFieldNames="ID\_DETAIL";

DataModule2->IBTable3->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::RadioButton16Click(TObject \*Sender)

{

DataModule2->IBTable3->Active=false;

DataModule2->IBTable3->IndexFieldNames="PROIZVOD";

DataModule2->IBTable3->Active=true;

}

//---------------------------------------------------------------------------