



Государственное бюджетное образовательное учреждение высшего образования
Московской области

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

Колледж космического машиностроения и технологии

КУРСОВОЙ ПРОЕКТ

По МДК.01.02 «Прикладное программирование»

Тема: «Разработка приложения “Ежедневник”»

Выполнил студент

Саполетова А.В.

Группа П1-17

_____ (Подпись)

_____ (Дата сдачи работы)

Принял преподаватель

Гусятинер Л.Б.

_____ (Оценка)

_____ (Подпись)

Королев 2020 г.

Оглавление

Введение.....	3
Глава 1. Теоретическая часть.....	4
1.1. Описание предметной области.....	4
1.2. Описание существующих разработок.....	7
Глава 2. Проектная часть.....	9
2.1. Диаграмма прецедентов.....	9
2.2. Выбор инструментов.....	9
2.3. Проектирование сценария.....	10
2.4. Диаграмма классов.....	11
2.5. Описание главного модуля.....	12
2.6. Описание спецификаций к модулям.....	13
2.7. Описание модулей.....	14
2.8. Описание тестовых наборов модулей.....	16
2.9. Описание применения средств отладки.....	19
2.10. Анализ оптимальности использования памяти и быстродействия.....	19
Глава 3. Эксплуатационная часть.....	23
3.1. Руководство оператора.....	23
3.1.1. Назначение программы.....	23
3.1.2. Условия выполнения программы.....	23
3.1.3. Выполнение программы.....	24
3.1.4. Сообщение оператору.....	27
Заключение.....	28
Список литературы и интернет-источников.....	29

Введение

Целью данного курсового проекта было создание ежедневника, в котором содержится телефонный справочник. Эта тема является актуальной в наше время, потому что все стремятся все упростить.

Ежедневник - это программа, которая помогает оптимизировать рабочее время пользователя и избавляет записывать важную информацию на листках бумаги, которая может легко потеряться.

В первой части будет рассмотрено описание предметной области и описание существующих разработок.

Во второй части будут рассмотрена проектная часть.

В третьей части будет рассмотрена эксплуатационная часть.

В заключительной части будет приведен To-do лист с планами по доработке программы.

Глава 1. Теоретическая часть

1.1. Описание предметной области

Ежедневник – теперь уже привычный атрибут современного делового человека. Это одновременно и записки о важных делах, а также календарь и памятка для важной информации.

История.

Первое упоминание о ежедневнике принято связывать с Италией 1650 г., откуда и взялось первое слово (лат. agenda: букв. «вещи, которые надо сделать»).

В современном понимании - деловой (обыкновенный) блокнот, предназначенный для ежедневного ведения записей о делах, планах, результатах и прочих записях, связанных с ежедневной деловой деятельностью человека или компании/организации. Наиболее популярными видами являются ежедневники в виде книги, в различном переплете. Одним из первых видов ежедневников, которыми пользовались Сартр, Пикассо, А. С. Куракин[уточнить], Ван Гог, Аполлинер, Матисс, Гертруда Стайн, Хемингуэй, Брюс Чатвин назван в честь прочного материала, который позволял «выживать» бумажному другу в различных условиях вместе со своим хозяином — Чертовой кожей (дословно англ. Moleskine — мех крота), молескиновыми робами могли в давние времена похвастаться ремесленники и золотоискатели.

Итальянские издатели очень гордятся тем, что являются исторической родиной ежедневников, и долгое время итальянский город Бергамо вместе с административным центром этой провинции — Миланом, считался не только европейским, а и мировым центром изготовления самых популярных марок ежедневников.

С конца XVII века и по сей день, наличие ежедневника свидетельствовало о значимости данной особы, а также подразумевало то, что в ежедневнике было что записывать. Это могли быть долговые обязательства, дела, налоговые ведомости и приказы распорядителям. Поэтому сам блокнот изготавливался со всеми регалиями и гербами, свойственными фамилии — по специальному заказу переплетных дел мастерами. Чаще всего переплетался в кожу, с большим количеством позолоты и содержал в своем первоначальном виде персональные данные владельца, где указано кому он принадлежит, куда и за какое вознаграждение следует вернуть ежедневник в случае его утери.

В наши дни ежедневник, и традиция его использования, получил широкое распространение, а сам принцип ведения списка дел и планирования времени перерос в целые направления по управлению временем (Тайм-менеджмент, Проектный - менеджмент).

В настоящее время ежедневники приобретают самые разнообразные формы исполнения, которые зависят от возможностей издателя. Встречаются даже ежедневники-раскраски. Да и значимость Итальянских производителей, как законодателей моды в данном вопросе, утратила силу. В СНГ уже существуют национальные производители, которые не уступают, а иногда и превосходят по качеству исполнения, аутентичным основателям традиции планирования.

Несмотря на все разнообразие - самым популярным остается ежедневник в виде книги в благородном твердом переплете, выполненным в мягкой коже, как правило снабженный дополнительной информацией для комфортной работы его владельца (коды телефонной связи, таблицы расстояний между местными областными центрами, конвертация единиц измерения, государственные и международные праздники, и т. п.), а также (только для датированных ежедневников) подготовленный для ведения

ежегодного, ежемесячного, еженедельного и ежедневного расписания. Что, впрочем, не мешает существованию недатированных ежедневников, даты и события в которых проставляет сам владелец.

Форматы.

Наиболее традиционные форматы ежедневников: А5, А4. В этом плане каждый определяет для себя наиболее удобные размеры:

Ежедневник А4 - 21*29см - настольный вариант, т.к. носить такой ежедневник на встречи и деловые переговоры некомфортно.

Ежедневник А5 - 15*21см - самый популярный формат, т.к. позволяет легко носить ежедневник в портфеле или деловой сумке.

Помимо размеров, производство ежедневников предполагает выбор их внутреннего наполнения. Бумагу для них чаще всего берут плотностью 70-80 гр/м², цветовая гамма чаще всего белая либо кремовая.

С развитием бизнес - среды в различных странах, ежедневник, из персонального продукта, продаваемого в розничных сетях, превращается в деловой сувенир, который принято дарить от лица компании лучшим клиентам или партнерам. Для этого рекламные агентства освоили различные методы брендинга и эксклюзивного исполнения ежедневников, с целью подчеркнуть имидж компании, которая совершает такой подарок, а также ценность подарка, и уважительное отношение к особе, которой такой подарок был преподнесен.

В наши дни ежедневники используются всё реже и вытесняются другими более универсальными средствами для планирования личного и рабочего времени, начиная от схожих по использованию ежедневников-планеров, до мобильных приложений, включающих в себя планировщики задач и поручений.

1.2. Описание существующих разработок

Представленная программа называется «Ежедневник». Эта программа относится к числу программ, расширяющих возможности компьютера.

Программа предназначена для планирования распорядка дня пользователя и выполняет функции планировщика задач с будильником.

Функциональные требования к программе:

- ежедневник открывается на текущей дате;
- просмотр записей без ограничений по дате;
- установление будильника на определенное время
- добавление и изменение заметок о распорядке дня;
- информация о свободном времени на следующую неделю;
- телефонный справочник;
- сохранение информации о всех внесенных записях во внешнем файле;
- программа должна обладать понятным интерфейсом и относительной простотой работы с ней;
- реализовать вызов файла помощи, содержащего руководство по пользованию программой.

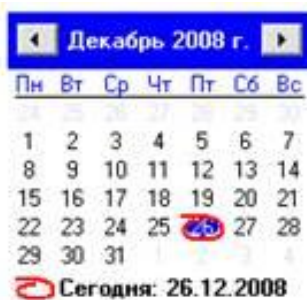


Рисунок 1. Электронный ежедневник

Необходимо написать программу в среде, реализующую функцию «Ежедневник». Данная программа должна полностью соответствовать требованиям технического задания. Необходимо, чтобы при запуске программы ежедневник открывал страницу с текущей датой и информацией

о свободном времени на следующую неделю. Кроме того, необходимо реализовать функцию выбора произвольной даты с формированием списка дел на выбранную дату. Кроме планировщика заданий в программе реализован справочник телефонных номеров. При разработке программы следует учесть, что информация о внесенных событиях и контактах должны храниться в файле.

В программе используется пять модулей:

- модуль описания классов;
- модуль основной части программы;
- модуль редактор дат;
- модуль для удаления записей из телефонного справочника;
- модуль для календаря.

Глава 2. Проектная часть

2.1. Диаграмма прецедентов

Была построена диаграмма для «Руководителя» и «Сотрудников». Руководитель может добавлять, удалять, редактировать и просматривать информацию. Сотрудники могут только просматривать информацию (Рисунок 2).

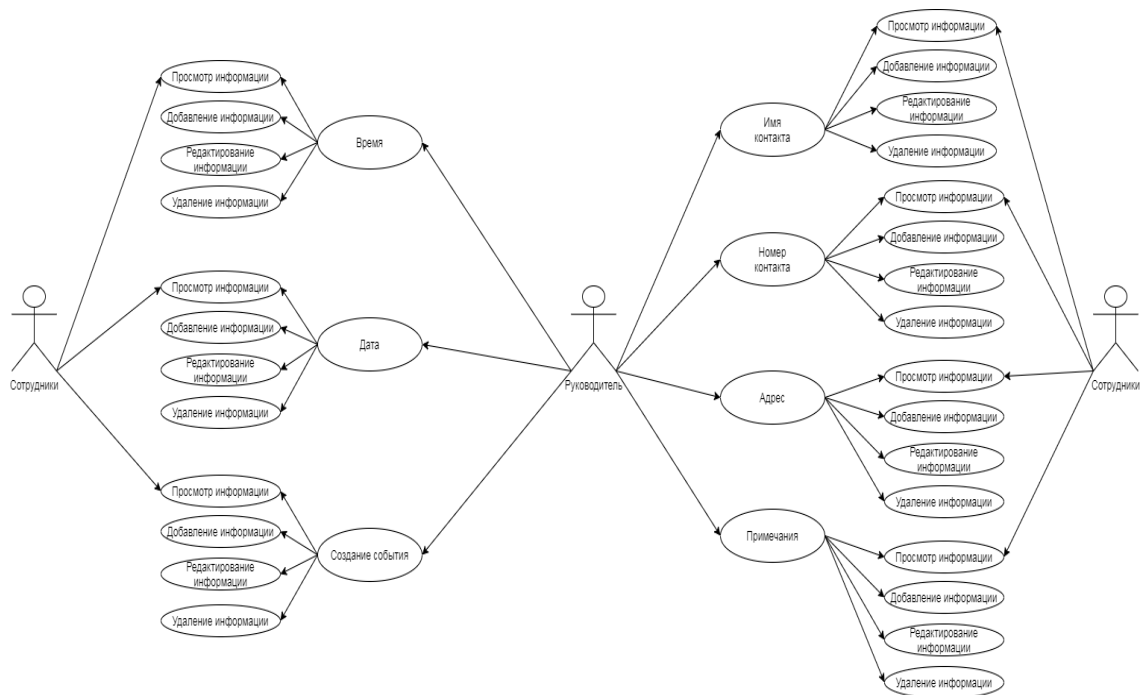


Рисунок 2. Диаграмма прецедентов

2.2. Выбор инструментов

При выборе инструментов были рассмотрены некоторые критерии. Важность критерия были выбраны из: низкая, ниже средней, средняя, ниже высокой, высокая.

Таблица 1.

Важность критериев выбора

Критерий	Участие в корпоративном проекте	Простота сопровождения	Наличие библиотек	Наличие документации на русском языке	Скорость разработки
Важность критерия	Ниже средней	Ниже высокой	Средняя	Ниже средней	Ниже высокой

Исходя из этих критериев, было проведено сравнение 2-х языков программирования от 0 до 10 баллов за критерий.

Таблица 2.

Выбор языка программирования

Критерий/Язык программирования	C++	Object Pascal
Участие в корпоративном проекте	6	7
Простота сопровождения	7	7
Наличие библиотек	6	5
Наличие документации на русском языке	8	9
Скорость разработки	6	7
Итого баллов	33	35

По результатам сравнения был выбран язык программирования Object Pascal.

2.3. Проектирование сценария

В данном разделе приведен сценарий использования программы руководителем (Рисунок 3).



Рисунок 3. Проектирование сценария

Пользователь после запуска программы может выполнить 1 действие: просмотреть информацию о программе и начать работу с таблицами. При выборе работы с таблицами пользователь попадает в новое окно, где может выбрать, с какой таблицей работать и какие действия совершать.

2.4. Диаграмма классов

В данном разделе представлены все классы, использующиеся в проекте, а также их отношения между собой (Рисунок 4).

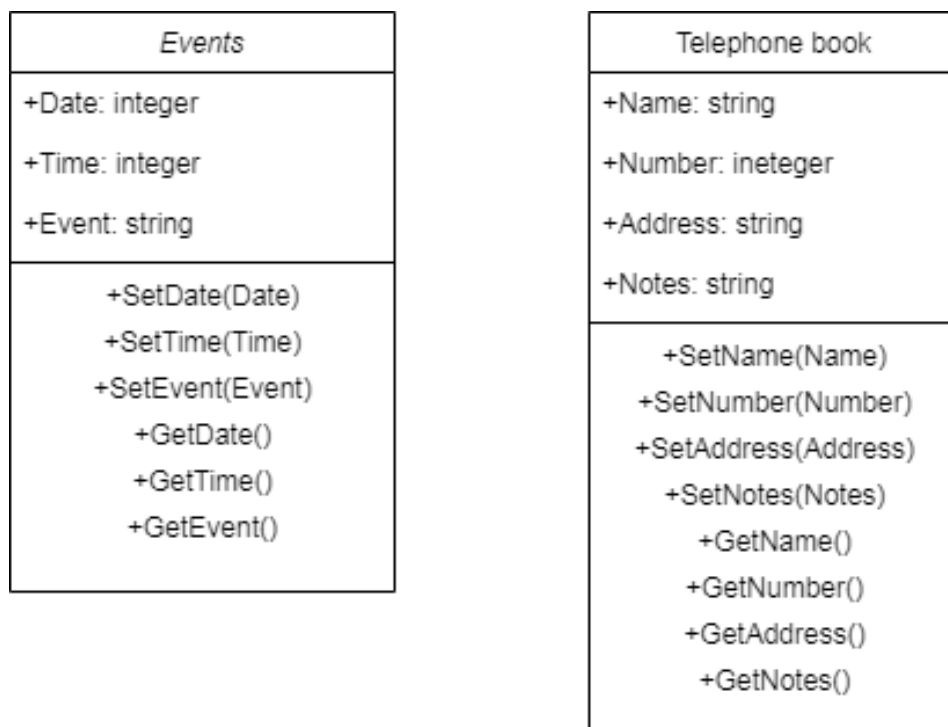


Рисунок 4. Диаграмма классов

2.5. Описание главного модуля

В главном модуле инициализируется основное приложение (Application) и создаются основные формы.

Листинг 1. Главная функция

```

program Planner;

{$mode objfpc}{$H+}

uses
  {$IFDEF UNIX}{$IFDEF UseCThreads}
  cthreads,
  {$ENDIF}{$ENDIF}
  Interfaces, // this includes the LCL widgetset
  Forms, MainForm, DBC, openform
  { you can add units after this };

{$R *.res}

begin
  RequireDerivedFormResource:=True;
  Application.Scaled:=True;
  Application.Initialize; //инициализация
  Application.CreateForm(TOpen_Form, Open_Form); //создание экземпляра
  формы Open_Form
  
```

```

Application.CreateForm(TMain_Form, Main_Form); //создание экземпляра
формы Main_Form
Application.CreateForm(TDataModule1, DataModule1);
//создание экземпляра модуля данных DataModule1
Application.Run;
end.

```

Блок-схема программы

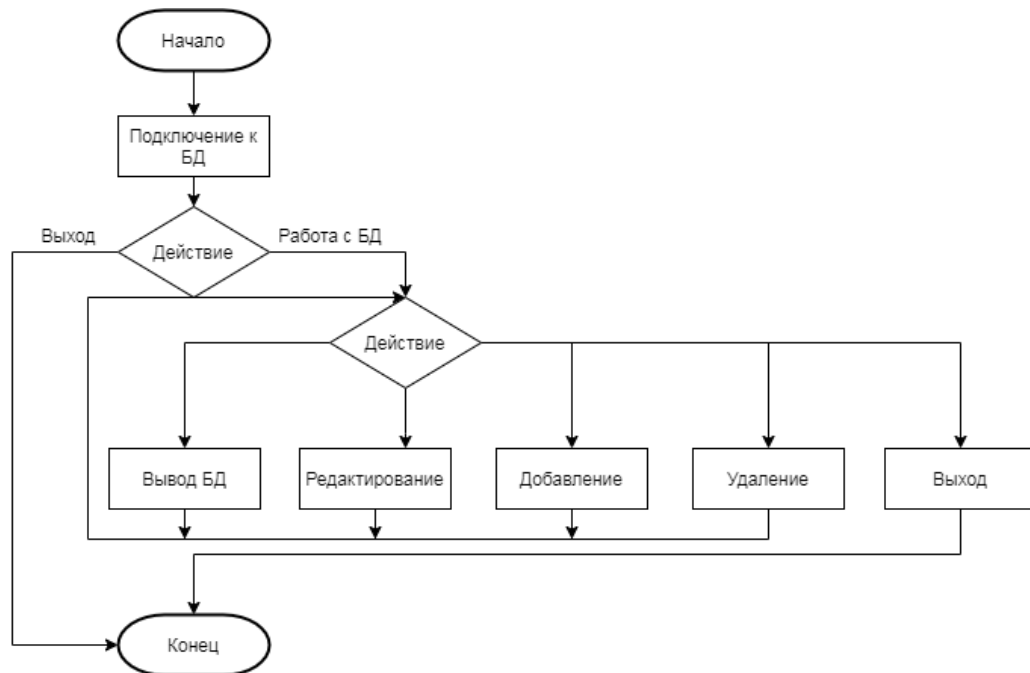


Рисунок 5. Блок-схема программы

2.6. Описание спецификаций к модулям

Были разработаны 3 модуля, не включая главный (Рисунок 6):

- MainForm – модуль, который позволяет работать с БД
- OpenForm – модуль главной формы
- DBC – модуль соединения с БД

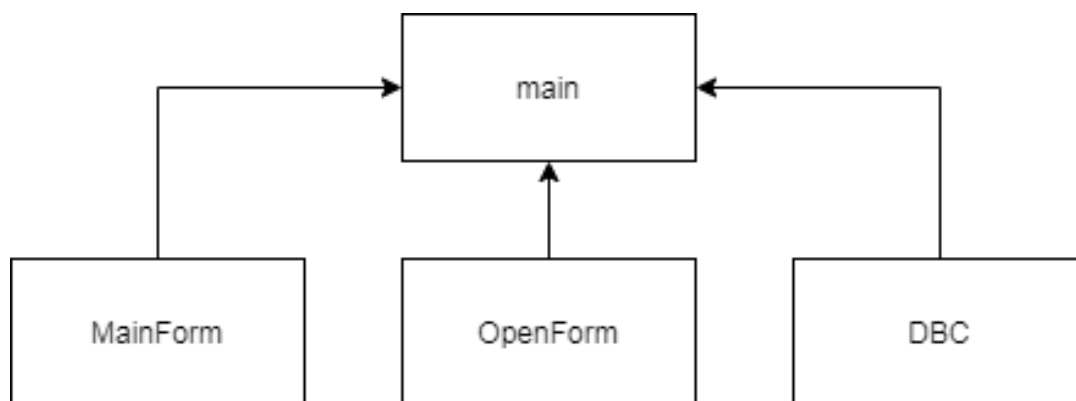


Рисунок 6. Структура подключения модулей

2.7. Описание модулей

Модуль 1. MainForm

Создается класс TMain_Form. Реализуются поля и методы классов.

Листинг 1. Создание класса TMain_Form

```

type

{ TMain_Form }

TMain_Form = class(TForm) //класс формы с таблицами БД
    DBGrid1: TDBGrid;      //компонент для отображения данных таблицы
    DBGrid2: TDBGrid;      //компонент для отображения данных таблицы
    DBNavigator1: TDBNavigator; //компонент для работы с таблицей
    DBNavigator2: TDBNavigator; //компонент для работы с таблицей
    Label1: TLabel;        //текст
    Label2: TLabel;        //текст
private
public
end;
```

Модуль 2. OpenForm

Создается класс TOpen_Form. Реализуются поля и методы классов.

Листинг 2. Создание класса TOpen_Form

```

type

{ TOpen_Form }

TOpen_Form = class(TForm) //класс главной формы
  Label1: TLabel;        //курсовой проект
  Label2: TLabel;        //тема: "Ежедневник"
  Label3: TLabel;        //выполнила: Саполетова Анастасия
  Label4: TLabel;        //студентка группы П1-17
  Label5: TLabel;        //17.06.2020
  Label6: TLabel;        //текст
  Open: TButton;         //кнопка открытия "Базы данных"
  procedure OpenClick(Sender: TObject); //процедура открытия
private

public

end;

```

Модуль 3. DBC

Создается класс TDataModule. Реализуются поля и методы классов. Обеспечивает связь с БД.

Листинг 3. Создание класса TDataModule.

```

type

{ TDataModule1 }

TDataModule1 = class(TDataModule) //класс модуля подключения
БД
  IBPHONEBOOKADDRESS: TIBStringField;
  IBPHONEBOOKID: TIBIntegerField;
  IBPHONEBOOKNAME: TIBStringField;
  IBPHONEBOOKNOTE: TIBStringField;
  IBPHONEBOOKPHONE: TIBStringField;
  PhoneBooks: TDataSource;
  EventsS: TDataSource;
  ibdb: TIBDatabase;
  IBEVENTS: TIBTable;
  IBEVENTSDATA: TDateField;
  IBEVENTSDBEVENT: TIBStringField;
  IBEVENTSID: TIBIntegerField;
  IBPHONEBOOK: TIBTable;
  ibtr: TIBTransaction;
  procedure DataModuleCreate(Sender: TObject);
private

```

```
public
```

```
end;
```

2.8. Описание тестовых наборов модулей

Тест 1. Ввод неправильной информации.

Если ввести в ID буквы, то туда не запишется никакая информация, кроме цифр (Рисунок 7).

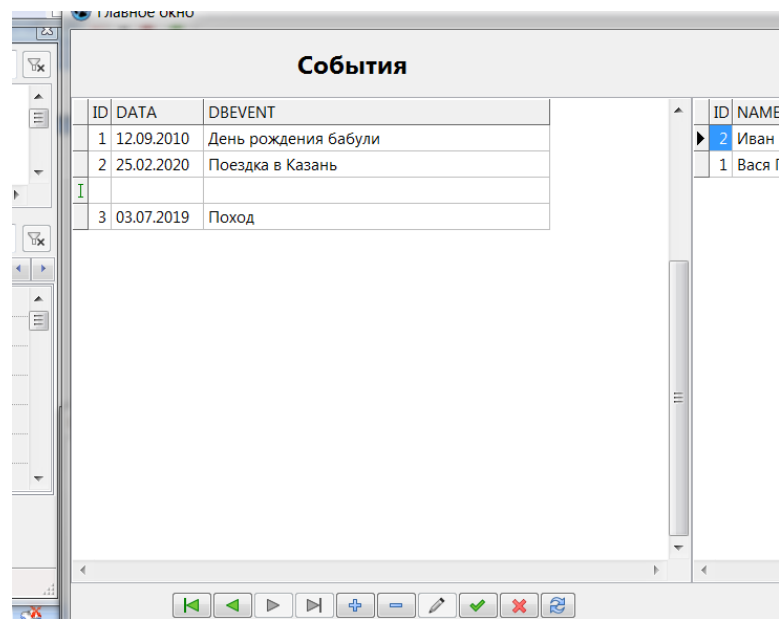


Рисунок 7. Тест 1

Тест 2. Отсутствие db.ini

При тестировании программы была обнаружена ошибка при указании неверного пути к БД в db.ini. Программа выводила пустые таблицы (Рисунок 8):

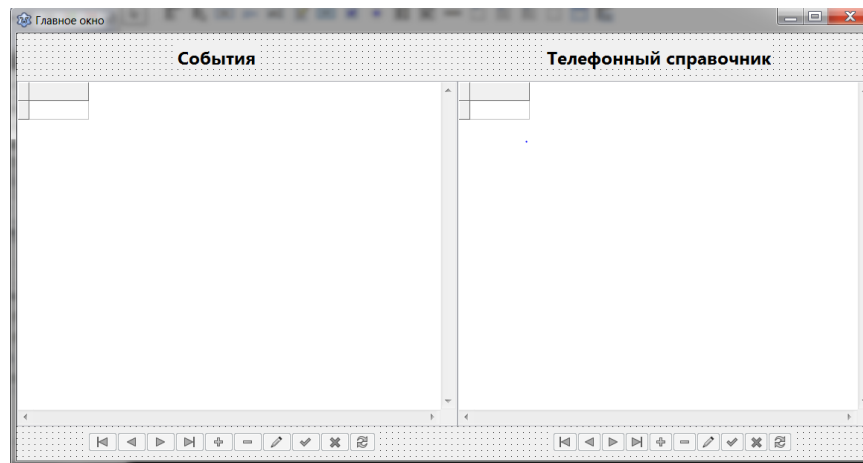


Рисунок 8. Тест 2. Ошибка

После обновления приложения эта ошибка отлавливается и выводится в главном модуле программы, кнопка перехода к работе с БД отключается (Рисунок 9):

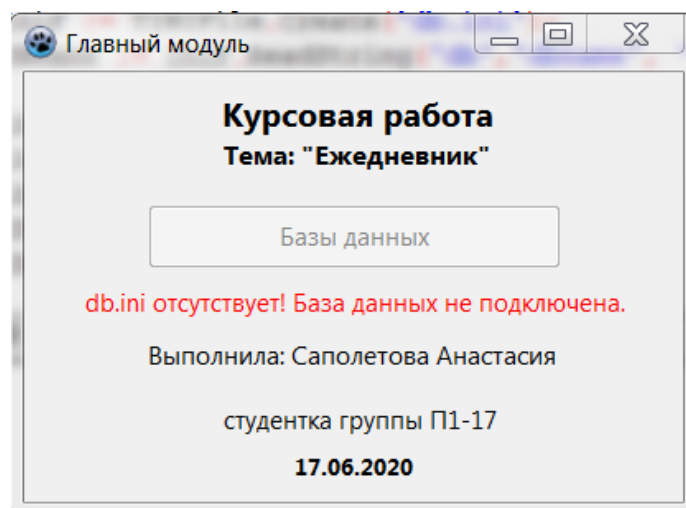


Рисунок 9. БД не подключена

Тест 3. Неправильный путь к БД

Если ввести неправильный путь к БД, то программа выдавала ошибку (Рисунок 10):

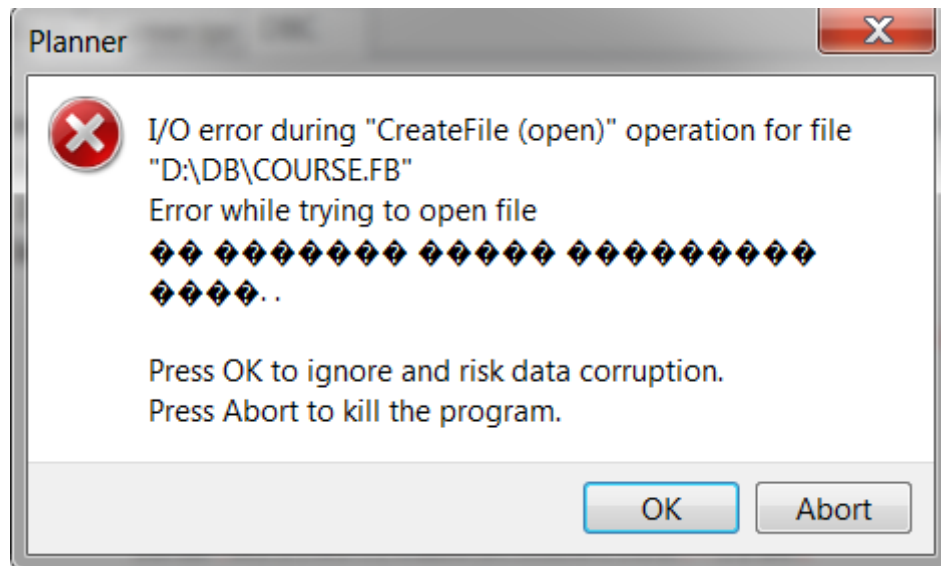


Рисунок 10. Ошибка

После обновления приложения ошибка отлавливается и выводится в главном модуле программы, кнопка перехода к работе с БД отключается (Рисунок 11):

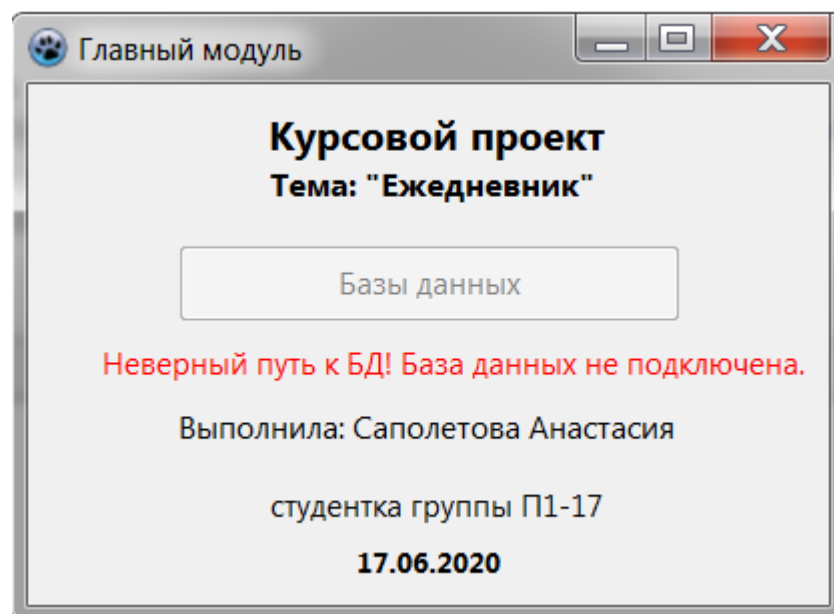


Рисунок 11. БД не подключена

2.9. Описание применения средств отладки

В ходе написания курсового проекта при попытке запустить программу были получены ошибки (Рисунок 12):

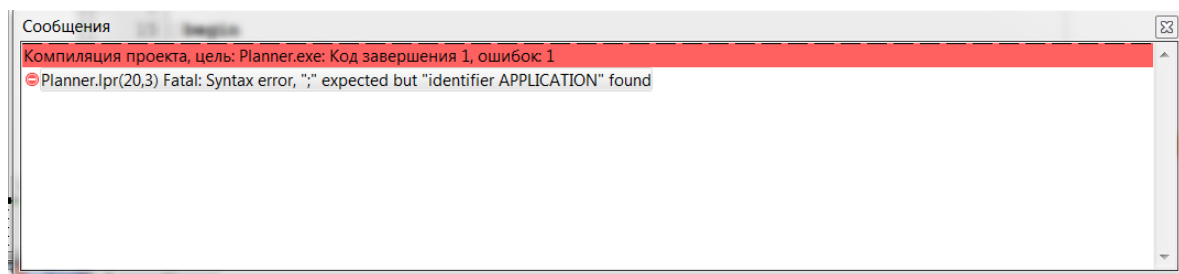


Рисунок 12. Сообщение об ошибках

При проверке кода были исправлены найденные ошибки, в результате при запуске программы ошибок не было (Рисунок 13):

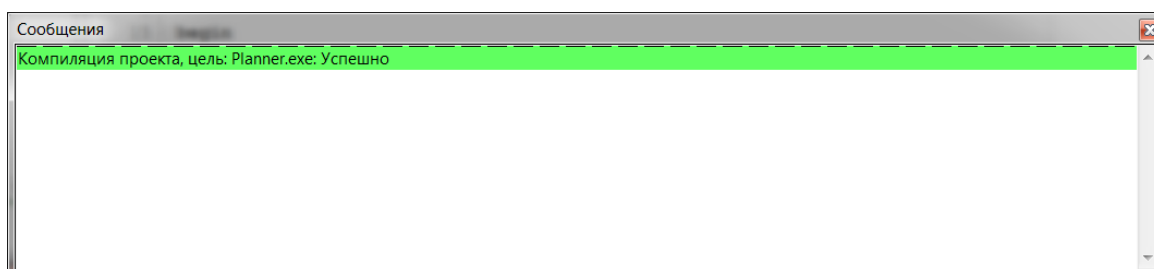


Рисунок 13. Сообщение об ошибках

2.10. Анализ оптимальности использования памяти и быстродействия

Удаление мертвого кода является одним из методов оптимизации. Удаление мертвого кода способно ускорить работу программы за счёт уменьшения количества исполняемых в ней операций и уменьшить размер программы. Для быстродействия программы было решено воспользоваться этим методом. Было удалено:

Удаление мертвого кода

Листинг 2.10.1. Мертвый код в модуле MainForm

```
Procedure TMainForm.FormCreate(Sender: TObject);
begin

end;
```

Оптимизация компиляции

Нам нужно оптимизировать приложение. Для этого нужно открыть проект и зайти в параметры проекта, далее компиляция и компоновка и затем отметить пункты, показанные на рисунке 14:

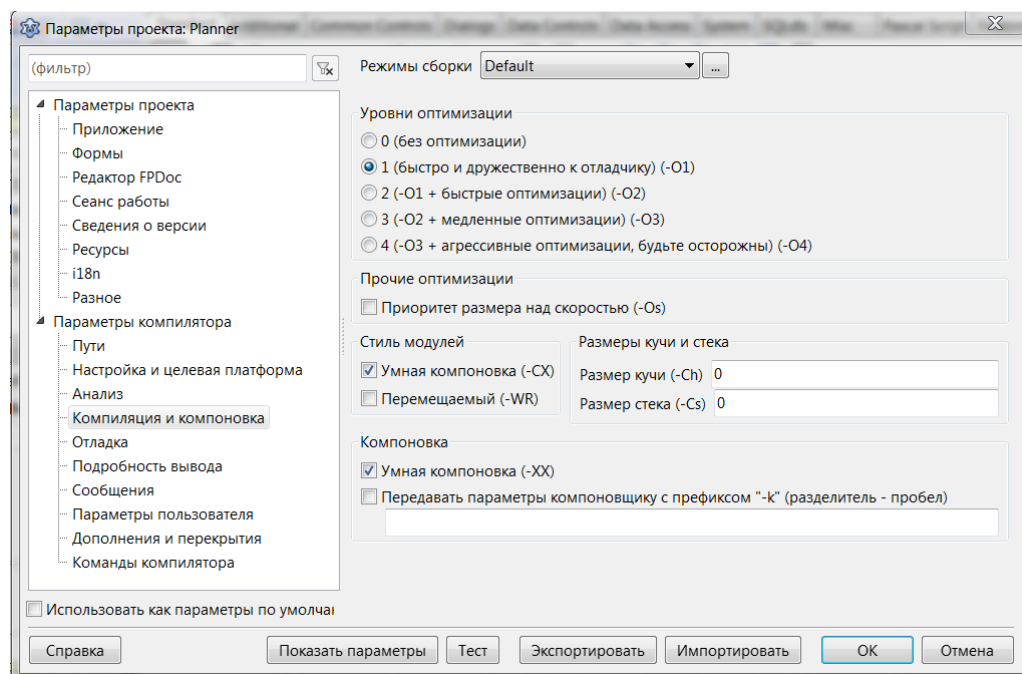


Рисунок 14. Компиляция и компоновка

Далее заходим в отладку. Отключаем отладочную информацию и включаем - Вырезать символы из исполняемого файла (Рисунок 15).

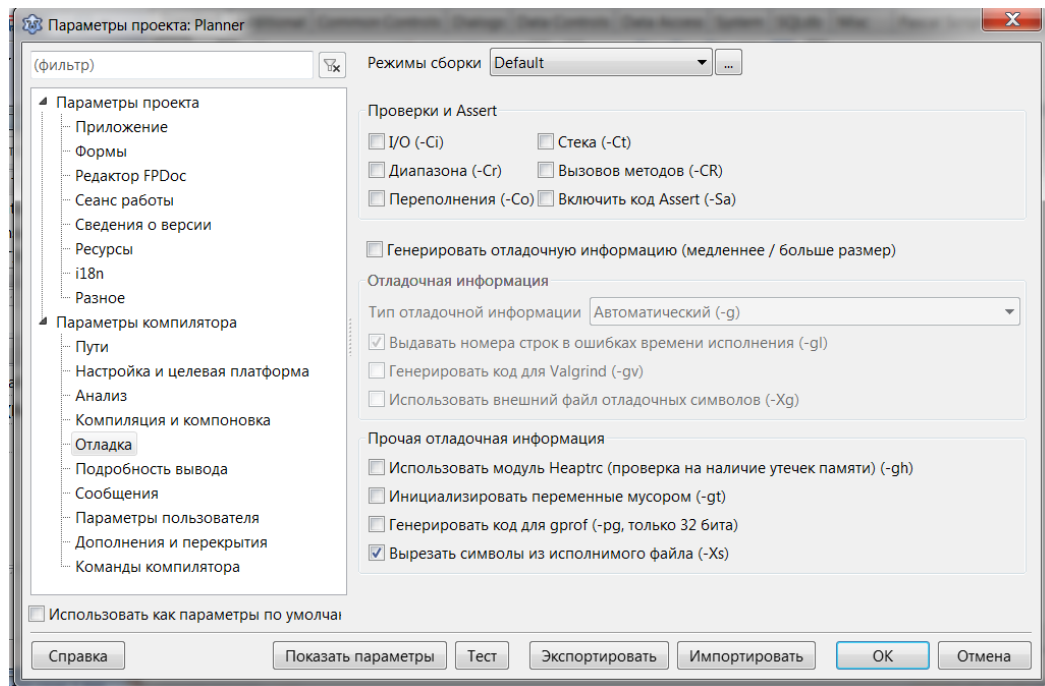


Рисунок 15. Отключение отладочной информации

Заходим в приложение и нажимаем очистить значок (Рисунок 16).

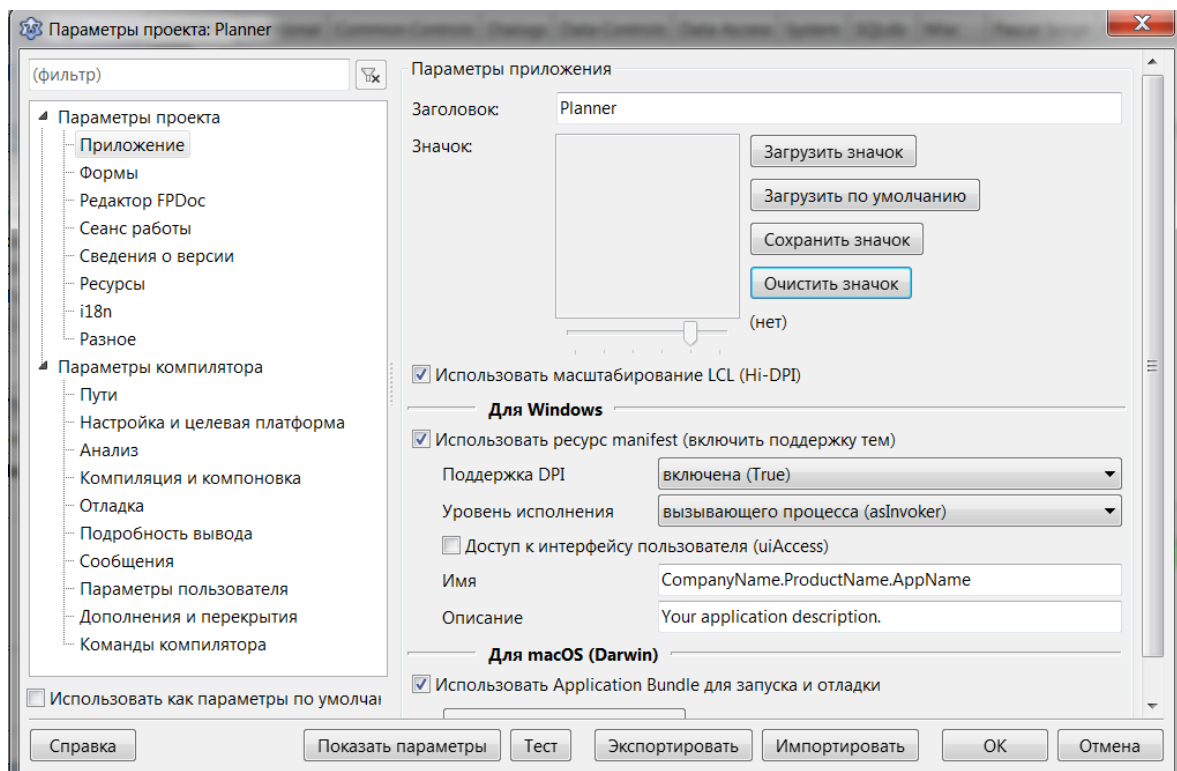


Рисунок 16. Очистение значка

Размер файла до оптимизации (Рисунок 17).


 Planner.exe	20.06.2020 10:03	Приложение	29 678 КБ
---	------------------	------------	-----------

Рисунок 17. До оптимизации

Размер файла после оптимизации (Рисунок 18).


 Planner.exe	20.06.2020 13:20	Приложение	4 300 КБ
---	------------------	------------	----------

Рисунок 18. После оптимизации

Глава 3. Эксплуатационная часть

3.1. Руководство оператора

3.1.1. Назначение программы

Разработанное приложение помогает оптимизировать рабочее время пользователя и избавляет записывать важную информацию на листках бумаги. Пользователю не обязательно уметь работать с базами данных. Программа работает с таблицами БД FireBird – записывает все изменения, внесенные пользователем в таблицу.

3.1.2. Условия выполнения программы

Минимальные требования:

Процессор: 2.0 GHz Dual Core Processor

Частота ЦПУ: Info

Оперативная память: 1 GB

ОС: Windows /7/8/10

Свободное место на диске: 20 MB

3.1.3. Выполнение программы

После запуска приложения открывается окно, где есть кнопка «База данных» и кто выполнил работу (Рисунок 19):

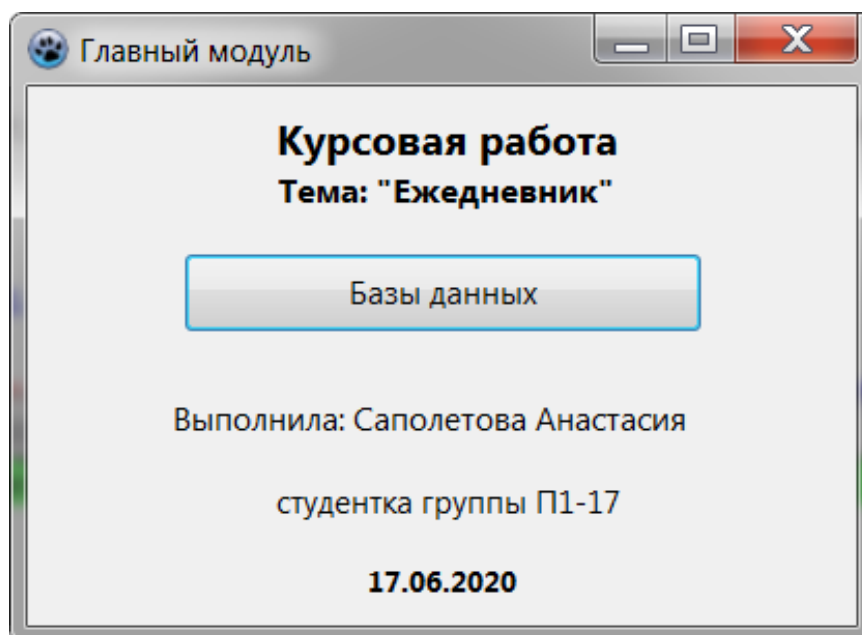


Рисунок 19. Главный модуль

При нажатии на кнопку «База данных» открывается главное окно, где есть две таблицы «События» и «Телефонный справочник» (Рисунок 20):

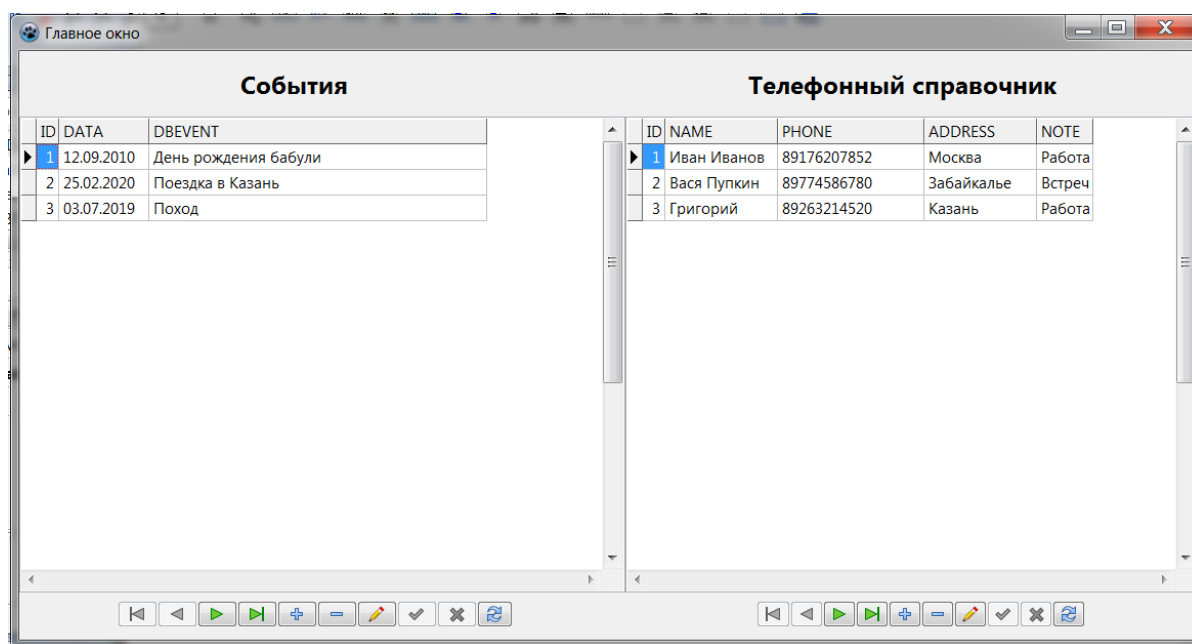


Рисунок 20. Главное окно

В этом окне есть две таблицы с данными из двух БД, две панели управления таблицами (Рисунок 21), в которых имеется: кнопки передвижения по таблице, добавление записи, удаление записи, редактирование записи, сохранение, отмена, обновление.



Рисунок 21. Панель управления таблицами

В начале работы с приложением нужно отредактировать в db.ini переменную dbname, указав путь до БД на своем ПК (Рисунок 22):

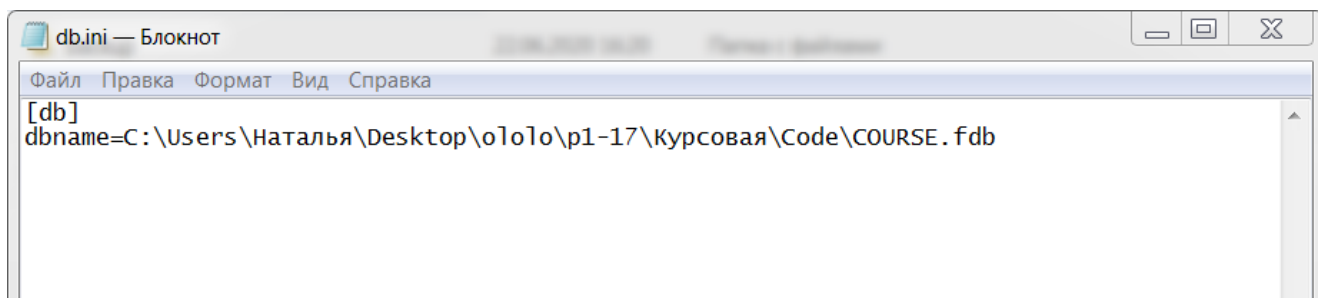


Рисунок 22. Путь до БД

После этого при указании верного пути до БД, зайдя в приложение и нажав на кнопку «База данных» можно увидеть таблицу. Отредактировав одно из полей (Рисунок 23) и зайдя в программу администрирования баз данных (IBExpert), можно увидеть изменения (Рисунок 24):

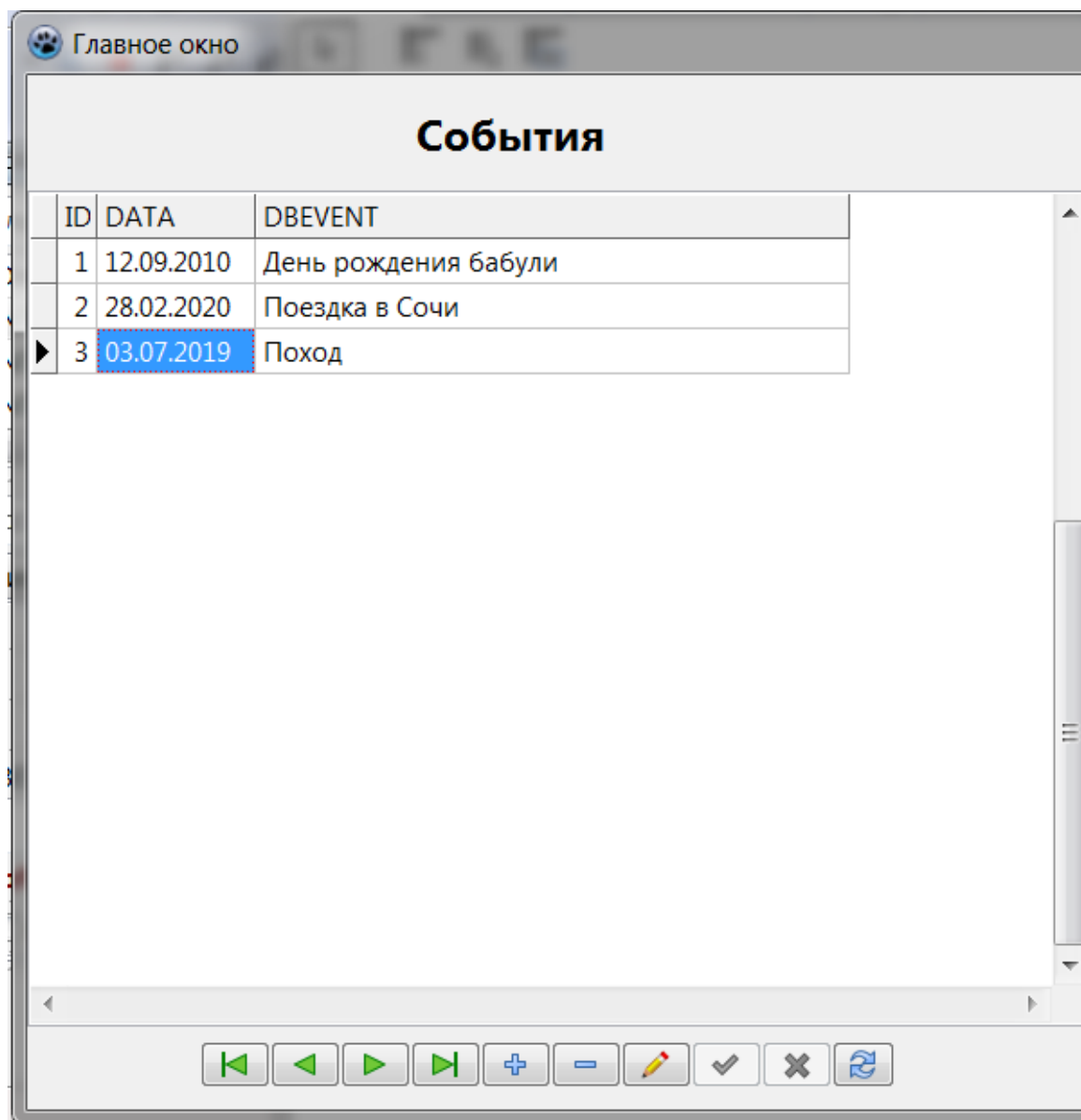


Рисунок 23. Редактирование полей

Table :: EVENTS :: COURSE (C:\Users\Наталья\Desktop\ololo\p1-17\Курсовая\Code\COURSE.fdb)

Fields Constraints Indices Dependencies Triggers Data Master/Detail View Description DDL Grants Logging Comparison To-do

Record: 1 Font size: 8 3 records fetched

ID	DATA	DBEVENT
1	12.09.2010	День рождения бабули
2	28.02.2020	Поездка в Сочи
3	03.07.2019	Поход

Рисунок 24. Измененная таблица в IBExpert

3.1.4. Сообщение оператору

Если попытаться добавить значение большего размера, чем задано, программа «Planner.exe» выдаст сообщение об ошибке, показанное на рис. 25

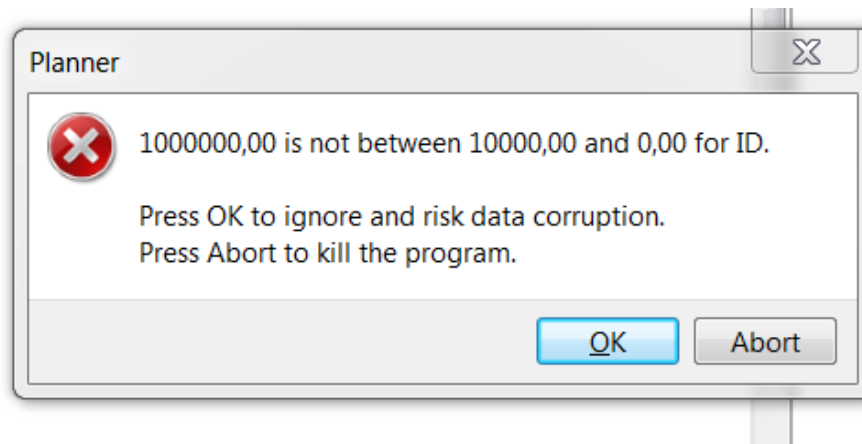


Рисунок 25. Сообщение об ошибке

Заключение

В результате выполнения курсового проекта была разработана программа «Ежедневник». Она помогает пользователю рационально планировать свой распорядок дня, а также хранить наиболее важные номера телефонов. Программа имеет удобный и простой интерфейс. Разработанная программа выполнена в соответствии с требованиями технического задания.

В дальнейшей работе планируется разработать календарь, который будет открывать событие на текущей дате, также будильники и сервисы напоминания о событиях, заданных пользователем.

Список литературы и интернет-источников

1. Ежедневник [1.1]

<https://ru.wikipedia.org/wiki/Ежедневник>

2. Ежедневник [1.2]

<https://www.skachatreferat.ru/referaty/Ежедневник/103481.html>

3. Кто придумал первый ежедневник? [1.2]

<http://www.kanzlerperm.ru/articles/76>

Приложение 1. Код главного модуля.

```

program Planner;

{$mode objfpc}{$H+}

uses
  {$IFDEF UNIX}{$IFDEF UseCThreads}
  cthreads,
  {$ENDIF}{$ENDIF}
  Interfaces, // this includes the LCL widgetset
  Forms, MainForm, DBC, openform
  { you can add units after this };

{$R *.res}

begin
  RequireDerivedFormResource:=True;
  Application.Scaled:=True;
  Application.Initialize; //инициализация
  Application.CreateForm(TOpen_Form, Open_Form);
  //создание экземпляра формы Open_Form
  Application.CreateForm(TMain_Form, Main_Form);
  //создание экземпляра формы Main_Form
  Application.CreateForm(TDataModule1, DataModule1);
  //создание экземпляра модуля данных DataModule1
  Application.Run;
end.

```

Приложение 2. MainForm.

```

unit MainForm;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls,
  ExtCtrls,
  DBGrids, DBCtrls;

type

  { TMain_Form }

  TMain_Form = class(TForm) //класс формы с таблицами БД
    DBGrid1: TDBGrid;      //компонент для отображения данных таблицы
    DBGrid2: TDBGrid;      //компонент для отображения данных таблицы
    DBNavigator1: TDBNavigator; //компонент для работы с таблицей
    DBNavigator2: TDBNavigator; //компонент для работы с таблицей
    Label1: TLabel;        //события
    Label2: TLabel;        //телефонный справочник
    procedure FormCreate(Sender: TObject); //процедура создания формы
  private

  public

  end;

var
  Main_Form: TMain_Form;

implementation

{$R *.lfm}

  { TMain_Form }

  procedure TMain_Form.FormCreate(Sender: TObject); //процедура
  begin

  end;

  end.

```

Приложение 3. OpenForm.

```

unit openform;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls;

type
  { TOpen_Form }

  TOpen_Form = class(TForm) //класс главной формы
    Label1: TLabel;        //курсовой проект
    Label2: TLabel;        //тема: "Ежедневник"
    Label3: TLabel;        //выполнила: Саполетова Анастасия
    Label4: TLabel;        //студентка группы П1-17
    Label5: TLabel;        //17.06.2020
    Label6: TLabel;        //текст
    Open: TButton;         //кнопка открытия "Базы данных"
    procedure OpenClick(Sender: TObject); //процедура открытия
  private
  public

  end;

var
  Open_Form: TOpen_Form;

implementation
uses
  MainForm;

{$R *.lfm}

{ TOpen_Form }

procedure TOpen_Form.OpenClick(Sender: TObject); //процедура события
begin
  Main_Form.Show; //открытие главной формы
end;

end.

```


Приложение 4. DBC.

```

unit DBC;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, db, IBDatabase, IBTable, IBCustomDataSet,
  INIFiles;

type

  { TDataModule1 }

  TDataModule1 = class(TDataModule)    //класс модуля подключения БД
    IBPHONEBOOKADDRESS: TIBStringField;
    IBPHONEBOOKID: TIBIntegerField;
    IBPHONEBOOKNAME: TIBStringField;
    IBPHONEBOOKNOTE: TIBStringField;
    IBPHONEBOOKPHONE: TIBStringField;
    PhoneBooks: TDataSource;
    EventsS: TDataSource;
    ibdb: TIBDatabase;
    IBEVENTS: TIBTable;
    IBEVENTSDATA: TDateField;
    IBEVENTSDBEVENT: TIBStringField;
    IBEVENTSID: TIBIntegerField;
    IBPHONEBOOK: TIBTable;
    ibtr: TIBTransaction;
    procedure DataModuleCreate(Sender: TObject);
  private
  public

  end;

var
  DataModule1: TDataModule1;
  IniF:TINIFile; //класс для работы с INI-файлами
  DBname: String;

implementation
uses
  openform;
{$R *.lfm}

{ TDataModule1 }

```

```

procedure TDataModule1.DataModuleCreate(Sender: TObject);
//открытие таблиц
begin
  IF(FileExists('db.ini'))then //проверка ini файла
  begin
    IniF := TINIFile.Create('db.ini');
    DBname := IniF.ReadString('db','dbname', '');
//считывание пути файла

    If (FileExists(DBname)) then //проверка на существование БД
    begin
      ibdb.DatabaseName:=DBname;
      ibdb.Connected:=True;
      ibdb.AllowStreamedConnected:=True;
      IBEVENTS.Active:=True;
      IBPHONEBOOK.Active:=True;

    end
    Else
    begin
      Open_Form.Label6.Caption:='Неверный путь к БД! База данных не
подключена.';
      Open_Form.Label6.Visible:=True;
      Open_Form.Open.Enabled:=False;
    end;

  end
  Else
  begin
    Open_Form.Label6.Caption:='db.ini отсутствует! База данных не
подключена.';
    Open_Form.Label6.Visible:=True;
    Open_Form.Open.Enabled:=False;
  end;
end;

end.

```