Федеральное агентство по образованию и науке

Государственное образовательное учреждение высшего профессионального образования

Камская государственная

инженерно-экономическая академия

Кафедра прикладной информатики и управления

**Курсовая работа**

**по дисциплине «Базы данных»**

Информационно-аналитическая система «MTTS»

**Выполнила:** студент гр. 1312

Маркелова И. А.

**Проверил:** доцент**:**

Хузятов Ш.Ш.

г. Набережные Челны

2009 г.

# **СОДЕРЖАНИЕ**

ВВЕДЕНИЕ

Задание на курсовую работу

1.МЕТОД РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ

1.1 Создание псевдонима для работы с базой данных

1.2 Организация доступа к таблицам база данных

2. Проектирование алгоритмов обработки информации

2.1 Форма «MTTS»

2.2. Добавление записи в таблицу

2.3. . Изменение полей выбранной записи

2.4. Удаление записи из таблицы

2.5. Поиск данных

2.6. Сортировка данных

2.7. Фильтрация данных

2.8. Создание отчета

2.9. Изменение цвета формы и изменение шрифта

2.10.Регистрация разговоров

3. Форма «MTTS страница «АБОНЕНТЫ»

4. Форма «MTTS страница ТАРИФЫ»

5. Форма «MTTS страница ПЕРЕГОВОРЫ»

ЗАКЛЮЧЕНИЕ

Список использованных источников

# **ВВЕДЕНИЕ**

Основы современной информационной технологии составляют базы данных (БД) и системы управления базами данных (СУБД), роль которых как единого средства хранения, обработки и доступа к большим объемам информации постоянно возрастает. При этом существенным является постоянное повышение объемов информации, хранимой в БД, что влечет за собой требование увеличения производительности таких систем. Резко возрастает также в разнообразных применениях спрос на интеллектуальный доступ к информации. Это особенно проявляется при организации логической обработки информации в системах баз знаний, на основе которых создаются современные экспертные системы.

Целью данной курсовой работы является создание Информационно-аналитической системы (базы данных) «Реализация кондитерских изделий». В работе были использованы такие методы как сортировка, добавление, удаление, изменение и поиск данных, фильтрация, диапазон.

Delphi является средой разработки, используемой, прежде всего для поддержки и разработки приложений, предназначенных как для отдельных рабочих станций, так и для серверов. Delphi может функционировать под управлением операционной системы Windows 95, 98, NT. Delphi имеет графический пользовательский интерфейс, подобный используемому в Microsoft Visual Basic и C++. Поскольку пользовательский интерфейс создается визуально, о Delphi говорят как о среде, предназначенной для быстрого создания приложений. В основе идеологии Delphi лежит технология визуального проектирования и событийного программирования (программирование процедур обработки событий), применение которых позволяет существенно сократить время разработки и облегчить процесс создания приложений.

# **Задание на курсовую работу**

Курсовая работа должна представлять собой информационно – аналитическую систему управления базой данных «MTTS».

# При запуске программы перед основной формой должна появляться заставка, содержащая оригинальное изображение, соответствующее теме программы. Основная форма должна содержать 3 вкладки (Page Control), на каждой из которых представляется информация из отдельной таблицы: на первой вкладке – с использованием компонента DBGrid, на второй – с использованием DBEdit, на третьей – Query. Форма должна содержать действующие компоненты MainMenu, PopupMenu с основными командами (сброс, выход, о программе, редактировать, добавить, изменить, удалить, поиск и т.д.).

# **1. МЕТОД РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ**

Информационно-аналитическая система «МТТS» представляет собой приложение, с помощью которого пользователю представлены возможности для:

* Добавления (регистрации) абонентов;
* Изменения данных абонентов;
* Удаления абонентов;
* Регистрации междугородних переговоров;

## 

## 1.1 Создание псевдонима для работы с базой данных

Процесс создание БД состоит из 3 этапов:

1. Создание каталога;
2. Создание таблиц;
3. Создание псевдонима.

Для создание таблицы в PARADOX7:

Tools – Database Desktop – File – New – Table – Paradox7 – Ok.

Указываешь тип данных и сохраняешь.

Создание Alias – это некоторое короткое слово, предоставленное для указание пути к БД (к каталогу, в котором хранятся файлы таблиц). Пуск – Программы –Borland Delphi7 – BDE Administrator (в левой части перечислены все псевдонимы). Меню Object – New – OK. Для того чтобы зарегистрировать псевдоним: Object – Apply – OK. В свойствах для компонента Table – Database Name указать название псевдонима, а в свойстве Active – True.

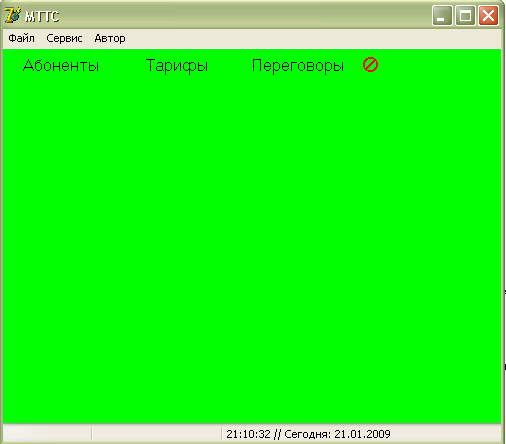
## 1.2 Организация доступа к таблицам база данных

#### Для того чтобы отобразить данные на фоне, нужно выбрать вкладку DataAccess - компонент DataSource и вкладку DataControls - компонент DBGrid. DataSourse свойство - Dataset - выбираем название таблицы. А в DBGrid указываем Data Source, которая была использована. В компоненте Table – свойство Active – заменить на True. Заполнение осуществляется, после нажатие Run, после заполнения нужно обратно заменить свойство Active для компонента Table на False и обратно на True. Если таблицы взяты из СУБД ACCESS и SQL, нужно выбрать вкладку ADO - компоненты ADOTable, ADO Connection. В компоненте ADO Connection указываем путь к таблицам, которые должны открыться. Для того чтобы отобразить данные на фоне, выполняем те же действия, что и для предыдущего. В компоненте ADOTable указываем в свойстве Connection – ADOConnection, в котором выбран путь к таблицам, в свойстве DataSourse выбираем необходимый из них, в свойстве Table Name - название таблицы, свойство Active - True. Для того чтобы связать подчиненные таблицы необходимо выбрать таблицу, которую нужно связать, в ее свойствах MasterSource – выбрать нужный, в свойстве MasterFields – выделяем поля, которые нужно связать – Add – ok.

# **2. Проектирование алгоритмов обработки информации**

## 

## 2.1 Форма «MTTS»

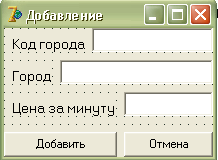


Форма «MTTS» содержится из трех вкладок. На первой вкладке указаны абоненты, где можно выполнить сортировку по возрастанию и по убыванию, также можно выполнить фильтрацию. Можно выполнить добавление новых данных, изменение, удаление не нужных данных, выполнить в таблице поиск по коду абонента, также можно представить регистрацию абонента, просмотреть квитанцию на переговоры. Можно просмотреть отчет «Список абонентов» и «Список должников». В меню «Файл» находится справка по данной информационной-аналитической системе; можно произвести настройки шрифта, цвета и размера таблиц.

На второй вкладке указана таблица в режимы формы, при помощи Navigatorа можно листать данные, удалять, добавлять и т.д.

## 2.2. Добавление записи в таблицу

Нажав на кнопку «Добавление» на самой основной форме, открывается форма, где нужно ввести новый абонент:



Нажать «Добавление».

Программа:

procedure Ttaradd.Button1Click(Sender: TObject);

var i: integer;

begin

if (edit1.Text='') or (edit2.Text='') or (edit3.Text='') then

begin

showmessage('Введены не все поля'+#13+'Добавление невозможно');

end else

begin

dd.Query2.Filter:='tarkod='+edit1.Text;

dd.Query2.Filtered:=true;

if dd.Query2.RecordCount=1 then

begin

showmessage('Введенный код уже зарегестрирован за городом '+dd.Query2.Fields[1].Text+#13+'Совпадение кодов недопустимо' );

edit1.SetFocus;

dd.Query2.Filtered:=false;

end else

begin

dd.DataSource2.DataSet:=dd.Table2;

dd.Table2.Append;

dd.Table2.Fields[0].AsFloat:=strtofloat(edit1.text);

dd.Table2.Fields[1].Text:=edit2.Text;

dd.Table2.Fields[2].AsFloat:=strtofloat(edit3.Text);

dd.Table2.Post;

dd.DataSource2.DataSet:=dd.Query2;

dd.Query2.Close;

dd.Query2.SQL.Clear;

dd.Query2.SQL.Add('Select \* FROM Tarif ORDER BY tarkod');

dd.Query2.Filtered:=false;

dd.Query2.Open;

edit1.Clear;

edit2.Clear;

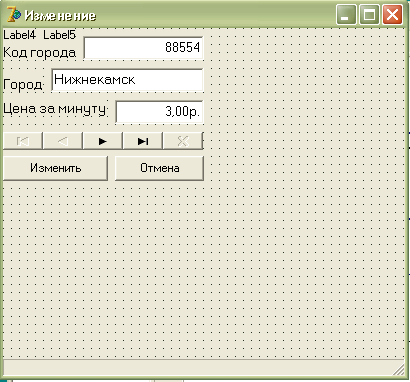
edit3.Clear;

edit1.SetFocus;

end;

## 2.3. Изменение полей выбранной записи

Нажав на кнопку «Изменение» на самой основной форме, открывается форма, где выбирается из раскрывающегося списка абонент, по которому нужно выполнить изменения:



Нажать «Изменить».

Программа:

procedure Ttared.Button1Click(Sender: TObject);

var i, n, j: integer;

//// ИЗМЕНЕНИЕ В ТАРИФАХ

begin

//dd.Query2.RecNo:=dd.Table2.RecNo;

dd.DataSource2.DataSet:=dd.Table2;

dd.Table2.Edit;

dd.Table2.Fields[0].AsFloat:=strtofloat(kod.Text); //strtofloat(label4.Caption);

dd.Table2.Fields[1].Text:=gorod.Text; //label5.Caption;

//dd.Table2.Fields[2].AsCurrency:=strtofloat(dbedit3.Text)\*1;

dd.Table2.Post;

//// ИЗМЕНЕНИЕ В ПЕРЕГОВОРАХ

begin

label4.Caption:=kod.Text;

label5.Caption:=gorod.Text;

for i:=1 to dd.Table3.RecordCount do

begin

dd.Table3.Edit;

//dd.Table3.RecNo:=i;

dd.Table3.Fields[6].AsFloat:=strtofloat(label4.Caption);

dd.Table3.Fields[1].Text:=label5.Caption;

dd.Table3.Post;

//dd.Table3.Filter:='pertel='+dbedit1.Text;

//dd.Table3.Filtered:=true;

dd.Table3.Next;

end;

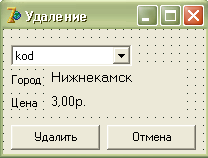
end;

dd.Table3.First;

end.

## 2.4. Удаление записи из таблицы

Нажав на кнопку «Удаление» на самой основной форме, открывается форма, где выбирается из раскрывающегося списка абонент, который нужно удалить:



Нажать «Удалить».

Программа:

procedure Ttardel.Button1Click(Sender: TObject);

var i: integer;

begin

if messagedlg('Удалить город: '+dd.Table2.Fields[1].Text, mtinformation,[mbYes,mbNo],0)=mrYes

then

begin

dd.Table2.Delete;

//dd.Table1.Post;

dd.Query2.Close;

dd.DataSource2.DataSet:=dd.Query2;

dd.Query2.Open;

for i:=1 to dd.Table3.RecordCount do

begin

dd.Table3.RecNo:=i;

dd.Table3.Delete;

end;

dd.Query3.Close;

dd.DataSource3.DataSet:=dd.Query3;

dd.Query3.Open;

// dd.Table3.Post;

end

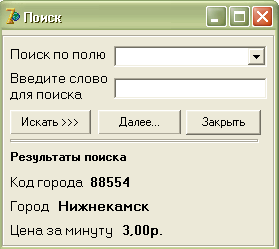
else exit;

end;

## 

## 2.5. Поиск данных

Нажав на кнопку «Поиск» на самой основной форме, открывается форма, где выбирается сама таблица, по которой нужно выполнить поиск и из раскрывающегося списка выбирается те значения, которые нужно найти. Для таблицы «Изделие» поиск выполняется следующим образом:



Нажать «Поиск».

Программа:

procedure Ttarfind.Button1Click(Sender: TObject);

var s, s1: string;

begin

if pole.ItemIndex=0 then

begin

dd.DataSource2.DataSet:=dd.Query2;

s:='tarkod='+ search.Text;

dd.Query2.Filter:=s;

dd.Query2.Filtered:=true;

if dd.Query2.RecordCount=0 then

showmessage('Поиск закончен. Запись НЕ НАЙДЕНА');

end;

if pole.ItemIndex=1 then

begin

dd.DataSource2.DataSet:=dd.Query2;

dd.Query2.close;

dd.Query2.SQL.Clear;

dd.Query2.SQL.Add('SELECT \* FROM tarif');

s1:='WHERE targor'+' LIKE "'+search.Text+'%"';

dd.Query2.SQL.Add(s1);

dd.Query2.Open;

if dd.Query2.RecordCount>=1 then

label6.Caption:='Результаты поиска: Найдены :'+floattostr(dd.Query2.RecordCount)+' зап.';

if dd.Query2.RecordCount=0 then

begin

showmessage('Поиск закончен. Запись НЕ НАЙДЕНА');

search.SetFocus;

end;

end;

if dd.Query2.RecordCount>1 then

begin

label1.Visible:=true;

label2.Visible:=true;

label3.Visible:=true;

label6.Visible:=true;

dbtext1.Visible:=true;

dbtext2.Visible:=true;

dbtext3.Visible:=true;

abfind.height:=264;

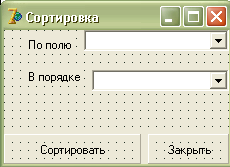
button2.Enabled:=true;

end;

end;

## 2.6 Сортировка данных

Сортировка выполняется на первой вкладке формы «MTTS». Можно выполнить сортировку по возрастанию и по убыванию. Нужно выбрать то поле, по которому нужно выполнить сортировку.



Программа для выполнения сортировки по возрастанию:

procedure Ttarsor.SpeedButton1Click(Sender: TObject);

begin

//Сортировка по КОДУ

if (pole.ItemIndex=0) and (pole.ItemIndex=0) then

begin

dd.Query2.Close;

dd.Query2.SQL.Clear;

dd.Query2.SQL.Add('SELECT \* FROM TARIF ORDER BY tarkod');

dd.Query2.Open;

end;

if (pole.ItemIndex=0) and (pole.ItemIndex=1) then

begin

dd.Query2.Close;

dd.Query2.SQL.Clear;

dd.Query2.SQL.Add('SELECT \* FROM TARIF ORDER BY tarkod DESC');

dd.Query2.Open;

end;

//Сортировка по ГОРОДУ

if (pole.ItemIndex=1) and (pole.ItemIndex=0)then

begin

dd.Query2.Close;

dd.Query2.SQL.Clear;

dd.Query2.SQL.Add('SELECT \* FROM TARIF ORDER BY targor');

dd.Query2.Open;

end;

if (pole.ItemIndex=1) and (pole.ItemIndex=1)then

begin

dd.Query2.Close;

dd.Query2.SQL.Clear;

dd.Query2.SQL.Add('SELECT \* FROM TARIF ORDER BY targor DESC');

dd.Query2.Open;

end;

//Сортировка по ЦЕНЕ

if (pole.ItemIndex=2) and (pole.ItemIndex=0)then

begin

dd.Query2.Close;

dd.Query2.SQL.Clear;

dd.Query2.SQL.Add('SELECT \* FROM TARIF ORDER BY tarprice');

dd.Query2.Open;

end;

if (pole.ItemIndex=2) and (pole.ItemIndex=1)then

begin

dd.Query2.Close;

dd.Query2.SQL.Clear;

dd.Query2.SQL.Add('SELECT \* FROM TARIF ORDER BY tarprice DESC');

dd.Query2.Open;

end;

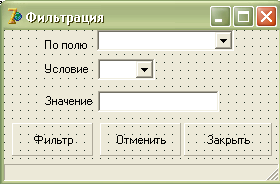
end;

Аналогично и для других полей таблицы.

## 

## 2.7. Фильтрация данных

Фильтрация выполняется на первой вкладке формы «MTTS». Нужно выбрать то условие, по которому нужно выполнить фильтрацию, можно вернуть все обратно.



Программа:

procedure Ttarfilter.SpeedButton1Click(Sender: TObject);

begin

if (pole.Text='') or (ysl.Text='') or (text.Text='') then

begin

showmessage('Выполнены не все условия для фильтрации');

end else

begin

if pole.ItemIndex=0 then

begin

dd.Query2.Filtered:=false;

dd.Query2.Filter:='TARkod'+ysl.Text+text.Text;

dd.Query2.Filtered:=true;

if dd.Query2.RecordCount=0 then

begin

showmessage('Не найдено ни одной записи'+#13+'Нет смысла фильтровать');

dd.Query2.Filtered:=false;

end;

end;

if pole.ItemIndex=1 then

begin

dd.Query2.Filtered:=false;

dd.Query2.Filter:='TARgor'+ysl.Text+''''+ text.Text+'''';

dd.Query2.Filtered:=true;

if dd.Query2.RecordCount=0 then

begin

showmessage('Не найдено ни одной записи'+#13+'Нет смысла фильтровать');

dd.Query2.Filtered:=false;

end;

end;

if pole.ItemIndex=2 then

begin

dd.Query2.Filtered:=false;

dd.Query2.Filter:='TARprice'+ysl.Text+''''+text.Text+'''';

dd.Query2.Filtered:=true;

if dd.Query2.RecordCount=0 then

begin

showmessage('Не найдено ни одной записи'+#13+'Нет смысла фильтровать');

dd.Query2.Filtered:=false;

end;

end;

sp.Panels[0].Text:='Результат фильтрации: '+floattostr(dd.Query2.RecordCount)+' зап.';

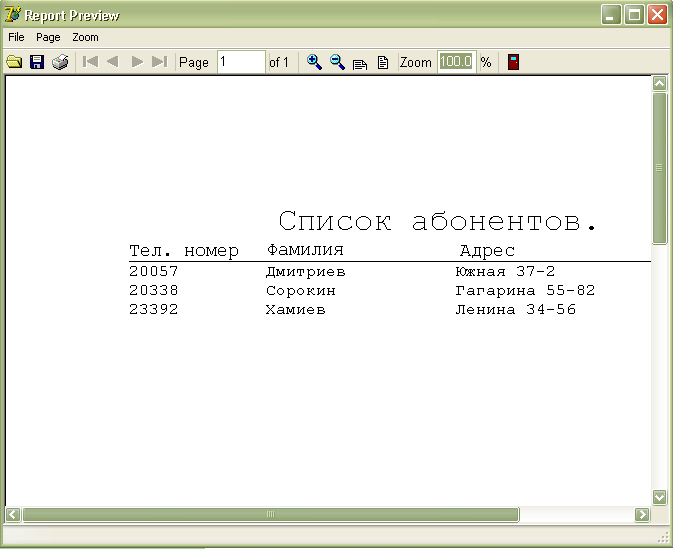
end;

end;

2.8. Создание отчета

По данной таблице создан отчет «Список абонентов». Для создания отчета нужно выбрать вкладку Rave компоненты RVProject, RTF и RVDataSetConection. На компоненте RVDataSetConection свойство DataSet нужно выбрать ту таблицу, которая должна быть представлена в виде отчета. После этого: Tools – Rave Designer – File – NewDataObject – DirectDataNiew – Finish. Tools – Report Wieardz – Simple Table – Next – All – Next – Geniered. Нужно изменить шрифт, форматировать отчет, сохранить. RVProject – Project File – название отчета.

В меню «Сервис» можно просмотреть отчет.



Программа для запуска отчета:

procedure Tdd.N32Click(Sender: TObject);

begin

dd.RvProject1.Execute;

end;

2.9. Изменение цвета формы и изменение шрифта

В меню «Формат» можно изменить цвет формы и изменить шрифт.

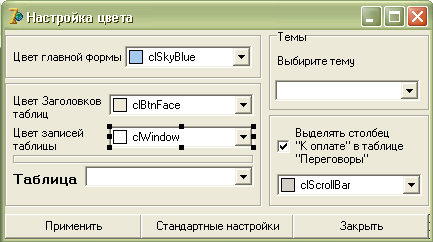
Программа для выполнения изменений цвета формы:

procedure Tdd.N55Click(Sender: TObject);

begin

capt.ShowModal;

end;



Программа для выполнения изменений шрифта:

procedure Tdd.N54Click(Sender: TObject);

begin

if dd.FontDialog1.Execute then

begin

main.DBGrid1.Font:=dd.FontDialog1.Font;

main.DBGrid2.Font:=dd.FontDialog1.Font;

main.DBGrid3.Font:=dd.FontDialog1.Font;

end;

end;

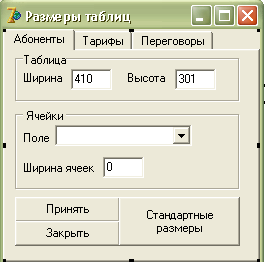
Программа для изменения размера таблиц:

procedure Tdd.N57Click(Sender: TObject);

begin

size.showmodal;

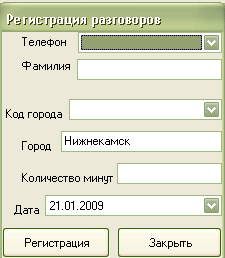
end;



## 

## 2.10. Регистрация разговоров

На форме «MTTS страница «Абоненты» можно произвести регистрацию разговоров.



Программа для регистрации:

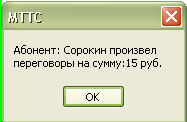
procedure Tdd.N12Click(Sender: TObject);

begin

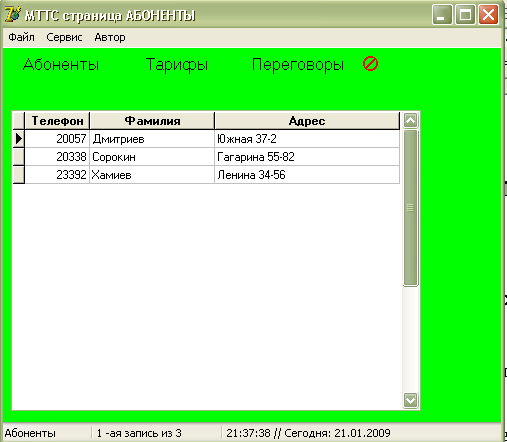
pereg.ShowModal;

end;

При нажатии на кнопку регистрация появляется следующая форма:

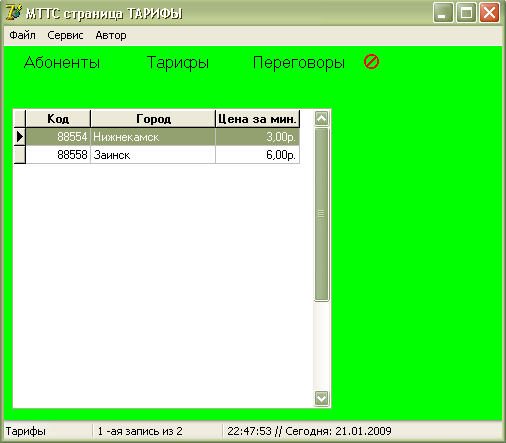


**3.** **Форма «MTTS страница «АБОНЕНТЫ»**



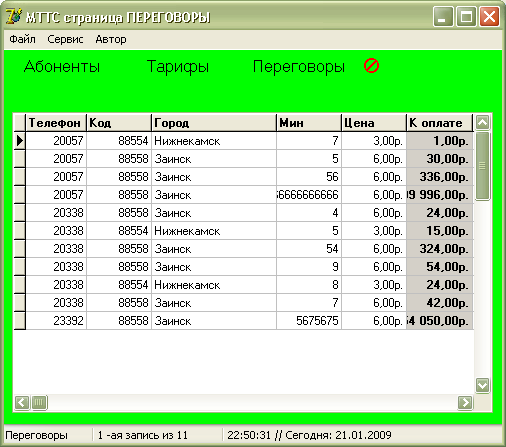
Форма «MTTS страница «Абоненты» содержится таблица «Абоненты». Она состоит из трех полей: телефон, фамилия и адрес. Можно выполнить регистрацию абонентов (отчет), добавление новых данных, изменение, удаление не нужных данных, выполнить в таблице поиск и фильтрацию по номеру абонента. Можно просмотреть отчет «Список абонентов».

**4. Форма «MTTS страница ТАРИФЫ»**



Форма «MTTS страница ТАРИФЫ» состоит из «Файла», где указаны справка, настройка (шрифт, цвет, размер таблиц), а также выход из самой программы. В меню «Сервис» указаны тарифы (добавить, изменить, удалить, поиск, сортировка, фильтрация.)

# **5. Форма «MTTS страница ПЕРЕГОВОРЫ»**



Форма «MTTS страница ПЕРГОВОРЫ» состоит из «Файла», где указаны справка, настройка (шрифт, цвет, размер таблиц), а также выход из самой программы. В меню «Сервис» указаны следующие операции по переговорам: выписать квитанцию, должники, поиск, сортировка, фильтрация.

**ЗАКЛЮЧЕНИЕ**

**база данные программирование delphi**

В данной курсовой выполнены все требования, а именно создание базы данных и СУБД для управления ею.

На основе проделанной работы можно сделать вывод о том, что язык программирования Delphi является мощной средой, позволяющая создавать различные банки данных, разрабатывать удобный интерфейс для работы с ними. Данный язык имеет огромное количество различных компонент: DBGrid, ADOTable, Query, DBEdit, ComboBox и т.д., которые позволяют работать с данными базы.

Все поставленные задачи, для разработки программы, выполнены. Программа представляет пользователю возможность просматривать все таблицы в нескольких видах, соответствующих сортировке таблицы по каждому из полей, а также при выполнении фильтрации. Также можно выполнить добавление новых данных, изменение, удаление не нужных данных, поиск, диапазон и т.д.

Все кнопки, расположенные на пользовательских формах, оснащены всплывающими подсказками, которые позволяют более точно истолковать назначение данных кнопок.

# **Список использованных источников**

1. А.Г. Исавнин, Д.М. Лысанов, В.С. Фрикк «Создание отчетов с помощью RAVE – компонентов Delphi», издательство ИНЭКА, Набережные Челны, 2006г.
2. Ш.Ш. Хузятов «Delphi7. Основные компоненты и элементы языка программирования», издательство КамПИ, Набережные Челны, 2005г.
3. В.В. Фаронов «Delphi7 Программирование на языке высокого уровня». Учебник для вузов. 2007г.