

Министерство образования и науки РФ
ФГБПОУ ВПО Тульский государственный университет
КАФЕДРА АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ЗАПИСЕЙ И ФАЙЛОВ

Лабораторная работа № 5
по курсу «Программирование на ЯВУ»

Вариант № 4

Выполнил: студент группы 220601

_____ Белым А.А.
(подпись)

Проверил: к. ф.-м. н., доцент

_____ Сулимова В.В.
(подпись)

Тула 2011

Цель работы

Цель работы заключается в том, чтобы изучить правила работы с компонентами TOpenDialog и TSaveDialog и написать программу с использованием файлов и данных типа "запись".

Задание на работу

В справочной автовокзала хранится расписание движения автобусов. Для каждого рейса указаны его номер, тип автобуса, пункт назначения, время отправления и прибытия. Вывести информацию о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени.

[illegible]

[illegible]

[illegible]

Схема алгоритма

На рисунке 1 представлена схема общего алгоритма вывода информации о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени.

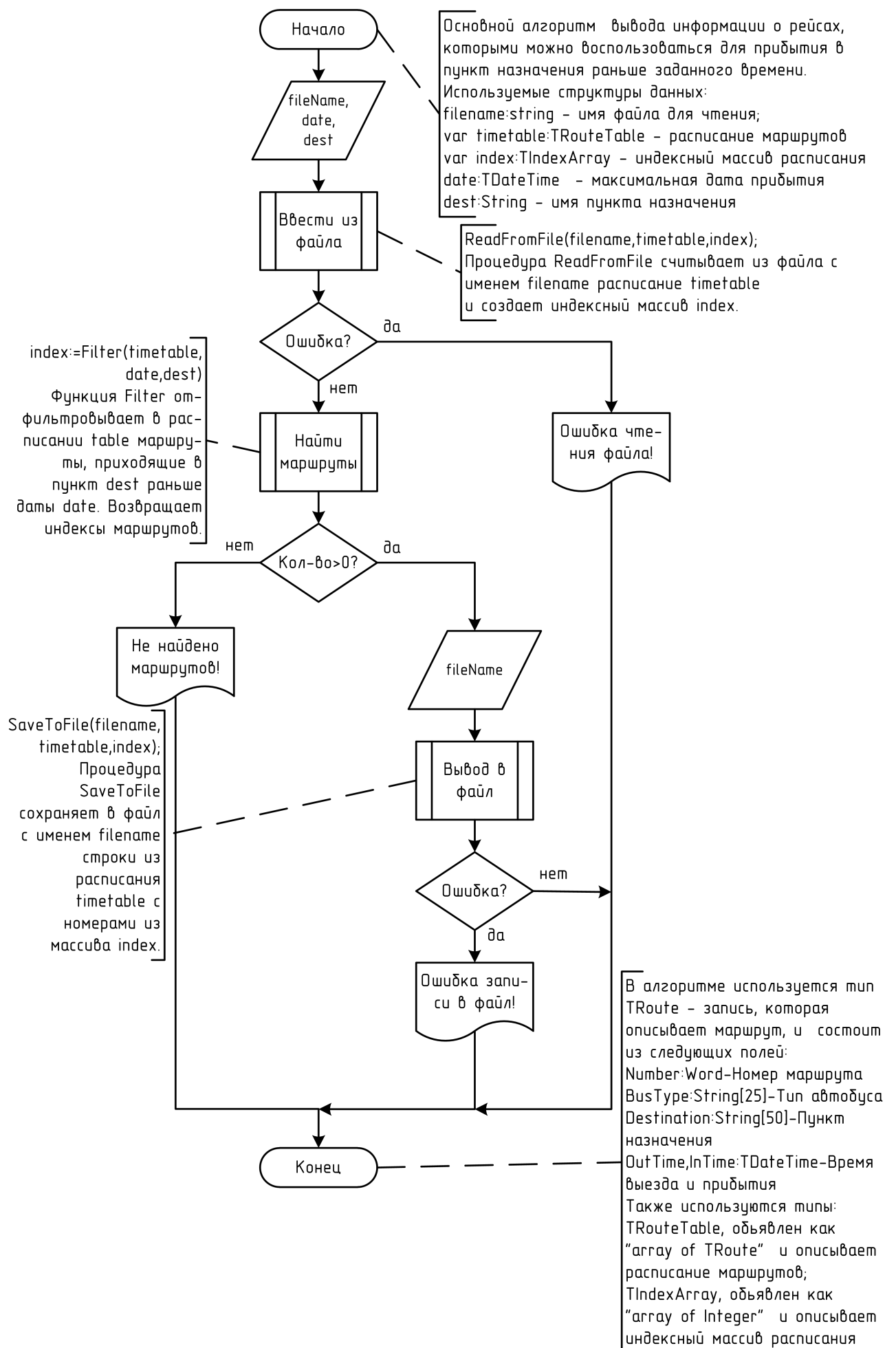


Рисунок 1 - Схема общего алгоритма вывода информации о рейсах

На рисунке 2 представлена схема алгоритма загрузки расписания из файла и инициализации индексного массива.

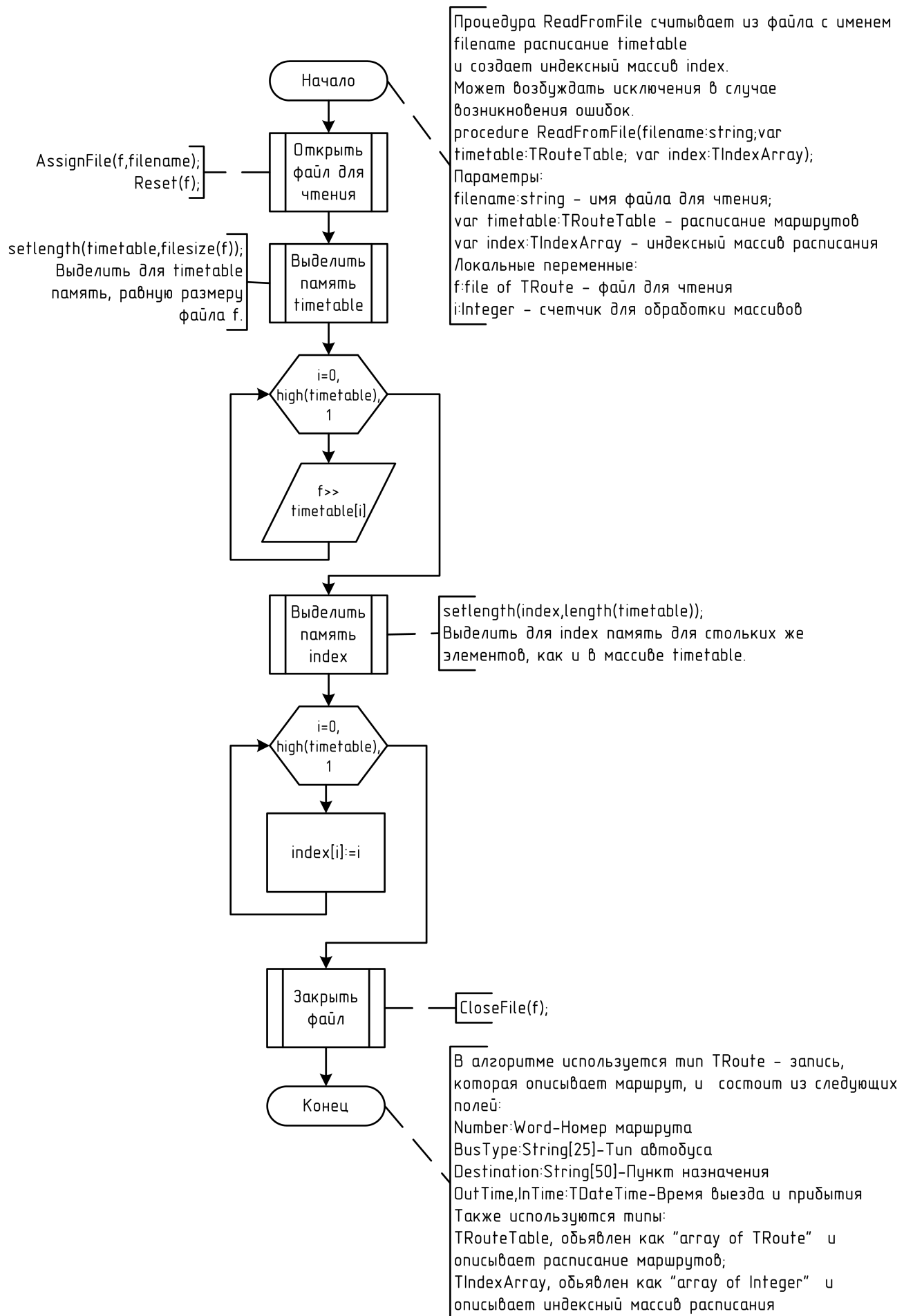


Рисунок 2 - Схема алгоритма загрузки расписания из файла

воспользоваться для прибытия в пункт назначения раньше заданного времени,.

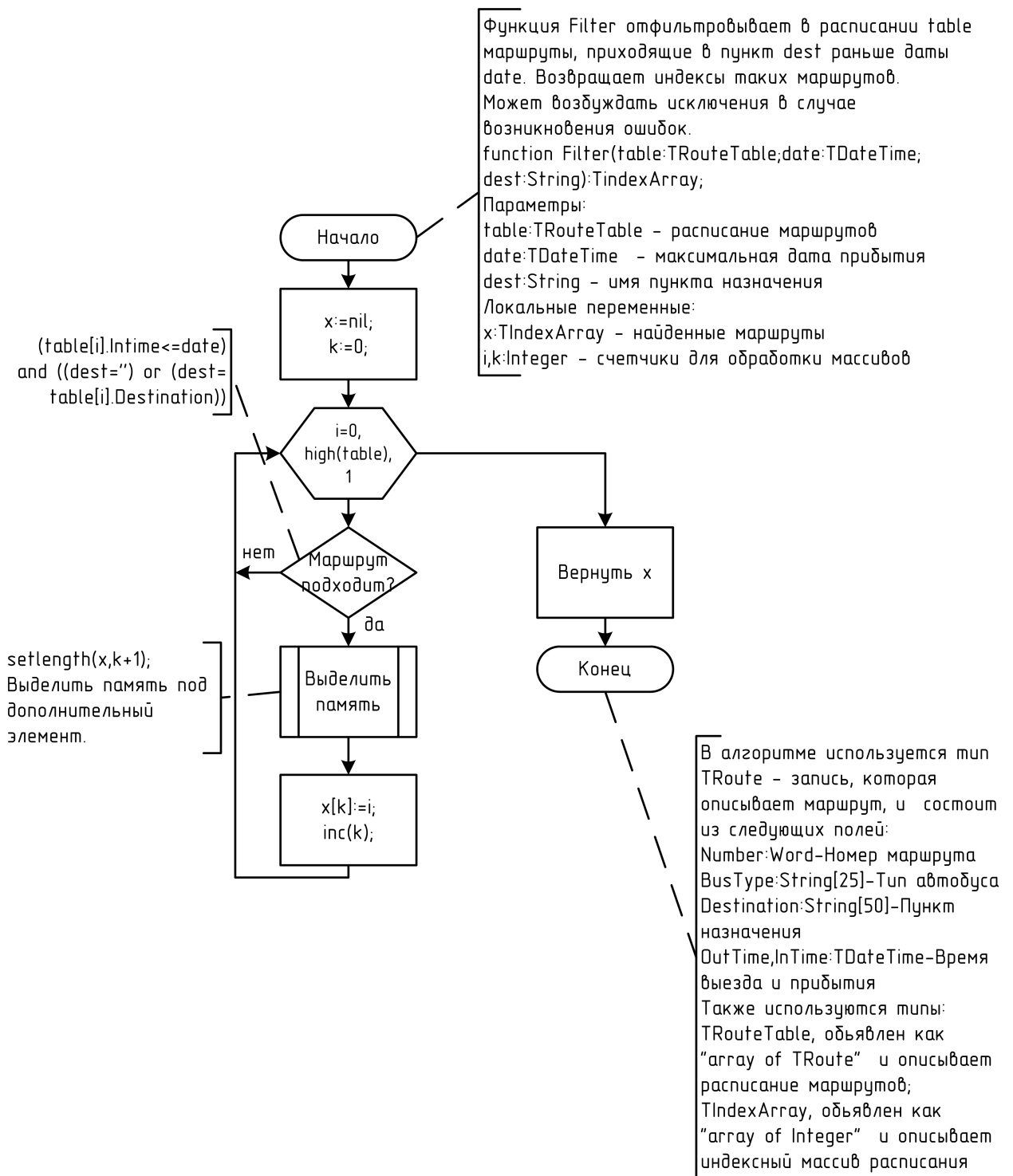


Рисунок 3 - Схема алгоритма поиска рейсов

На рисунке 4 представлена схема алгоритма сохранения найденных маршрутов в файл.

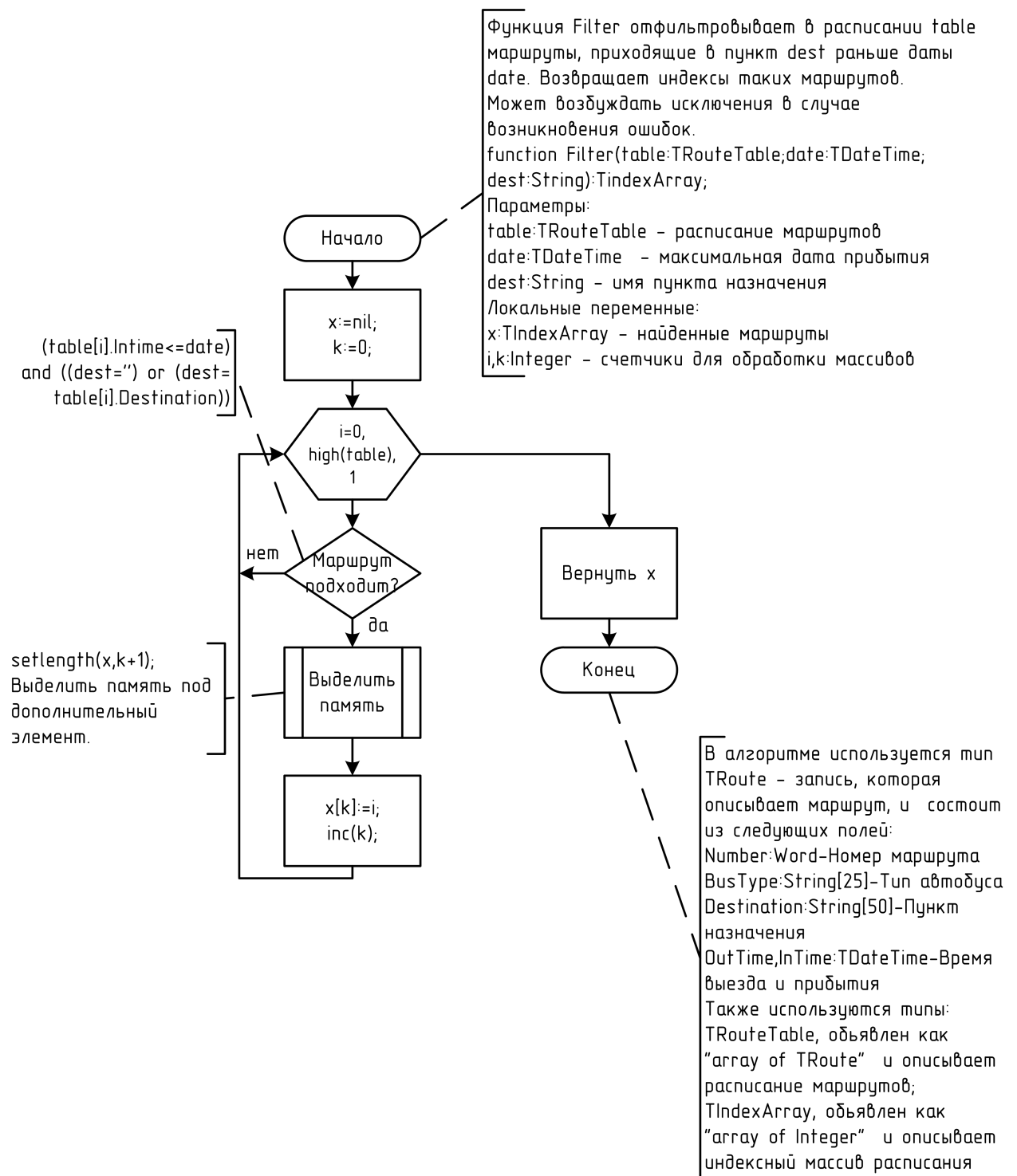


Рисунок 4 - Схема алгоритма сохранения маршрутов

Инструкция пользователю

Программа позволяет создавать, редактировать, открывать и сохранять расписания автобусных маршрутов, а также осуществлять вывод информации о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени.

Сверху в окне программы располагаются кнопки создания нового расписания, открытия и сохранения расписания.

Основную часть окна занимает метка расписания. Самая верхняя строка сетки подсказывает предназначение каждой ячейки таблицы. Любую ячейку можно свободно редактировать. При редактировании ячеек с датами появляется вспомогательное окошко, в котором можно выбрать дату и время; если вы хотите ввести данные параметры вручную, нажмите кнопку "Отмена". При неправильном вводе при попытке любых операций с таблицей программа сообщит координаты ячеек с неправильными значениями.

Нижнюю часть программы занимают элементы фильтрации, которые позволяют осуществить вывод информации о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени. Просто введите нужные параметры и нажмите кнопку "Отфильтровать чтобы просмотреть результаты; кнопку "Сбросить фильтр чтобы получить полное расписание и кнопку "Сохранить" для сохранения найденных маршрутов в файл.

В случае возникновения непредвиденной ошибки программа выводит сообщение и блокирует все элементы управления, кроме кнопок создания и открытия расписания. Если не получается открыть или создать расписание, перезапустите программу.

Инструкция программиста

При создании программы вывода информации о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени, были объявлены типы:

1. TRoute - запись, которая описывает маршрут, и состоит из полей, представленных в таблице 1.

Таблица 1 - Поля типа "маршрут"

имя	тип	предназначение
Number	Word	Номер маршрута
BusType	String[25]	Тип автобуса
Destination	String[50]	Пункт назначения
OutTime,InTime	TDateTime	Время выезда и прибытия

2. TRouteTable, объявлен как "array of TRoute"и описывает расписание маршрутов;

3. TIndexArray, объявлен как "array of Integer"и описывает индексный массив расписания.

Также описан класс THackStringGrid=class(TStringGrid), который описывает вариант таблицы TStringGrid с некоторыми дополнительными полезными методами.

Были объявлены следующие глобальные переменные:

Таблица 2 - Глобальные переменные модуля

имя	тип	предназначение
timetable	TRouteTable	Расписание маршрутов
index	TIndexArray	Индексный массив расписания

Далее программа была разбита на следующие подпрограммы:

1. Процедура ReadFromFile считывает из файла с именем filename расписание timetable и создает индексный массив index. Может возбуждать исключения в случае возникновения ошибок.

```
procedure ReadFromFile(filename:string;var timetable:TRouteTable;
var index:TIndexArray);
```

Параметры процедуры представлены в таблице 3 :

Таблица 3 - Параметры процедуры чтения расписания

имя	тип	предназначение
filename	string	имя файла для чтения
var timetable	TRouteTable	расписание маршрутов
var index	TIndexArray	индексный массив расписания

Локальные переменные процедуры представлены в таблице 4 :

Таблица 4 - Локальные переменные процедуры чтения расписания

имя	тип	предназначение
f	file of TRoute	файл для чтения
i	Integer	счетчик для обработки массивов

2. Функция Filter отфильтровывает в расписании table маршруты, приходящие в пункт dest раньше даты date. Возвращает индексы таких маршрутов. Может возбуждать исключения в случае возникновения ошибок.

```
function Filter(table:TRouteTable;date:TDateTime;dest:String):TIndexArray;
```

Параметры функции представлены в таблице 5 :

Таблица 5 - Параметры функции фильтрации расписания

имя	тип	предназначение
table	TRouteTable	расписание маршрутов
date	TDateTime	максимальная дата прибытия
dest	String	имя пункта назначения

Локальные переменные функции представлены в таблице 6 :

Таблица 6 - Локальные переменные функции фильтрации расписания

имя	тип	предназначение
x	TIndexArray	найденные маршруты
i,k	Integer	счетчики для обработки массивов

3. Процедура SaveToFile сохраняет в файл с именем filename строки из расписания timetable с номерами из массива index. Может возбуждать исключения в случае возникновения ошибок.

```
procedure SaveToFile(filename:string; timetable:tRouteTable; index:TIndexArray);
```

Параметры процедуры представлены в таблице 7 :

Таблица 7 - Параметры процедуры сохранения расписания

имя	тип	предназначение
filename	string	имя файла для записи
timetable	tRouteTable	расписание маршрутов
index	TIndexArray	индессы строк расписания для записи

Локальные переменные процедуры представлены в таблице 8 :

Таблица 8 - Локальные переменные процедуры сохранения расписания

имя	тип	предназначение
f	file of TRoute	файл для записи
i	Integer	счетчик для обработки массивов

4. Процедура btnDelLineClick. Обработчик нажатия кнопки удаления строки.

```
procedure TfrmMain.btnDelLineClick(Sender: TObject);
```

Параметры процедуры представлены в таблице 9 :

Таблица 9 - Параметры процедуры-обработчика события нажатия кнопки удаления строки

имя	тип	предназначение
Sender	TObject	объект-возбудитель события

5. Процедура btnAddLineClick. Обработчик нажатия кнопки добавления строки.

```
procedure TfrmMain.btnAddLineClick(Sender: TObject);
```

Параметры процедуры представлены в таблице 10 :

Таблица 10 - Параметры процедуры-обработчика события нажатия кнопки добавления строки

имя	тип	предназначение
Sender	TObject	объект-возбудитель события

6. Процедура FormCreate. Обработчик события создания формы.

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

Параметры процедуры представлены в таблице 11 :

Таблица 11 - Параметры процедуры-обработчика события создания формы

имя	тип	предназначение
Sender	TObject	объект-возбудитель события

7. Процедура btnFilterClick. Обработчик нажатия кнопки фильтрации.

```
procedure TfrmMain.btnFilterClick(Sender: TObject);
```

Параметры процедуры представлены в таблице 12 :

Таблица 12 - Параметры процедуры-обработчика события нажатия кнопки фильтрации

имя	тип	предназначение
Sender	TObject	объект-возбудитель события

8. Процедура UpdateGrid. Метод обновления таблицы из расписания.

```
procedure TfrmMain.UpdateGrid;
```

9. Процедура updatetable. Метод обновления расписания из таблицы.

```
procedure TfrmMain.updatetable;
```

10. Процедура btnNoFilterClick. Обработчик нажатия кнопки отмены фильтрации.

```
procedure TfrmMain.btnNoFilterClick(Sender: TObject);
```

Параметры процедуры представлены в таблице 13 :

Таблица 13 - Параметры процедуры-обработчика события нажатия кнопки отмены фильтрации

имя	тип	предназначение
Sender	TObject	объект-возбудитель события

11. Процедура `tbtNewClick`. Обработчик нажатия кнопки создания расписания.

```
procedure TfrmMain.tbtNewClick(Sender: TObject);
```

Параметры процедуры представлены в таблице 14 :

Таблица 14 - Параметры процедуры-обработчика события нажатия кнопки создания расписания

имя	тип	предназначение
Sender	TObject	объект-возбудитель события

12. Процедура `tbtSaveClick`. Обработчик нажатия кнопки сохранения расписания.

```
procedure TfrmMain.tbtSaveClick(Sender: TObject);
```

Параметры процедуры представлены в таблице 15 :

Таблица 15 - Параметры процедуры-обработчика события нажатия кнопки сохранения расписания

имя	тип	предназначение
Sender	TObject	объект-возбудитель события

13. Процедура `tbtOpenClick`. Обработчик нажатия кнопки открытия расписания.

```
procedure TfrmMain.tbtOpenClick(Sender: TObject);
```

Параметры процедуры представлены в таблице 16 :

Таблица 16 - Параметры процедуры-обработчика события нажатия кнопки открытия расписания

имя	тип	предназначение
Sender	TObject	объект-возбудитель события

14. Процедура sgdTimetableMouseUp. Обработчик отпускания клавиши мыши на таблице.

```
procedure TfrmMain.sgdTimetableMouseUp ( Sender: TObject;  
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
```

Параметры процедуры представлены в таблице 17 :

Таблица 17 - Параметры процедуры-обработчика события отпускания клавиши мыши на
таблице

имя	тип	предназначение
Sender	TObject	объект-возбудитель события
Button	TMouseButton	состояние кнопок мыши
Shift	TShiftState	состояние кнопки Shift
X, Y	Integer	координаты курсора

15. btnSaveFilteredClick. Обработчик нажатия кнопки сохранения отфильтрованных маршрутов.

```
procedure TfrmMain.btnSaveFilteredClick(Sender: TObject);
```

Параметры процедуры представлены в таблице 18 :

Таблица 18 - Параметры процедуры-обработчика события нажатия кнопки сохранения
отфильтрованных маршрутов

имя	тип	предназначение
Sender	TObject	объект-возбудитель события

16. Процедура FormDestroy. Обработчик события уничтожения формы.

```
procedure TfrmMain.FormDestroy(Sender: TObject);
```

Параметры процедуры представлены в таблице 19 :

Таблица 19 - Параметры процедуры-обработчика события уничтожения формы

имя	тип	предназначение
Sender	TObject	объект-возбудитель события

Текст программы

Далее приведен текст программы вывода информации о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени, написанной на языке Delphi 7.

```
unit UnitMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ImgList, ComCtrls, ToolWin, StdCtrls, Grids, UnitDlg;

type
  TRoute=record //Маршрут
    Number:Word; //Номер маршрута
    BusType:String[25]; //Тип автобуса
    Destination:String[50]; //Пункт назначения
    OutTime,InTime:TDateTime; //Время выезда и прибытия
  end;
  TRouteTable=array of TRoute; //Расписание маршрутов
  TIndexArray=array of Integer; //Индексный массив расписания

THackStringGrid = class(TStringGrid); //Таблица с некоторыми полезными методами
TfrmMain = class(TForm) //Основная форма
  tlbFileOps: TToolBar; //Панель инструментов
  tbtNew: TToolButton; //Кнопка панели "Создать расписание"
  imlToolbar: TImageList; //Список изображений для панели
  tbtOpen: TToolButton; //Кнопка панели "Открыть расписание"
  tbtSave: TToolButton; //Кнопка панели "Сохранить расписание"
  grpFilter: TGroupBox; //Группа виджетов фильтрации
  dtpDate: TDateTimePicker; //Выбор даты
  grpTimeTable: TGroupBox; //Группа виджетов, отображающих расписание
  sgdtimetable: TStringGrid; //Таблица расписания
  btnAddLine: TButton; //кнопка "Добавить строку"
  btnDelLine: TButton; //кнопка "Удалить строку"
  dtpTime: TDateTimePicker; //выбор времени
  btnFilter: TButton; //кнопка "Отфильтровать"
  btnNoFilter: TButton; //Кнопка "Отменить фильтр"
  sdgSaveFile: TSaveDialog; //Диалог сохранения файла
  odgOpenFile: TOpenDialog; //Диалог открытия файла
  lblInfoDate: TLabel; //Надпись—указание о выборе даты
  lblInfoDest: TLabel; //Надпись—указание о выборе пункта назначения
  edtDestination: TEdit; //Ввод пункта назначения
  lblProgInfo: TLabel; //Информация о программе
  btnSaveFiltered: TButton; //Кнопка сохранения результатов фильтрации

  //Обработчик нажатия кнопки удаления строки
  procedure btnDelLineClick(Sender: TObject);
  //Обработчик нажатия кнопки добавления строки
  procedure btnAddLineClick(Sender: TObject);
  //Обработчик события создания формы
  procedure FormCreate(Sender: TObject);
  //Обработчик нажатия кнопки фильтрации
  procedure btnFilterClick(Sender: TObject);
  //Метод обновления таблицы из расписания
  procedure UpdateGrid;
  //Метод обновления расписания из таблицы
  procedure updatetable;
  //Обработчик нажатия кнопки отмены фильтрации
  procedure btnNoFilterClick(Sender: TObject);
  //Обработчик нажатия кнопки создания расписания
```

```

procedure tbtNewClick(Sender: TObject);
//Обработчик нажатия кнопки созранения расписания
procedure tbtSaveClick(Sender: TObject);
//Обработчик нажатия кнопки открытия расписания
procedure tbtOpenClick(Sender: TObject);
//Обработчик отпускания клавиши мыши на таблице
procedure sgdTimetableMouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
//Обработчик нажатия кнопки сохранения отфильтрованных маршрутов
procedure btnSaveFilteredClick(Sender: TObject);
//Обработчик события уничтожения формы
procedure FormDestroy(Sender: TObject);

private
    { Private declarations }

public
    { Public declarations }
end;

//Процедура чтения расписания из файла
procedure ReadFromFile(filename:string;var timetable:TRouteTable;
    var index:TIndexArray);

//Процедура фильтрации маршрутов
function Filter(table:TRouteTable;date:TDateTime;dest:String):TIndexArray;

//Процедура сохранения расписания в файл
procedure SaveToFile(filename:string; timetable:tRouteTable; index:TIndexArray);

var
    frmMain: TfrmMain; //Основная форма
    timetable:TRouteTable; //Расписание маршрутов
    index:TIndexArray; //Индексный массив расписания
implementation
    (*
    Процедура ReadFromFile считывает из файла с именем filename расписание timetable
    и создает индексный массив index.
    Может возбуждать исключения в случае возникновения ошибок.
    Параметры:
    filename:string – имя файла для чтения;
    var timetable:TRouteTable – расписание маршрутов
    var index:TIndexArray – индексный массив расписания
    Локальные переменные:
    f:file of TRoute – файл для чтения
    i:Integer – счетчик для обработки массивов
    *)
procedure ReadFromFile(filename:string;var timetable:TRouteTable;
    var index:TIndexArray);
var f:file of TRoute; i:Integer;
begin
    AssignFile(f,filename);
    Reset(f);
    try
        setlength(timetable,filesize(f));
        For i:=0 to high(timetable) do
            Read(f,timetable[i]);

        setlength(index,length(timetable));
        for i:=0 to high(timetable) do
            index[i]:=i;
        finally CloseFile(f);
    end;
end;
    (*

```

Функция *Filter* отфильтровывает в расписании *table* маршруты, приходящие в пункт *dest* раньше даты *date*. Возвращает индексы таких маршрутов.

Может возбуждать исключения в случае возникновения ошибок.

Параметры:

table:TRouteTable – расписание маршрутов

date:TDateTime – максимальная дата прибытия

dest:String – имя пункта назначения

Локальные переменные:

x:TIndexArray – найденные маршруты

i,k:Integer – счетчики для обработки массивов

*)

```
function Filter(table:TRouteTable;date:TDateTime;dest:String):TIndexArray;  
var x:TIndexArray; i,k:Integer;  
begin  
  x:=nil;  
  k:=0;  
  for i:=0 to high(table) do begin  
    if (table[i].Intime<=date) and ((dest='') or (dest=table[i].Destination))  
    then begin  
      setlength(x,k+1);  
      x[k]:=i;  
      inc(k);  
    end;  
  end;  
  result:=x;  
end;
```

(*

Процедура *SaveToFile* сохраняет в файл с именем *filename* строки из расписания *timetable* с номерами из массива *index*.

Может возбуждать исключения в случае возникновения ошибок.

Параметры:

filename:string – имя файла для записи

timetable:tRouteTable – расписание маршрутов

index:TIndexArray – индесы строк расписания для записи

Локальные переменные:

f:file of TRoute – файл для записи

i:Integer – счетчик для обработки массивов

*)

```
procedure SaveToFile(filename:string; timetable:tRouteTable; index:TIndexArray);  
var f:file of TRoute; i:Integer;  
begin  
  Assign(f,filename);  
  Rewrite(f);  
  try  
    for i:=0 to length(index)-1 do  
      Write(f,timetable[index[i]]);  
    finally CloseFile(f);  
  end;  
end;  
{ $R *.dfm }  
(*)
```

```
procedure DeleteIndex(var anArray:TIndexArray; const aPosition:integer);  
var  
  lg, j : integer;  
begin  
  lg := length(anArray);  
  if aPosition > lg-1 then  
    exit  
  else if aPosition = lg-1 then begin  
    Setlength(anArray, lg -1);  
    exit;  
  end;  
  for j := aPosition to lg-2 do  
    anArray[j] := anArray[j+1]-1;
```

```

        SetLength(anArray, lg-1);

end;
(*
*)
procedure DeleteRoute(var anArray:TRouteTable; const aPosition:integer);
var
    lg, j : integer;
begin
    lg := length(anArray);
    if aPosition > lg-1 then
        exit
    else if aPosition = lg-1 then begin
        Setlength(anArray, lg -1);
        exit;
    end;
    for j := aPosition to lg-2 do
        anArray[j] := anArray[j+1];
    SetLength(anArray, lg-1);

end;
procedure TfrmMain.updatetable;
var i,j,k:integer; errors:string; next:Boolean;
begin
    errors:='';
    for i:=1 to sgdTimetable.RowCount-1 do
        for j:=0 to sgdTimetable.ColCount-1 do
            try
                case j of
                    0: begin
                        for k:=1 to sgdTimetable.RowCount-1 do
                            if (strtoint(sgdTimetable.Cells[j,i])=strtoint(sgdTimetable.Cells[j,k]))
                                and (i<>k)
                                then raise EConvertError.Create('Дубликат маршрута!');
                            timetable[index[i-1]].Number:=strtoint(sgdTimetable.Cells[j,i]);
                        end;
                    1:if sgdTimetable.Cells[j,i]<> '' then
                        timetable[index[i-1]].BusType:=sgdTimetable.Cells[j,i]
                    else raise EConvertError.Create('Пустая строка!');
                    2:if sgdTimetable.Cells[j,i]<> '' then
                        timetable[index[i-1]].destination:=sgdTimetable.Cells[j,i]
                    else raise EConvertError.Create('Пустая строка!');
                    3:begin
                        try
                            strtoDateTime(sgdTimetable.Cells[j+1,i]);
                            next:=False
                        except else
                            next:=True;
                        end;
                    if not next and
                        (strtoDateTime(sgdTimetable.Cells[j+1,i])<=
                            strtoDateTime(sgdTimetable.Cells[j,i])) then
                        raise EConvertError.Create('TIME PARADOX!') else

                            timetable[index[i-1]].outtime:=strtodatetime(sgdTimetable.Cells[j,i])
                    end;
                    4:begin
                        try
                            strtoDateTime(sgdTimetable.Cells[j-1,i]);
                            next:=False;
                        except else
                            next:=True;
                        end;
                    if not next and
                        (strtoDateTime(sgdTimetable.Cells[j-1,i])>=

```

```

        strtodateTime(sgdTimetable.Cells[j,i])) then
            raise EConvertError.Create('TIME PARADOX!') else
timetable[index[i-1]].intime:=strtodateTime(sgdTimetable.Cells[j,i])
        end;
    end;
except on E:EConvertError do
    errors:=errors+Format('Неправильное значение в ячейке [%d,%d]! '+e.
        Message +#10#13,[i,j+1]);
end;
if errors<>'' then
    raise EConvertError.Create('Ошибки при заполнении расписания:'+ #10#13+errors);
end;

procedure TfrmMain.btnDelLineClick(Sender: TObject);
var i:Integer; x:boolean;
begin
    try
        x:=false;
        try
            if (sgdTimetable.Selection.Top=1) and
                (sgdTimetable.Selection.Bottom=sgdTimetable.RowCount-1) then x:=true;
            for i:=sgdTimetable.Selection.Top to sgdTimetable.Selection.Bottom do begin

                sgdTimetable.Rows[i].Clear;
                THackStringGrid(sgdTimetable).DeleteRow(i);
                DeleteRoute(timetable,index[i-1]);
                DeleteIndex(index,i-1);
            end;
        except else
            if (sgdTimetable.Selection.Top=1) and
                (sgdTimetable.Selection.Bottom=sgdTimetable.RowCount-1) then x:=true
            else begin
                setlength(timetable,
                    sgdTimetable.RowCount+
                    sgdTimetable.Selection.Top-
                    sgdTimetable.Selection.Bottom-1);
                setlength(index,
                    sgdTimetable.RowCount+
                    sgdTimetable.Selection.Top-
                    sgdTimetable.Selection.Bottom-1);
            end;
            for i:=sgdTimetable.Selection.Top to sgdTimetable.Selection.Bottom do
                THackStringGrid(sgdTimetable).DeleteRow(i);
            end;
            if x then
                begin

                    sgdTimetable.RowCount:=sgdTimetable.RowCount+1;
                    sgdTimetable.FixedRows:=1;
                    setlength(timetable,length(timetable)+1);setlength(index,length(index)+1);
                    index[high(index)]:=high(timetable);
                end;
        except else
            MessageDlg('Ошибка удаления строки!',mtError,[mbOk],0);
            grpFilter.Enabled:=false;
            grpTimetable.Enabled:=false;
            tbtSave.Enabled:=False;
        end;
    end;

procedure TfrmMain.btnAddLineClick(Sender: TObject);
var sel:TGridRect;
begin
    try
        sgdTimetable.RowCount:=sgdTimetable.RowCount+1;

```

```

sgdTimetable.Rows[sgdTimetable.RowCount-1].Clear;
with sel do begin
  Top:=sgdTimetable.RowCount-1;
  Bottom:=Top;
  Left:=0;Right:=0;
end;
setlength(timetable,length(timetable)+1);setlength(index,length(index)+1);
index[high(index)]:=high(timetable);
sgdTimetable.Selection:=sel;
except else MessageDlg('Ошибка добавления строки!',mtError,[mbOk],0);
grpFilter.Enabled:=false;
  grpTimetable.Enabled:=false;
  tbtSave.Enabled:=False;
end;
end;

procedure TfrmMain.FormCreate(Sender: TObject);
begin
  try
    sgdTimetable.Cells[0,0]:='№';sgdTimetable.ColWidths[0]:=30;
    sgdTimetable.Cells[1,0]:='Тип автобуса';sgdTimetable.ColWidths[1]:=90;
    sgdTimetable.Cells[2,0]:='Пункт назначения';sgdTimetable.ColWidths[2]:=115;
    sgdTimetable.Cells[3,0]:='Дата отправления';sgdTimetable.ColWidths[3]:=120;
    sgdTimetable.Cells[4,0]:='Дата прибытия';sgdTimetable.ColWidths[4]:=120;
    setlength(timetable,1);setlength(index,1);
    index[0]:=0;
  except else MessageDlg('Ошибка инициализации!',mtError,[mbOk],0);
  grpFilter.Enabled:=false;
    grpTimetable.Enabled:=false;
    tbtSave.Enabled:=False;
  end;
end;

procedure TfrmMain.UpdateGrid;
var i:integer;
begin
  if length(index)<>0 then begin
    sgdTimetable.RowCount:=length(index)+1;
    for i:=1 to length(index) do begin
      sgdTimetable.Cells[0,i]:= inttostr(timetable[index[i-1]].number);
      sgdTimetable.Cells[1,i]:= timetable[index[i-1]].BusType;
      sgdTimetable.Cells[2,i]:= timetable[index[i-1]].Destination;
      sgdTimetable.Cells[3,i]:=
        copy(datetimetostr(timetable[index[i-1]].OutTime),
          1,length(datetimetostr(timetable[index[i-1]].OutTime))-3);
      sgdTimetable.Cells[4,i]:=
        copy(datetimetostr(timetable[index[i-1]].inTime),1,
          length(datetimetostr(timetable[index[i-1]].inTime))-3);
    end;
  end;
end;

procedure TfrmMain.btnFilterClick(Sender: TObject);
var deadline:TDateTime; dest:string;newindex:TIndexArray;
begin
  newindex:=nil;
  try
    dtpTime.Date:=dtpDate.Date;
    deadline:=dtpTime.DateTime;dest:=edtDestination.Text;
    updatetable;
    newindex:=Filter(timetable,deadline,dest);

    if length(newindex)=0 then
      MessageDlg('Рейсов нет!',mtInformation,[mbok],0)
  
```



```

    else begin
        index:=nil;
        index:=newindex;
        newindex:=nil;
        UpdateGrid;
        btnNoFilter.Enabled:=True;
        btnSaveFiltered.Enabled:=True;
        btnAddLine.Enabled:=False;
    end;
except on e:EConvertError do
    MessageDlg(e.Message,mtError,[mbOk],0);
else MessageDlg('Ошибка фильтрации!',mtError,[mbOk],0);
grpFilter.Enabled:=false;
grpTimetable.Enabled:=false;
tbtSave.Enabled:=False;
end;
end;

procedure TfrmMain.btnNoFilterClick(Sender: TObject);
var i:integer;
begin
    try
        updatetable;
        setlength(index,length(timetable));
        for i:=0 to high(timetable) do
            index[i]:=i;

        UpdateGrid;
        try
            updatetable;
        except on e:EConvertError do
            MessageDlg(e.Message,mtWarning,[mbOk],0);
        end;
        btnNoFilter.Enabled:=False;
        btnSaveFiltered.Enabled:=False;
        btnAddLine.Enabled:=True;
        except on e:EConvertError do
            MessageDlg(e.Message,mtError,[mbOk],0);
        else MessageDlg('Ошибка фильтрации!',mtError,[mbOk],0);
        grpFilter.Enabled:=false;
        grpTimetable.Enabled:=false;
        tbtSave.Enabled:=False;
    end;
end;

procedure TfrmMain.tbtNewClick(Sender: TObject);
var sel:TGridRect;
begin
    try
        with sel do begin
            Top:=1;
            Bottom:=sgdTimetable.RowCount-1;
            Left:=0;Right:=0;
        end;
        sgdTimetable.Selection:=sel;
        btnDelLineClick(Sender);
        grpFilter.Enabled:=true;
        grpTimetable.Enabled:=true;
        tbtSave.Enabled:=true;
        btnNoFilter.Enabled:=False;
        btnSaveFiltered.Enabled:=False;
    except else MessageDlg('Ошибка создания расписания!',mtError,[mbOk],0);
    grpFilter.Enabled:=false;
    grpTimetable.Enabled:=false;
    tbtSave.Enabled:=False;
end;

```

```

    end;
end;

procedure TfrmMain.tbtsaveClick(Sender: TObject);
var i:Integer; save:Boolean; allindex:TIndexArray;
begin
    try
        try
            updatetable;
            save:=true
        except on e:EConvertError do
            save:=MessageDlg(e.Message+
                'Продолжить сохранение?'+#10#13,
                mtWarning, [mbok,mbcancel],0)=1;
        end;
        if save then
            if btnNoFilter.Enabled then
                save:=MessageDlg(
                    'На экране показана лишь часть расписания!'+#10#13#10#13+
                    'После проведения фильтрации на экране отображается лишь часть расписания,'+
                    #10#13+
                    'однако в файл расписание будет сохраняться полностью.'+#10#13+
                    'Продолжить сохранение?'+#10#13,
                    mtWarning, [mbok,mbcancel],0)=1;
            if save and sdgSaveFile.Execute then begin
                setlength(allindex,length(timetable));
                for i:=0 to length(timetable)-1 do
                    allindex[i]:=i;
                SaveToFile(sdgSaveFile.FileName,timetable,allindex);
                allindex:=nil;
            end;
        except else
            MessageDlg('Ошибка записи в файл!',mtError,[mbOk],0);
        end;
    end;

procedure TfrmMain.tbtopenClick(Sender: TObject);
begin
    try
        if odgOpenFile.Execute then begin
            ReadFromFile(odgOpenFile.FileName,timetable,index);
            UpdateGrid;
            btnNoFilter.Enabled:=False;
            btnSaveFiltered.Enabled:=False;
            grpFilter.Enabled:=true;
            grpTimetable.Enabled:=true;
            tbtsave.Enabled:=true;
        end;
        except else
            MessageDlg('Ошибка чтения файла!',mtError,[mbOk],0);
            grpFilter.Enabled:=false;
            grpTimetable.Enabled:=false;
            tbtsave.Enabled:=False;
        end;
    end;

procedure TfrmMain.sgdTimetableMouseUp(Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var acol,arow:integer;
begin
    sgdTimetable.MouseToCell(X,Y,acol,arow);
    if (ACol=3) or (ACol=4) then
        if dlgChooseDT.ShowModal=mrOk then begin
            sgdTimetable.Cells[ACol,ARow]:=
                copy(DateTimeToStr(dlgChooseDT.Selected),1,

```

```

        length(DateTimeToStr(dlgChooseDT.Selected))-3);
    end
    else THackStringGrid(sgdTimetable).EditorMode:=true;

end;

procedure TfrmMain.btnSaveFilteredClick(Sender: TObject);
var save:Boolean;
begin
    try
        try
            updatetable;
            save:=true
        except on e:EConvertError do
            save:=MessageDlg(e.Message+
                'Продолжить сохранение?'+#10#13,
                mtWarning, [mbok,mbcancel], 0)=1;
        end;
        if save and sgdSaveFile.Execute then begin
            SaveToFile(sgdSaveFile.FileName, timetable, index);
        end;
        except else
            MessageDlg('Ошибка записи в файл!', mtError, [mbOk], 0);
        end;
    end;

end;

procedure TfrmMain.FormDestroy(Sender: TObject);
begin
    TimeTable:=nil;
    Index:=Nil;
end;

end.

```

Тестовый пример

Ниже на рисунке 5 представлен пример работы программы при составлении расписания маршрутов.

Лабораторная работа №5

Программа позволяет просматривать и редактировать расписание автобусных маршрутов, отходящих от автовокзала города N, а также отфильтровывать маршруты, приходящие в указанный пункт раньше заданного времени.

Расписание маршрутов

№	Тип автобуса	Пункт назначения	Дата отправления	Дата прибытия
4	Газель	Москва	30.11.2011 13:35	30.11.2011 16:35
7	ПАЗ	Владивосток	28.11.2011 13:38	28.12.2011 13:38
5	Mercedes	Берлин	26.11.2011 12:25	30.11.2011 0:15
1	MAN	Москва	01.12.2011 13:00	09.12.2011 17:00
15	Icarus	Москва	21.12.2011 17:00	21.12.2011 20:00

Добавить строку Удалить строку

Фильтрация

Вывести маршруты до 13:39 30.10.2011 в пункт

Отфильтровать Сбросить фильтр Сохранить в файл

Рисунок 5 - Пример работы программы при составлении расписания

Ниже на рисунке 6 представлен пример работы программы, отображающей маршруты, прибывающие до 13:39 в Москву для указанного выше расписания.

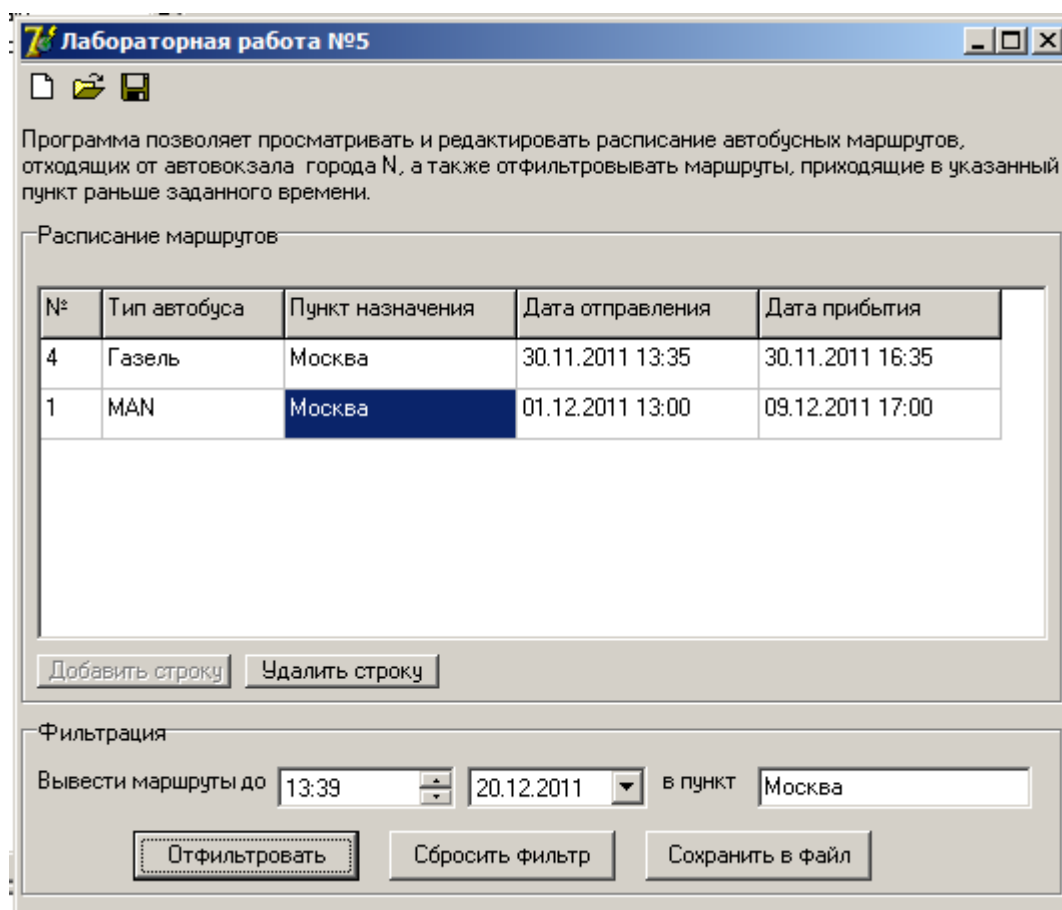


Рисунок 6 - Пример результатов поиска маршрутов программой

Вывод

В ходе выполнения данной лабораторной работы я научился использовать файлы в программах на Delphi. Delphi расширяет стандартный набор функций и процедур для работы с файлами, заимствованный из Паскаля, и кроме того, предоставляет удобные визуальные интерфейсы для организации помощи пользователю при работе с файлами. Файлы позволяют долговременно хранить большие объёмы данных и передавать их между совершенно различными программами. К сожалению, работа с файлами ведется медленнее, чем с оперативной памятью.