Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Конструирование программ и языки программирования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

АДРЕСНАЯ КНИГА

БГУИР КП 1-58 01 01 25 ПЗ

Студент: гр. 010902 Кузнецов А.Г.

Руководитель: ассистент кафедры ЭВМ   
Насуро Е. В.

Минск 2012

Учреждение образования

«Белорусский государственный университет информатики и

радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2012 г.

ЗАДАНИЕ

По курсовому проектированию

Студенту *Кузнецову Антону Германовичу*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Тема проекта «Адресная книга»

2. Срок сдачи студентом законченного проекта\_\_*31 декабря 2012г.\_\_*

3. Исходные данные проекта. Разработать программу адресной книги . Программа должна иметь удобный пользовательский интерфейс с необходимыми пунктами меню. Работа с информацией должна производиться в окнах. Информация должна храниться в различных файлах.. Работа с информацией должна производиться в окнах. Кроме адреса и телефона предусмотреть хранение дополнительной информации об абоненте Предусмотреть возможность поиска человека по имени, телефону и т. д.

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке): *Введение. 1.Обзор методов и алгоритмов постав-ленной задачи 2. Обоснование выбранных методов и алгоритмов 3. Описание программы для программиста 4. Описание алгоритмов решения задачи 5. Руководство пользователя. Заключение* \_

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

*1. Диаграмма классов*  ы

*2. Блок-схемы алгоритмов* ы

*3. Примеры работы программы* ы

6. Консультация по проекту (с обозначением разделов проекта) *Насуро Е.В.*

7. Дата выдачи задания *14 сентября 2012 г.* –

РУКОВОДИТЕЛЬ *Е.В.* *Насуро*

Задание принял к исполнению А.Г.Кузнецов

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc346060832)

1 Обзор МЕТОДОВ И АЛГОРИТМОВ ПОСТАВЛЕННОЙ ЗАДАЧИ 6

[1.1 Задание: 6](#_Toc346060833)

[1.2 Обзор методов и алгоритмов решения поставленной задачи: 6](#_Toc346060834)

[2 ОБОСНОВАНИЕ ВЫБРАННЫХ МЕТОДОВ 7](#_Toc346060835)

[2.1 Обоснование выбора языка программирования: 7](#_Toc346060836)

[2.2 Описание инструментов разработки программы: 7](#_Toc346060837)

[3 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПРОГРАММИСТА: 10](#_Toc346060838)

[3.1 Общее описание: 10](#_Toc346060839)

[3.2 Описание использованных структур данных: 10](#_Toc346060840)

[4 ОПИСАНИЕ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ 11](#_Toc346060841)

[5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 13](#_Toc346060842)

[5.1 Требования к программному и аппаратному обеспечению: 13](#_Toc346060843)

[5.2Памятка клиенту: 13](#_Toc346060844)

[ЗАКЛЮЧЕНИЕ 16](#_Toc346060845)

[СПИСОК ИСПОЛЬЗАВАНЫХ ИСТОЧНИКОВ 17](#_Toc346060846)

[ПРИЛОЖЕНИЕ А- листинг программы 19](#_Toc346060847)

[ПРИЛОЖЕНИЕ Б – скриншоты работы программы 23](#_Toc346060848)

# ВВЕДЕНИЕ

Сегодня на дворе 21 век - век информационных технологий, которые стремительно развиваются каждую секунду нашей жизни. Технологический, а также информационный прогресс, особенно за последние годы, сделал огромный шаг вперед. Современное человечество быстро превращается в информационное общество, а в особенности это происходит в быстроразвивающихся странах, которые делают большой упор именно на развитие техники и информационных технологий.

Почти к каждой семье имеется Персональный Компьютер потенциал которого огромен, использовать его возможно в различных сферах жизнедеятельности: учеба, развлечение, общение. Сфере общения и будет посвящена данная курсовая работа.

**1 ОБЗОР МЕТОДОВ И АЛГОРИТМОВ ПОСТАВЛЕННОЙ ЗАДАЧИ**

## 1.1 Задание

Разработать программу адресной книги. Программа должна иметь удобный пользовательский интерфейс с необходимыми пунктами меню. Работа с информацией должна производиться в окнах. Информация должна храниться в различных файлах. Кроме адреса и телефона предусмотреть хранение дополнительной информации об абоненте. Реализовать функции: добавление, сортировка записей. Предусмотреть возможность поиска человека по имени, телефону и т. д.

## 1.2 Обзор методов и алгоритмов решения поставленной задачи

В программе мы использовали данные структурного типа с использованием файлов. Главный класс «TZap» содержит поля, необходимые для функционирования программы. Глобальный указатель «Fz» содержит адрес последнего созданного файла.

# 2 ОБОСНОВАНИЕ ВЫБРАННЫХ МЕТОДОВ

## 2.1 Обоснование выбора языка программирования

Для решения поставленной задачи была выбрана интегральная среда разработки (ИСР) C++ Builder 6 поскольку:

- в ней доступны широкие возможности языка C++

- используется широким кругом программистов-профессионалов, разрабатывающих в ОС Windows

- она имеет все необходимые средства и инструменты для разработки приложений любой сложности

- удобна в использовании, гибка в настройке

- создает очень эффективный код

- в ней имеется возможность использования готовых библиотек классов

- позволяет импортировать исходный код и проекты для ИСР других ОС

- имеет встроенный механизм обмена данными по сети при помощи socket-ов

Так как ИСР выбрана C++ Builder 6, соответственно языком программирования выбран C++, поскольку:

- «родной» язык для создания приложений в среде C++ Builder 6

- подлинная объектная ориентированность (всякая языковая сущность претендует на то, чтобы быть объектом)

- компонентно-ориентированное программирование

- безопасный код

- унифицированная система типизации

- поддержка событийно-ориентированного программирования

При разработке данного проекта было принято решение создать клиент для чата, который был реализован в C++ Builder 6.

## 2.2 Описание инструментов разработки программы

C++ Builder 6 — программный продукт, инструмент быстрой разработки приложений (RAD), интегрированная среда программирования (IDE), система, используемая программистами для разработки программного обеспечения на языке программирования C++.

Изначально разрабатывался компанией Borland Software, а затем её подразделением CodeGear, ныне принадлежащим компании Embarcadero Technologies.

C++ Builder объединяет в себе комплекс объектных библиотек (STL, VCL, CLX, MFC и др.), компилятор, отладчик, редактор кода и многие другие компоненты. Цикл разработки аналогичен Delphi. Большинство компонентов, разработанных в Delphi, можно использовать и в C++ Builder без модификации, но обратное утверждение не верно.

C++ Builder содержит инструменты, которые при помощи drag-and-drop действительно делают разработку визуальной, упрощает программирование благодаря встроенному WYSIWYG — редактору интерфейса и пр.

Альтернатива асинхронному подходу — блокирующие соединения, при которых ваше приложение ожидает завершения операций чтения или записи перед выполнением следующей строки кода. В этом случае следует согласованно писать код для обеих сторон, поскольку иначе не будут генерироваться события. При работе с блокирующим соединением необходимо создать на сервере поток и, как правило, работать с потоком также и на стороне клиента.

**• Единая модель программирования для всех приложений** - При создании приложений ранее разработчикам приходилось использовать различные приемы программирования, которые существенным образом зависели от типа приложения — технологии разработки клиентского программного обеспечения, общедоступных веб-приложений, программного обеспечения для мобильных устройств и бизнес-логики промежуточного уровня значительно различались между собой. Среда разработки Builder 6 решает данную проблему, предоставляя в распоряжение разработчиков единую модель создания приложений всех категорий. Эта интегрированная модель обладает привычным и одновременно интуитивно понятным интерфейсом, позволяя разработчикам использовать свои навыки и знания для эффективного создания широкого спектра приложений.

**• Всесторонняя поддержка жизненного цикла разработки** - Среда Builder 6 обеспечивает поддержку всего жизненного цикла разработки: начиная с планирования и проектирования через разработку и тестирование и вплоть до развертывания и последующего управления. Builder 6 предоставляет всестороннюю адаптируемую среду для создания всех приложений, жизненно необходимых для успешной работы современных компаний.

**• Описание использованных технологий. Язык программирования C++.**

C++ (произносится «си плюс плюс») — компилируемый статически типизированный язык программирования общего назначения. Поддерживая разные парадигмы программирования, сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником— языком C, — наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования. Название «C++» происходит от языка C, в котором унарный оператор ++ обозначает инкремент переменной.

Являясь одним из самых популярных языков программирования, C++ широко используется для разработки программного обеспечения. Область его применения включает операционные системы, прикладные программы, драйверы устройств, приложения для встраиваемых систем, высокопроизводительные серверы, а также развлекательные приложения, например, видеоигры. Существует несколько реализаций языка, как бесплатных, так и коммерческих. Их производят GNU Project, Microsoft, Intel и Embarcadero (Borland). C++ оказал огромное влияние на другие языки программирования, в первую очередь на C# и Java.

При создании C++ стремились сохранить совместимость с языком C. Большинство программ на C будут исправно работать и с компилятором C++. C++ имеет синтаксис, основанный на синтаксисе C.

# 3 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПРОГРАММИСТА

# 

## 3.1 Общее описание

## 

При создании приложения были созданы классы-описатели. Для их использования было решено использовать статическую структуру ввиду её достаточности в рамках данного задания.

При сохранении базы используются файловые потоки с текстовым режимом для удобства отладки. Сначала все данные переносятся в поля клсса а затем заносятся в файл. Таким образом, данные не склеиваются при загрузке из файла. Форма для сохранения и загрузки выбрана стандартная для системы Windows.

В формах используется компонент StringGrid, в который выводится вся информация в удобной для пользователя форме.

## 3.2 Описание использованных структур данных

Таблица 3.1 – Поля класса TZap

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | **Описание** | |
| name | Char[30] | Имя | |
| lname | Char[30] | Фамилия | |
| mnade | Char[30] | Отчество | |
| num1 | Char[20] | Номер телефона1 | |
| num2 | Char[20] | Номер телефона2 | |
| num3 | Char[20] | Номер телефона3 | |
| adr | Char[30] | Адрес | |
| date | Char[10] | | Дата Рождения |

Таблица 3.2 – Глобальные переменные

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| size | int | Переменная размера |
| File\_Zap | AnsiString | Имя файла |
| filters | AnsiString | Название Фильтра |
| date | AnsiString | Переменная Даты |

# 4 ОПИСАНИЕ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ

**4.1 Рассмотрим алгоритм добавления записи**

Шаг 1: Создаём временные переменные.

Шаг 2: Запускаем диалог открытия файла. Если мы получили путь, то записываем его в переменную «File\_Zap». Если нет-то завершаем работу функции.

Шаг 3: Выполняем алгоритм проверки заполнения основных полей

Шаг 4: Построчно копируем информацию из заполненных полей в соответствующие поля класса TZap

Шаг 5: Получаем переменную data используя свойство Date.FormatString компонента DateTimePicker в формате dd.mm.yyyy

Шаг 6: Записываем переменную data в соответствующее поле класса TZap.

Шаг 7: Записываем ***1***блоков по ***size*** байт каждый из области памяти, на которую указывает ***Zap*** в файл ***Fz***

Шаг 8: Очищаем используемые поля.

Шаг 9: Закрываем Доступ к файлу используя стандартную функцию fclose.

**4.2 Рассмотрим алгоритм Чтения и вывода файла на экран**

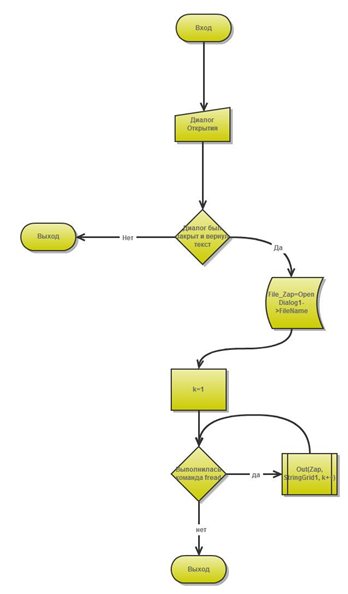


Рисунок 4.1 – Блок-схема алгоритма сохранения

# 5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

**5.1 Требования к программному и аппаратному обеспечению**

Операционная система Windows ME/2K/XP/Seven/Ubuntu

Видео SVGA (800x600, 65536 цветов)

Процессор x86-совместимый 600 MHz

Оперативная память 128 MB

Место на жестком диске не менее 10 MB

Устройства ввода Клавиатура, Мышь

Внимание! Для корректной работы программы необходимо, чтобы в вашей системе были установлено средства .NET Framework 3.5 или более поздняя версия. Если данный компонент не установлен у Вас в системе, то его можно найти на официальном сайте Microsoft.

## 5.2 Памятка клиенту

Программа состоит из одного \*.exe файла самой программы и базы данных, которые создаются по мере её использования. В случае первого использования программы требуется создать новый файл, для этого нажав кнопку «Добавить», это вызовет окно «Добавление».

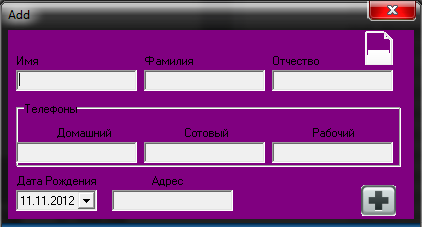


Рисунок 5.1 – Добавление Записи

В нём вводится Имя, Фамилия, Отчество, Различные телефоны такие как : домашний, рабочий, сотовый, дата рождения и адрес. После заполнения полей требуется нажать кнопку «Добавить». Это вернёт вас на главный экран. Файл будет сохранен ,пользователю останется выбрать его из списка для заполнения таблицы данными

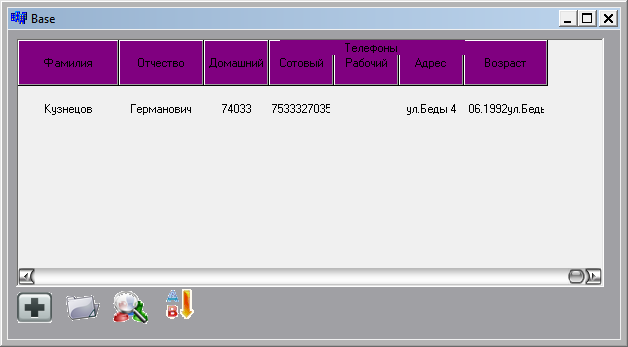


Рисунок 5.2 – Заполненная таблица

Для поиска следует нажать на кнопку «Поиск» которая перекинет вас в окно поиска

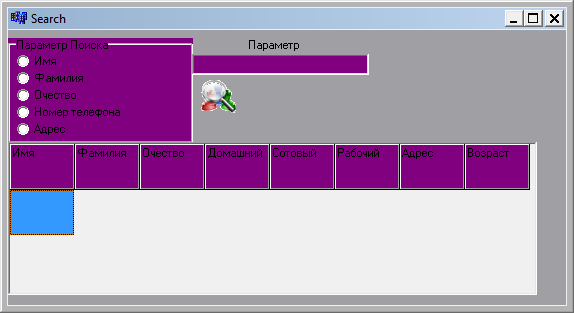


Рисунок 5.3 – Окно поиска

Для поиска требуется выбрать необходимый параметр поиска заполнить поле «Параметр» по которому будет происходить поиск и нажать на кнопку «поиск».

ЗАКЛЮЧЕНИЕ

В результате курсовой работы создана программа «Адресная Книга». Программа позволяет добавлять и сортировать данные, осушествлять поиск данных по различным параметрам. Данные загружаются вручную пользователем, сохранение производится с помощью соответствующего пункта меню.

Программный продукт выполнен в среде BorlandC++ Builder 6.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Дейтел, Х.М. Как программировать на С++ / Х.М. Дейтел, П.Д. Дейтел; пер. с англ. – М. : Бином, 2007. – 1152 с.

[2]  Страуструп, Б. Язык программирования С++ / Б. Страуструп; специальное издание. Пер. с англ. – СПб. : BHV, 2008. – 1098 с.

[3]  Скляров, В.А. Язык С++ и объектно-ориентированное программирование: справ. пособие /  В.А. Скляров. – Мн. : Вышэйшая школа, 1997. – 478 с.

[4]  Элджер, Дж. C++: библиотека программиста / Дж. Элджер. – СПб. : Питер, 2001. –  с.

[5] Шилд, Г. Программирование на Borland C++ для профессионалов / Г. Шилд.  – Мн. : ООО «Попури», 1998. – 800 с.

[6] Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, А. Джекобсон; пер. с англ. – СПб. : ДМК, 2004. – 429 с.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обозначение | | | | | | Наименование | | | | | Дополнительные  сведения | | |
|  | | | | | | Текстовые документы | | | | |  | | |
|  | | | | | |  | | | | |  | | |
| БГУИР КП 1-58 01 01 25 ПЗ | | | | | | Пояснительная записка | | | | | 17 с. | | |
|  | | | | | | Листинг программы | | | | | 12 с. | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | | Графические документы | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | | Скриншоты программы | | | | | Формат A4 | | |
|  | | | | | | Диаграмма классов | | | | | Формат A4 | | |
|  | | | | | | Блок-схемы алгоритмов | | | | | Формат A4 | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  |  | |  |  | |  |  | |  |  |  |  |  |
|  |  | |  |  | |  |  | |  |  |  |  |  |
|  |  | |  |  | |  |  | |  |  |  |  |  |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  | | | | | |  | | | | |  | | |
|  |  |  | | |  |  | | БГУИР КП 1-58 01 01 25 ПЗ | | | | | |
|  |  |  | | |  |  | |
|  |  |  | | |  |  | |
| Изм. | Л. | № Докум. | | | Подп. | Дата | | Адресная книга Ведомость курсового проекта |  | | | Лист | Листов |
| Разраб. | | Кузнецов А.Г. | | |  |  | | Т |  |  | 18 | 36 |
| Пров. | | Насуро Е.В. | | |  |  | |  | | | | |
|  | |  | | |  |  | |
|  | |  | | |  |  | |
|  | |  | | |  |  | |

# ПРИЛОЖЕНИЕ А- листинг программы

**Unit1.cpp**

**//---------------------------------------------------------------------------**

#include <vcl.h>

#pragma hdrstop

#include "Unit2.cpp"

#include "Unit1.h"

#include "Unit2.h"

#include "Unit3.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

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

: TForm(Owner)

{

}

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

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

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

{

StringGrid1->Cells[0][0]="Имя"; StringGrid1->Cells[1][0]="Фамилия";

StringGrid1->Cells[2][0]="Отчество"; StringGrid1->Cells[3][0]="Домашний";

StringGrid1->Cells[4][0]="Сотовый"; StringGrid1->Cells[5][0]="Рабочий";

StringGrid1->Cells[6][0]="Адрес"; StringGrid1->Cells[7][0]="Возраст";

Form1->Caption="Base";

}

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

void \_\_fastcall TForm1::StringGrid1DrawCell(TObject \*Sender, int ACol,

int ARow, TRect &Rect, TGridDrawState State)

{

TStringGrid\* StringGrid =

static\_cast<TStringGrid\*>(Sender);

StringGrid->Canvas->FillRect(Rect);

AnsiString text(

StringGrid->Cells[ACol][ARow]);

RECT RText = static\_cast<RECT>(Rect);

InflateRect(&RText, -3, -3);

DrawText(StringGrid->Canvas->Handle,

text.c\_str(), text.Length(), &RText,

DT\_CENTER | DT\_SINGLELINE |

DT\_VCENTER );

}

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

void \_\_fastcall TForm1::SpeedButton1Click(TObject \*Sender)

{

Form2->Visible=true;

}

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

void \_\_fastcall TForm1::SpeedButton2Click(TObject \*Sender)

{

OpenDialog1->Filter=filters;

OpenDialog1->Title="Выбирете Группу";

if (OpenDialog1->Execute()){

File\_Zap=OpenDialog1->FileName;

Form3->Edit2->Text=File\_Zap;

if ((Fz=fopen(File\_Zap.c\_str(),"rb"))==NULL) {

ShowMessage("Ошибка создания ФАЙЛА!");

return;

}

int k = 1;

while(1){

if(!fread(&Zap,size,1,Fz)) break;

Out(Zap, StringGrid1, k++);

}

}

fclose(Fz);

}

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

void \_\_fastcall TForm1::SpeedButton3Click(TObject \*Sender)

{

Form3->Visible=true;

}

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

void \_\_fastcall TForm1::SpeedButton4Click(TObject \*Sender)

{

TStringList \*SL = new TStringList;

AnsiString S;

for (int i=1; i < StringGrid1->RowCount-1; i++){

S="";

for (int j=0; j < StringGrid1->ColCount; j++) S += StringGrid1->Cells[j][i]+" ";

SL->Add(S);

}

SL->Sort();

for (int i=1; i < StringGrid1->RowCount-1; i++){

StringGrid1->Rows[i]->DelimitedText = SL->Strings[i-1];

}

delete SL;

}

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

**Unit2.cpp**

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

#include <vcl.h>

#pragma hdrstop

#include <io.h>

#include "Unit2.h"

#include "Unit1.h"

#include <stdio.h>

//#include "Class.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm2 \*Form2;

//TStringGrid \*StringGrid1;

class TZap{

public:

char name[30], lname[30], mname[30],date[10], adr[30],num1[20],num2[20],num3[20];

} Zap;

int size = sizeof(TZap);

FILE \*Fz;

AnsiString File\_Zap;

void Out(TZap, TStringGrid\*, int);

AnsiString filters = "\*.bzd";

AnsiString date;

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

void Out(TZap z, TStringGrid \*StringGrid1, int k)

{

StringGrid1->RowCount++;

StringGrid1->Cells[0][k]=(AnsiString)Zap.name;

StringGrid1->Cells[1][k]=(AnsiString)Zap.lname;

StringGrid1->Cells[2][k]=(AnsiString)Zap.mname;

StringGrid1->Cells[3][k]=(AnsiString)Zap.num1;

StringGrid1->Cells[4][k]=(AnsiString)Zap.num2;

StringGrid1->Cells[5][k]=(AnsiString)Zap.num3;

StringGrid1->Cells[7][k]=(AnsiString)Zap.date;

StringGrid1->Cells[6][k]=(AnsiString)Zap.adr;

}

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

\_\_fastcall TForm2::TForm2(TComponent \*Owner)

: TForm(Owner)

{

}

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

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

void \_\_fastcall TForm2::FormCreate(TObject \*Sender)

{

Edit1->Clear(); Edit2->Clear();Edit3->Clear();

Edit4->Clear(); Edit5->Clear();Edit6->Clear();

Edit7->Clear();

Label1->Caption="Имя"; Label2->Caption="Фамилия";

Label3->Caption="Отчество"; Label4->Caption="Домашний";

Label5->Caption="Сотовый"; Label6->Caption="Рабочий";

Label7->Caption="Дата Рождения"; Label8->Caption="Адрес";

GroupBox1->Caption="Телефоны";

Form2->Caption="Add";

}

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

void \_\_fastcall TForm2::SpeedButton1Click(TObject \*Sender)

{

if ((Fz = fopen(File\_Zap.c\_str(),"ab"))==NULL)

ShowMessage("Ан нет ВЫ НЕ СОЗДАЛИ ФАЙЛ!");

else{

if(Edit1->Text==""||Edit2->Text==""||Edit3->Text=="")

ShowMessage("Ан нет ВЫ ТАКИ НЕ ЗАПОЛНИЛИ ПОЛЕЙ!");

else{

strcpy( Zap.name, Edit1->Text.c\_str());

strcpy( Zap.num1, Edit4->Text.c\_str());

strcpy( Zap.num2, Edit5->Text.c\_str());

strcpy( Zap.num3, Edit6->Text.c\_str());

strcpy( Zap.lname,Edit2->Text.c\_str());

strcpy( Zap.mname,Edit3->Text.c\_str());

strcpy( Zap.adr,Edit7->Text.c\_str());

date = DateTimePicker1->Date.FormatString("dd.mm.yyyy");

strcpy( Zap.date, date.c\_str());

fwrite(&Zap, size, 1, Fz);

Edit1->Text=""; Edit2->Text=""; Edit3->Text="";

Edit4->Text=""; Edit5->Text=""; Edit6->Text=""; Edit7->Text="";

fclose(Fz); }

}

Form2->Visible = false;

}

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

void \_\_fastcall TForm2::SpeedButton2Click(TObject \*Sender)

{

OpenDialog1->Filter=filters;

OpenDialog1->Title="Выбирете Группу";

if (OpenDialog1->Execute()){

File\_Zap=OpenDialog1->FileName;

if ((Fz=fopen(File\_Zap.c\_str(),"ab"))==NULL) {

ShowMessage("Ошибка создания ФАЙЛА!");

return;

}

}

fclose(Fz);

}

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

**Unit3.cpp**

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

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

#include "Unit1.cpp"

#include "Unit3.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm3 \*Form3;

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

\_\_fastcall TForm3::TForm3(TComponent\* Owner)

: TForm(Owner)

{

}

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

void \_\_fastcall TForm3::SpeedButton1Click(TObject \*Sender)

{

File\_Zap=Edit2->Text;

if ((Fz=fopen(File\_Zap.c\_str(),"rb"))==NULL)

ShowMessage("Ошибка создания ФАЙЛА!");

else {

if(Edit1->Text=="") ShowMessage("Введите искомый параметр");

else {

char q[30];

strcpy( q, Edit1->Text.c\_str());

TZap st, \*mas\_Z;

Fz = fopen(File\_Zap.c\_str(),"rb");

int D\_f = fileno(Fz); // Находим дескриптор файла

int len = filelength(D\_f); // Находим размер файла

int i, j, kol, l;

kol = len/size; //Количество записей в файле

mas\_Z = new TZap[kol]; // Считываем записи из файла в динамический массив

for (i=0; i < kol; i++)

fread((mas\_Z+i), size, 1, Fz);

fclose(Fz);

int k=1;

switch(RadioGroup1->ItemIndex){

case 0:

for (i=0; i < kol; i++)

if (!strcmp(q,mas\_Z[i].name))

{ Out( mas\_Z[i], Form3->StringGrid1,k++);

k++;

Edit1->Text="";

}

if (k==1)

ShowMessage("Not Found");

k=1;

delete []mas\_Z;

break;

case 1:

for (i=0; i < kol; i++)

if (!strcmp(q,mas\_Z[i].lname))

{ Out( mas\_Z[i], Form3->StringGrid1,k++);

k++;

Edit1->Text="";

}

if (k==1)

ShowMessage("Not Found");

k=1;

delete []mas\_Z;

break;

case 2:

for (i=0; i < kol; i++)

if (!strcmp(q,mas\_Z[i].mname))

{ Out( mas\_Z[i], Form3->StringGrid1,k++);

k++;

Edit1->Text="";

}

if (k==1)

ShowMessage("Not Found");

k=1;

delete []mas\_Z;

break;

case 3:

for (i=0; i < kol; i++)

if (!strcmp(q,mas\_Z[i].num1))

{ Out( mas\_Z[i], Form3->StringGrid1,k++);

k++;

Edit1->Text="";

}

if (k==1)

ShowMessage("Not Found");

k=1;

delete []mas\_Z;

break;

case 4:

int k=0;

for (i=0; i < kol; i++)

if (!strcmp(q,mas\_Z[i].adr))

{ Out( mas\_Z[i], Form3->StringGrid1,k++);

k++;

Edit1->Text="";

}

if (k==1)

ShowMessage("Not Found");

k=1;

delete []mas\_Z;

break;

}

}

}

}

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

void \_\_fastcall TForm3::FormCreate(TObject \*Sender)

{

StringGrid1->Cells[0][0]="Имя"; StringGrid1->Cells[1][0]="Фамилия";

StringGrid1->Cells[2][0]="Очество"; StringGrid1->Cells[3][0]="Домашний";

StringGrid1->Cells[4][0]="Сотовый"; StringGrid1->Cells[5][0]="Рабочий";

StringGrid1->Cells[6][0]="Адрес"; StringGrid1->Cells[7][0]="Возраст";

Edit1->Clear();

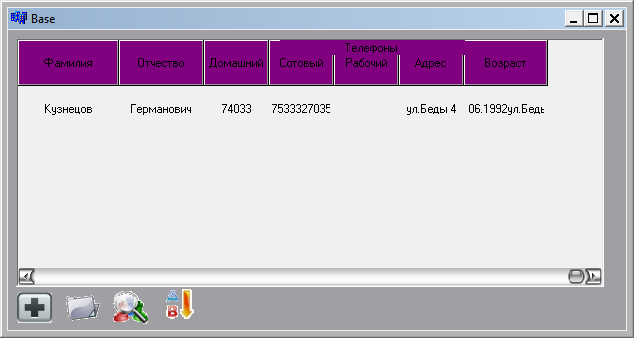
Label1->Caption="Параметр";

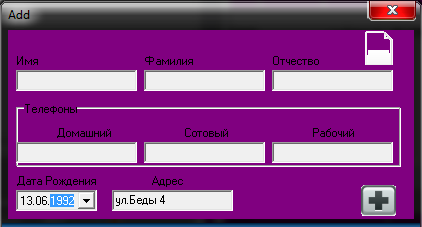
Form3->Caption="Search";

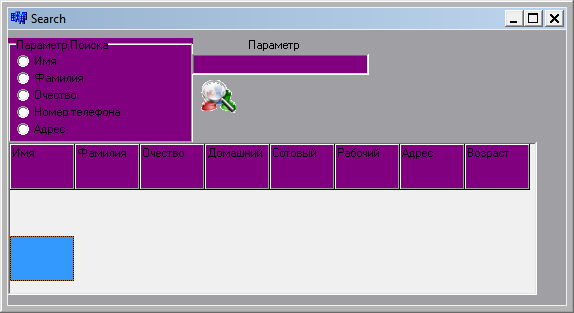
}

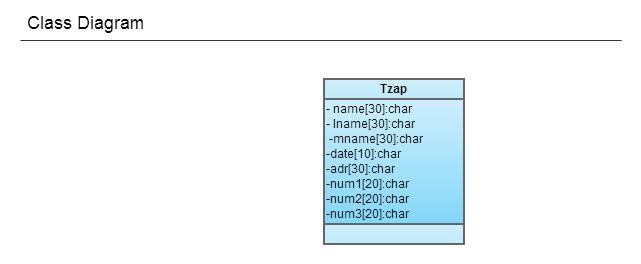
//---------------------------------------------------------------------------

# ПРИЛОЖЕНИЕ Б – скриншоты работы программы







******