|  |  |  |  |
| --- | --- | --- | --- |
| Министерство образования и науки Российской Федерации | | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  «Кузбасский государственный технический университет им.Т.Ф.Горбачева» | | | |
| Кафедра прикладных информационных технологий | | | |
| Дисциплина: | Информатика и программирование | | |
| Направление подготовки: | 09.03.03 Прикладная информатика | | |
|  | | | |
| Пояснительная записка к курсовой работе  на тему: «Разработка программного модуля для обработки текстовой информации» | | | |
|  | | Выполнили  студ. группы | Минлигареев М.А.  Гулёв Д.М.  ПИб-152 |
| Принял: | Ещин Е.К. |
|  | |
| Кемерово 2016 | | | |

**Содержание**

[1. Введение 3](#_Toc452311923)

[2. Постановка задачи 4](#_Toc452311924)

[3. Описание программы 5](#_Toc452311925)

[3.1 Общие сведения 5](#_Toc452311926)

[3.2 Перспективность использования 5](#_Toc452311927)

[3.3 Входные данные 6](#_Toc452311928)

[3.4 Выходные данные 6](#_Toc452311929)

[3.5 Описание логической структуры 6](#_Toc452311930)

[3.6 Вызов программы 42](#_Toc452311931)

[Заключение 44](#_Toc452311932)

[Список литературы 45](#_Toc452311933)

# Введение

Текстовая информация – это данные, представленные в виде структурированного текста в определённой последовательности и с определённым форматированием.

Под обработкой текстовой информации принято понимать воздействие на неё со стороны пользователя персонального компьютера с использованием специализированного программного обеспечения с целью предать ей конечный вид.

Задачей курсовой работы является разработка программного модуля для обработки текстовой информации.

Целью научно-исследовательских работ является получение новых и дополнение уже имеющихся знаний по данной теме для грамотного выполнения курсовой работы по дисциплине «Информатика и программирование».

# Постановка задачи

В результате анализа предметной области: [[[1]](#endnote-1)] (отсутствует шифрование текста), [[[2]](#endnote-2)] (отсутствует исправление текста), [[[3]](#endnote-3)] (имеет несоответствующий дизайн) было установлено, что необходимо разработать программный модуль (программное обеспечение), используемый для обработки текстовой информации, отвечающий приведённым ниже требованиям.

Наибольший интерес при реализации представляет шифрование и дешифрование текста, а также исправление текста, в случае его ввода с английской раскладки клавиатуры на русскую и наоборот. Помимо этого, одним из наиболее решающих факторов, послуживших для основания на разработку программного модуля, явился элемент свободно-распространяемости программного обеспечения. Это означает, что большинство программных модулей для обработки текстовой информации, имеющихся на настоящее время, также имеют потребность в их попериодной коммерциализации, что для данного проекта крайне нежелательно.

# Описание программы

# 3.1 Общие сведения

Программный модуль носит название «APPostate» и имеет расширение «.exe».

Для работоспособности программы необходимо следующее программное обеспечение:

* операционная система Windows.

Программа написана на языке программирования Object Pascal в среде программирования Delphi XE7.

Для стабильного функционирования программного модуля рекомендуется использовать электронно-вычислительные машины со следующей минимальной конфигурацией:

* центральный процессор с частотой не ниже 500 МГц;
* не менее 256 МБ оперативной памяти;
* жесткий диск с объёмом не менее 16 ГБ;
* монитор с разрешением не менее 1024x768 точек;
* клавиатура;
* мышь.

# 3.2 Перспективность использования

Программный модуль представляет собой простой в использовании текстовый редактор, который способен вытеснить стандартное приложение «Блокнот» из пользовательского обращения с ПК ввиду того, что он представляет его дополненную полезными функциями реализацию.

# 3.3 Входные данные

Входными данными для работы программы являются:

* вводимая с клавиатуры текстовая информация;
* загружаемый пользователем файл документа.

# 3.4 Выходные данные

Выходными данными работы программы являются:

* преобразованная текстовая информация;
* сохраняемый пользователем файл документа;
* характеристика документа.

# Описание логической структуры

Программный модуль для обработки текстовой информации состоит из 25 программных модулей. Большая их часть однотипна, поэтому ниже приведены программные модули представляющие наибольший интерес для пользователя

Алгоритм работы основного модуля выбора действия после запуска программы представлен на рисунке 1.

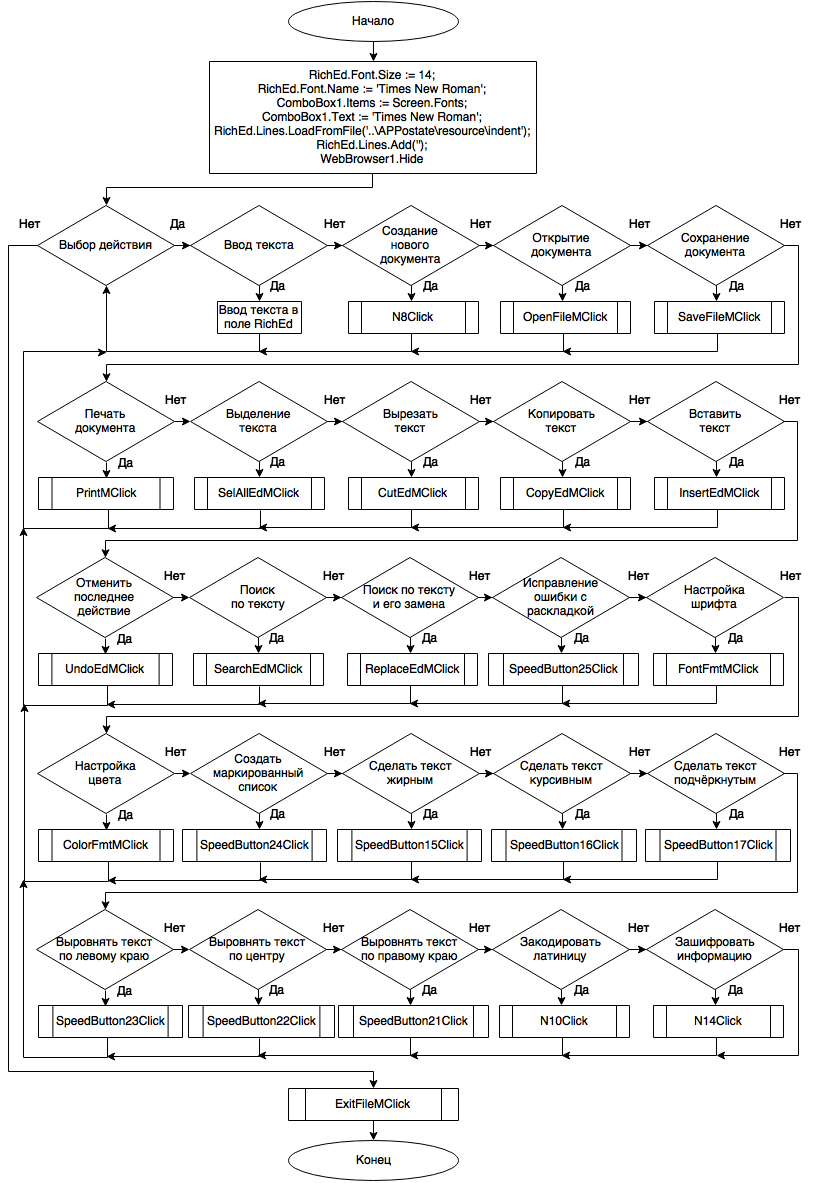


Рисунок 1 - Алгоритм выбора действия после запуска программы

Алгоритм работы модуля открытия документа представлен на рисунке 2.

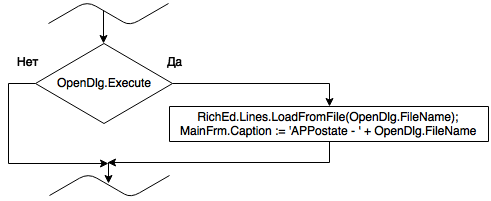


Рисунок 2 - Алгоритм открытия документа

Алгоритм работы модуля сохранения документа представлен на рисунке 3.

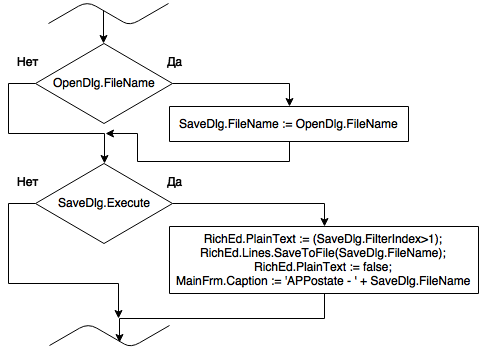


Рисунок 3 - Алгоритм открытия документа

Алгоритм работы модуля шифрования информации представлен на рисунке 4.

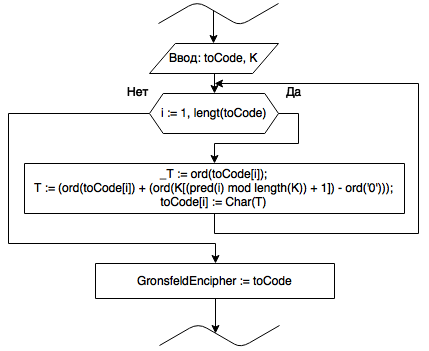


Рисунок 4 - Алгоритм шифрования информации

Алгоритм работы модуля дешифрования информации представлен на рисунке 5.

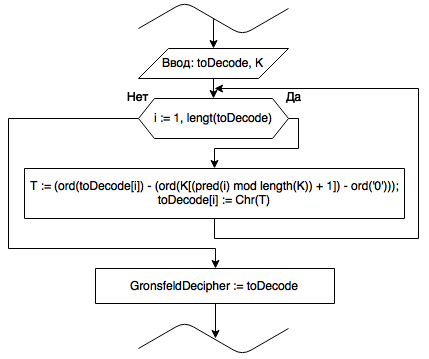


Рисунок 5 - Алгоритм дешифрования информации

Алгоритм работы модуля поиска и замены текста представлен на рисунке 6.

Алгоритм работы модуля исправления текста в случае его ввода с другой раскладки представлен на рисунке 7.

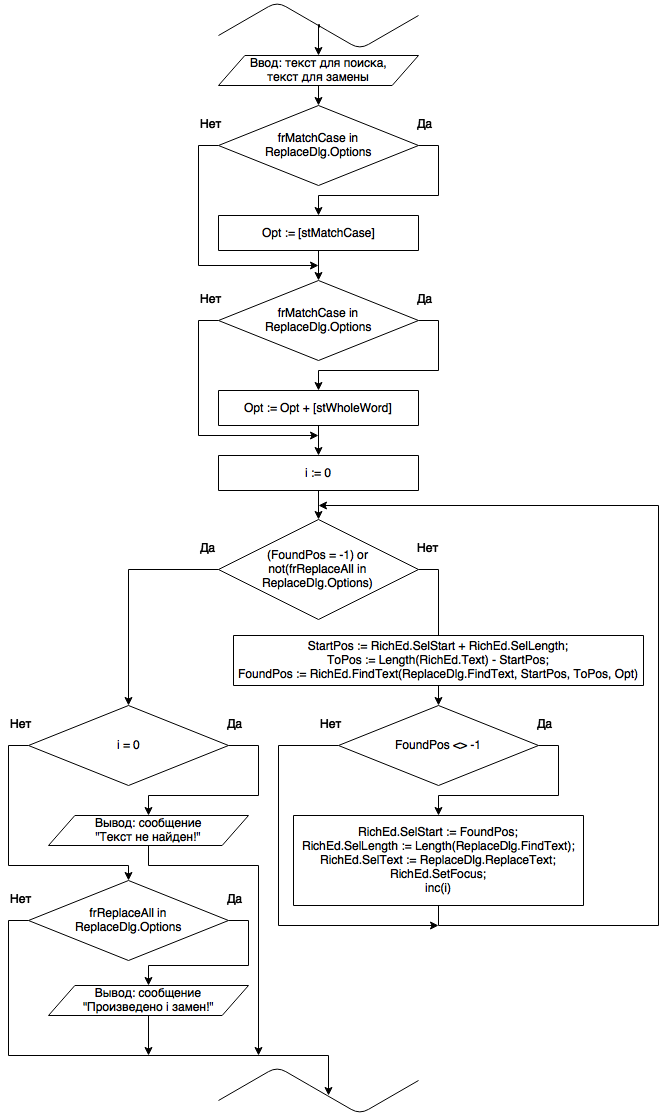


Рисунок 6 - Алгоритм поиска и замены текста

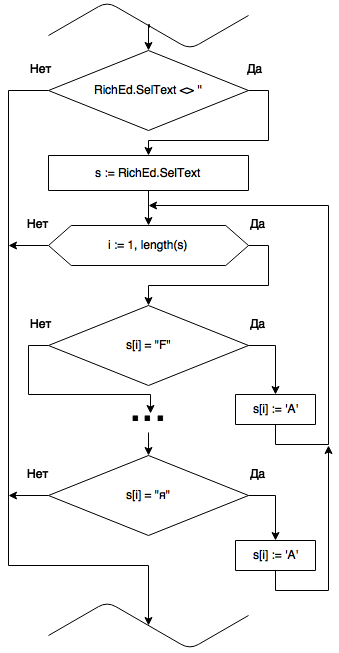


Рисунок 7 - Алгоритм исправления текста в случае его ввода с другой раскладки

unit MainForm;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Menus, StdCtrls, ComCtrls, ToolWin, Clipbrd, Vcl.ExtCtrls,

Vcl.Buttons, Vcl.Samples.Spin, Vcl.OleCtrls, SHDocVw, Vcl.FileCtrl, Vcl.Grids,

Vcl.Outline, IdCoder, IdCoder3to4, IdCoder00E, IdCoderXXE, IdBaseComponent,

IdCoderMIME;

type

TMainFrm = class(TForm)

MainMenu: TMainMenu;

RichEd: TRichEdit;

OpenDlg: TOpenDialog;

SaveDlg: TSaveDialog;

PrintDlg: TPrintDialog;

PrintSetupDlg: TPrinterSetupDialog;

FontDlg: TFontDialog;

ColorDlg: TColorDialog;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

OpenFileM: TMenuItem;

SaveFileM: TMenuItem;

PrintM: TMenuItem;

PrintSetupM: TMenuItem;

ExitFileM: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

UndoEdM: TMenuItem;

N6: TMenuItem;

CutEdM: TMenuItem;

CopyEdM: TMenuItem;

InsertEdM: TMenuItem;

SelAllEdM: TMenuItem;

FontFmtM: TMenuItem;

ColorFmtM: TMenuItem;

FindDlg: TFindDialog;

ReplaceDlg: TReplaceDialog;

N7: TMenuItem;

SearchEdM: TMenuItem;

ReplaceEdM: TMenuItem;

Panel1: TPanel;

Panel2: TPanel;

SpeedButton1: TSpeedButton;

SpeedButton2: TSpeedButton;

SpeedButton3: TSpeedButton;

SpeedButton4: TSpeedButton;

SpeedButton5: TSpeedButton;

SpeedButton6: TSpeedButton;

SpeedButton7: TSpeedButton;

SpeedButton8: TSpeedButton;

SpeedButton9: TSpeedButton;

SpeedButton10: TSpeedButton;

SpeedButton11: TSpeedButton;

SpeedButton12: TSpeedButton;

SpeedButton13: TSpeedButton;

ComboBox1: TComboBox;

SpinEdit1: TSpinEdit;

N8: TMenuItem;

Edit1: TEdit;

WebBrowser1: TWebBrowser;

SpeedButton14: TSpeedButton;

Label1: TLabel;

Label2: TLabel;

FileListBox1: TFileListBox;

DirectoryListBox1: TDirectoryListBox;

Label3: TLabel;

SpeedButton15: TSpeedButton;

SpeedButton16: TSpeedButton;

SpeedButton17: TSpeedButton;

SpeedButton18: TSpeedButton;

SpeedButton19: TSpeedButton;

SpeedButton20: TSpeedButton;

SpeedButton21: TSpeedButton;

SpeedButton22: TSpeedButton;

SpeedButton23: TSpeedButton;

SpeedButton24: TSpeedButton;

IdEncoderXXE1: TIdEncoderXXE;

IdDecoderXXE1: TIdDecoderXXE;

N9: TMenuItem;

N10: TMenuItem;

N11: TMenuItem;

IdDecoderMIME1: TIdDecoderMIME;

IdEncoderMIME1: TIdEncoderMIME;

SpeedButton25: TSpeedButton;

N12: TMenuItem;

N13: TMenuItem;

N14: TMenuItem;

N15: TMenuItem;

N16: TMenuItem;

N17: TMenuItem;

N18: TMenuItem;

procedure OpenFileMClick(Sender: TObject);

procedure SaveFileMClick(Sender: TObject);

procedure PrintMClick(Sender: TObject);

procedure PrintSetupMClick(Sender: TObject);

procedure ExitFileMClick(Sender: TObject);

procedure FontFmtMClick(Sender: TObject);

procedure ColorFmtMClick(Sender: TObject);

procedure UndoEdMClick(Sender: TObject);

procedure CutEdMClick(Sender: TObject);

procedure CopyEdMClick(Sender: TObject);

procedure InsertEdMClick(Sender: TObject);

procedure SelAllEdMClick(Sender: TObject);

procedure SearchEdMClick(Sender: TObject);

procedure FindDlgFind(Sender: TObject);

procedure ReplaceEdMClick(Sender: TObject);

procedure ReplaceDlgFind(Sender: TObject);

procedure ReplaceDlgReplace(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure ComboBox1Change(Sender: TObject);

procedure RichEdClick(Sender: TObject);

procedure SpinEdit1Change(Sender: TObject);

procedure N8Click(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure SpeedButton1Click(Sender: TObject);

procedure SpeedButton2Click(Sender: TObject);

procedure SpeedButton3Click(Sender: TObject);

procedure SpeedButton8Click(Sender: TObject);

procedure SpeedButton7Click(Sender: TObject);

procedure SpeedButton4Click(Sender: TObject);

procedure SpeedButton5Click(Sender: TObject);

procedure SpeedButton6Click(Sender: TObject);

procedure SpeedButton9Click(Sender: TObject);

procedure SpeedButton10Click(Sender: TObject);

procedure SpeedButton13Click(Sender: TObject);

procedure SpeedButton11Click(Sender: TObject);

procedure SpeedButton12Click(Sender: TObject);

procedure Edit1Click(Sender: TObject);

procedure Edit1KeyPress(Sender: TObject; var Key: Char);

procedure WebBrowser1TitleChange(ASender: TObject; const Text: WideString);

procedure SpeedButton14Click(Sender: TObject);

procedure RichEdChange(Sender: TObject);

procedure FileListBox1DblClick(Sender: TObject);

procedure SpeedButton15Click(Sender: TObject);

procedure SpeedButton16Click(Sender: TObject);

procedure SpeedButton17Click(Sender: TObject);

procedure SpeedButton23Click(Sender: TObject);

procedure SpeedButton21Click(Sender: TObject);

procedure SpeedButton22Click(Sender: TObject);

procedure SpeedButton24Click(Sender: TObject);

procedure FormResize(Sender: TObject);

procedure N10Click(Sender: TObject);

procedure N11Click(Sender: TObject);

procedure SpeedButton25Click(Sender: TObject);

procedure N17Click(Sender: TObject);

procedure N18Click(Sender: TObject);

procedure N14Click(Sender: TObject);

procedure N15Click(Sender: TObject);

private

procedure CMDialogKey(var msg: TCMDialogKey); // Процедура доступа к Key = VK\_TAB

message CM\_DIALOGKEY;

procedure TrimWorkingSet; // Процедура очистки ОЗУ по завершению сеанса

function EncodeBase64(Value: string): string; // Функция кодирования стандартом Base64

function DecodeBase64(Value: string): string; // Функция декодирования стандартом Base64

function GronsfeldEncipher(toCode, K: string): string; // Функция шифрования информации

function GronsfeldDecipher(toDecode, K: string): string; // Функция дешифрования информации

{ Private declarations }

public

{ Public declarations }

end;

var

MainFrm: TMainFrm;

implementation

{$R \*.dfm}

uses Contacts;

procedure TMainFrm.FormCreate(Sender: TObject); // Процедура стартовых параметров по созданию формы

begin

RichEd.Font.Size := 14; // Стартовый шрифт текста

RichEd.Font.Name := 'Times New Roman'; // Стартовый шрифт текста

ComboBox1.Items := Screen.Fonts; // Заполнения ComboBox1 шрифтами, которые имеются на пользовательском компьютере

ComboBox1.Text := 'Times New Roman'; // Установка стандартной гарнитуры шрифта

RichEd.Lines.LoadFromFile('..\APPostate\resource\indent'); // Подгрузка файла со стандартным стилем для отступа от левого края, т.к. RichEdit не позволяет этого сделать не нарушив при этом отступы с других сторон

RichEd.Lines.Add(''); // Добавление пустой строки для более красивого форматирования и удобной работы

WebBrowser1.Hide; // Сокрытие окна браузера

end;

procedure TMainFrm.FormResize(Sender: TObject); // Процедура сохранения/изменения параметров формы

begin

MainFrm.ClientHeight := 577;

MainFrm.ClientWidth := 898;

{if MainFrm.Width = Screen.Width then

RichEd.Left := Panel1.Width div 4;}

end;

//=============== Функции шифрования и дешифрования информации =================

function TMainFrm.GronsfeldEncipher(toCode, K: string): string;

var

i,

T,

\_T: integer;

begin

for i := 1 to length(toCode) do

begin

\_T := ord(toCode[i]);

T := (ord(toCode[i]) + (ord(K[(pred(i) mod length(K)) + 1]) - ord('0')));

toCode[i] := Char(T);

end;

GronsfeldEncipher := toCode;

end;

function TMainFrm.GronsfeldDecipher(toDecode, K: string): string;

var

i, T: integer;

begin

for i := 1 to length(toDecode) do

begin

T := (ord(toDecode[i]) - (ord(K[(pred(i) mod length(K)) + 1]) - ord('0')));

toDecode[i] := Chr(T);

end;

GronsfeldDecipher := toDecode;

end;

//==============================================================================

//============== Функции кодирования и декодирования информации ================

function TMainFrm.EncodeBase64(Value: String): String;

const

b64alphabet: PChar = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'

+'АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯабвгдеёжзийклмнопрстуфхцчшщъыьэюя'

+'0123456789+/';

pad: PChar = '====';

function EncodeChunk(const Chunk: String): String;

var

W: LongWord;

i,

n: Byte;

begin

n := Length(Chunk);

W := 0;

for i := 0 to n - 1 do

W := W + Ord(Chunk[i + 1]) shl ((2 - i) \* 8);

Result := b64alphabet[(W shr 18) and $3f] +

b64alphabet[(W shr 12) and $3f] +

b64alphabet[(W shr 06) and $3f] +

b64alphabet[(W shr 00) and $3f];

if n <> 3 then

Result := Copy(Result, 0, n + 1) + Copy(pad, 0, 3 - n); // Добавление отступа, если в Len не 24 бита

end;

begin

Result := '';

while Length(Value) > 0 do

begin

Result := Result + EncodeChunk(Copy(Value, 0, 3));

Delete(Value, 1, 3);

end;

end;

function TMainFrm.DecodeBase64(Value: String): String;

const

b64alphabet: PChar = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'

+'АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯабвгдеёжзийклмнопрстуфхцчшщъыьэюя'

+'0123456789+/';

function DecodeChunk(const Chunk: String): String;

var

W: LongWord;

i: Byte;

begin

W := 0;

Result := '';

for i := 1 to 4 do

if Pos(Chunk[i], b64alphabet) <> 0 then

W := W + Word((Pos(Chunk[i], b64alphabet) - 1)) shl ((4 - i) \* 6);

for i := 1 to 3 do

Result := Result + Chr(W shr ((3 - i) \* 8) and $ff);

end;

begin

Result := '';

if Length(Value) mod 4 <> 0 then

Exit;

while Length(Value) > 0 do

begin

Result := Result + DecodeChunk(Copy(Value, 0, 4));

Delete(Value, 1, 4);

end;

end;

//==============================================================================

procedure TMainFrm.N10Click(Sender: TObject); // Кодирование информации

begin

if RichEd.SelText <> '' then

RichEd.SelText := EncodeBase64(RichEd.SelText);

end;

procedure TMainFrm.N11Click(Sender: TObject); // Декодирование информации

begin

if RichEd.SelText <> '' then

RichEd.SelText := DecodeBase64(RichEd.SelText);

end;

procedure TMainFrm.N14Click(Sender: TObject); // Шифрование информации по ключевому слову

begin

if RichEd.SelText <> '' then

RichEd.SelText := GronsfeldEncipher(RichEd.SelText,

InputBox('Ключевое слово для шифрования', // Ввод ключевого слова

'Введите ключевое слово', ''));

end;

procedure TMainFrm.N15Click(Sender: TObject); // Дешифрование информации по ключевому слову

begin

if RichEd.SelText <> '' then

RichEd.SelText := GronsfeldDecipher(RichEd.SelText,

InputBox('Ключевое слово для дешифрования', // Ввод ключевого слова

'Введите ключевое слово', ''));

end;

procedure TMainFrm.N17Click(Sender: TObject);

begin

HtmlHelp(handle,'help/help.chm',HH\_DISPLAY\_TOPIC,0);

end;

procedure TMainFrm.N18Click(Sender: TObject);

begin

Cont.Show;

end;

procedure TMainFrm.N8Click(Sender: TObject); // Процедура создания документа

begin

if (OpenDlg.FileName <> '') or (SaveDlg.FileName <> '') then

begin

OpenDlg.FileName := '';

SaveDlg.FileName := '';

MainFrm.Caption := 'APPostate';

end;

RichEd.Clear;

FormCreate(Sender);

end;

procedure TMainFrm.OpenFileMClick(Sender: TObject); // Процедура открытия документа

begin

if OpenDlg.Execute then

begin

RichEd.Lines.LoadFromFile(OpenDlg.FileName); // Загрузка документа из файла

MainFrm.Caption := 'APPostate - ' + OpenDlg.FileName;

end;

end;

procedure TMainFrm.SaveFileMClick(Sender: TObject); // Процедура сохранения документа

begin

if OpenDlg.FileName <> '' then // Получение имени файла в случае, если документ был открыт ранее

SaveDlg.FileName := OpenDlg.FileName;

if SaveDlg.Execute then // Сохранения с параметрами

begin

RichEd.PlainText := (SaveDlg.FilterIndex>1);

RichEd.Lines.SaveToFile(SaveDlg.FileName);

RichEd.PlainText := false;

MainFrm.Caption := 'APPostate - ' + SaveDlg.FileName;

end;

end;

procedure TMainFrm.PrintMClick(Sender: TObject); // Вызов открытия диалога печати

begin

if PrintDlg.Execute then

RichEd.Print('');

end;

procedure TMainFrm.PrintSetupMClick(Sender: TObject); // Вызов открытия диалога настройки принтера

begin

PrintSetupDlg.Execute;

end;

procedure TMainFrm.Edit1Click(Sender: TObject); // Очистка поля ввода по клику

begin

Edit1.Text:='';

end;

procedure TMainFrm.Edit1KeyPress(Sender: TObject; var Key: Char); // Поиск в браузере по нажатию Enter

begin

if Key = #13 then

begin

WebBrowser1.Navigate('http://google.com.ru/search?q='+Edit1.Text);

WebBrowser1.Visible := True;

WebBrowser1.Show;

SpeedButton14.Visible := True;

Edit1.Width := 141;

end;

end;

procedure TMainFrm.ExitFileMClick(Sender: TObject); // Уничтожение программы по закрытию, чтобы в оперативной памяти не оставалось данных

var

buttonSelected: integer;

begin

buttonSelected := MessageDlg('Сохранить изменения в документе?', mtCustom, // Вывод соощения с предупреждением о сохранении файла

[mbYes, mbNo, mbCancel], 0);

if buttonSelected = mrYes then // Обработка случая, когда пользователь нажимает "Yes"

begin

if OpenDlg.FileName <> '' then // Получение имени файла в случае, если документ был открыт ранее

begin

SaveDlg.FileName := OpenDlg.FileName;

RichEd.PlainText := (SaveDlg.FilterIndex>1);

RichEd.Lines.SaveToFile(SaveDlg.FileName);

RichEd.PlainText := false;

end

else

if SaveDlg.Execute then // Сохранения с параметрами

begin

RichEd.PlainText := (SaveDlg.FilterIndex>1);

RichEd.Lines.SaveToFile(SaveDlg.FileName);

RichEd.PlainText := false;

end;

Application.Terminate; // Уничтожение приложения

TrimWorkingSet; // Очистка всех переменных в оперативной памяти

end;

if buttonSelected = mrNo then // Обработка случая, когда пользователь нажимает "No"

begin

Application.Terminate; // Уничтожение приложения

TrimWorkingSet; // Очистка всех переменных в оперативной памяти

end;

end;

procedure TMainFrm.FormClose(Sender: TObject; var Action: TCloseAction); // Уничтожение программы по закрытию, чтобы в оперативной памяти не оставалось данных

var

buttonSelected: integer;

begin

buttonSelected := MessageDlg('Сохранить изменения в документе?', mtCustom, // Вывод соощения с предупреждением о сохранении файла

[mbYes, mbNo, mbCancel], 0);

if buttonSelected = mrYes then // Обработка случая, когда пользователь нажимает "Yes"

begin

if OpenDlg.FileName <> '' then // Получение имени файла в случае, если документ был открыт ранее

begin

SaveDlg.FileName := OpenDlg.FileName;

RichEd.PlainText := (SaveDlg.FilterIndex>1);

RichEd.Lines.SaveToFile(SaveDlg.FileName);

RichEd.PlainText := false;

end

else

if SaveDlg.Execute then // Сохранения с параметрами

begin

RichEd.PlainText := (SaveDlg.FilterIndex>1);

RichEd.Lines.SaveToFile(SaveDlg.FileName);

RichEd.PlainText := false;

end;

Application.Terminate; // Уничтожение приложения

TrimWorkingSet; // Очистка всех переменных в оперативной памяти

end;

if buttonSelected = mrNo then // Обработка случая, когда пользователь нажимает "No"

begin

Application.Terminate; // Уничтожение приложения

TrimWorkingSet; // Очистка всех переменных в оперативной памяти

end;

end;

procedure TMainFrm.FontFmtMClick(Sender: TObject); // Процедура изменения параметров шрифта

begin

FontDlg.Font.Name := RichEd.SelAttributes.Name; // Отображение в диалоге изменения шрифта текущих параметров выделенного текста

FontDlg.Font.Color := RichEd.SelAttributes.Color;

FontDlg.Font.Charset := RichEd.SelAttributes.Charset;

FontDlg.Font.Size := RichEd.SelAttributes.Size;

FontDlg.Font.Style := RichEd.SelAttributes.Style;

if not FontDlg.Execute then

exit;

RichEd.SelAttributes.Name := FontDlg.Font.Name; // Изменение текущих параметров выделенного текста на выбранные в диалоге изменения шрифта

RichEd.SelAttributes.Color := FontDlg.Font.Color;

RichEd.SelAttributes.Charset := FontDlg.Font.Charset;

RichEd.SelAttributes.Size := FontDlg.Font.Size;

RichEd.SelAttributes.Style := FontDlg.Font.Style;

end;

procedure TMainFrm.ComboBox1Change(Sender: TObject); // Процедура изменения гарнитуры шрифта из ComboBox1

begin

if RichEd.SelText <> '' then

RichEd.SelAttributes.Name := ComboBox1.Items.Strings[ComboBox1.ItemIndex];

end;

procedure TMainFrm.SpinEdit1Change(Sender: TObject); // Процедура изменения размера шрифта согласно данным в SpinEdit1

begin

RichEd.SelAttributes.Size := SpinEdit1.Value;

end;

procedure TMainFrm.TrimWorkingSet; // Процедура очистки оперативной памяти по завершению программы

var

MainHandle: THandle;

begin

if Win32Platform = VER\_PLATFORM\_WIN32\_NT then

begin

MainHandle := OpenProcess(PROCESS\_ALL\_ACCESS, false, GetCurrentProcessID);

SetProcessWorkingSetSize(MainHandle, DWORD(-1), DWORD(-1));

CloseHandle(MainHandle);

end;

end;

procedure TMainFrm.RichEdChange(Sender: TObject); // Процедура подсчёта количества строк, слов и символов по изменению в RichEdit

var

i,

r,

k: Integer;

S: String;

begin

Application.ProcessMessages;

S := RichEd.Lines.Text;

k := 0;

r := 0;

for i := 1 to Length(S) do

begin // Перебор всей длины поля RichEdit

if (S[i] = ' ') or (S[i] = #13) then // В случае, если встречается пробел или перенос на новую строку, разумно предполагать, что ожидается новое слово

inc(k);

if (S[i] = #13) then // Подсчёт переносов строки

inc(r);

end;

if Length(S) <> 0 then

Label1.Caption := 'Кол-во символов: ' + IntToStr(Length(S)-2);

Label2.Caption := 'Кол-во слов: ' + IntToStr(k);

Label3.Caption := 'Кол-во строк: ' + IntToStr(r);

end;

procedure TMainFrm.RichEdClick(Sender: TObject); // Процедура вовзрата текущего значения текста в SinEdit1

begin

SpinEdit1.Value := RichEd.SelAttributes.Size;

end;

procedure TMainFrm.CMDialogKey(var msg: TCMDialogKey); // Процедура отлова табуляции

begin

if msg.Charcode <> VK\_TAB then

inherited;

end;

procedure TMainFrm.ColorFmtMClick(Sender: TObject); // Процедура определения цвета текста (принцип аналогичен процедуре FontFmtMClick)

begin

ColorDlg.Color := RichEd.SelAttributes.Color;

if not ColorDlg.Execute then

exit;

RichEd.SelAttributes.Color := ColorDlg.Color;

end;

procedure TMainFrm.UndoEdMClick(Sender: TObject); // Процедура отмены совершенного действия с компонентом RichEd

begin

if RichEd.CanUndo then

RichEd.Undo;

end;

procedure TMainFrm.WebBrowser1TitleChange(ASender: TObject; // Отображение в Edit1 текущего запроса

const Text: WideString);

begin

Edit1.Text := WebBrowser1.LocationName;

end;

procedure TMainFrm.CutEdMClick(Sender: TObject); // Процедура вырезки текста из RichEd

begin

if RichEd.SelText <> '' then

RichEd.CutToClipboard;

end;

procedure TMainFrm.CopyEdMClick(Sender: TObject); // Процедура rjgbhjdfybt текста из RichEd

begin

if RichEd.SelText <> '' then

RichEd.CopyToClipboard;

end;

procedure TMainFrm.InsertEdMClick(Sender: TObject); // Процедура вставки текста из RichEd

begin

if Clipboard.HasFormat(CF\_TEXT) then

RichEd.PasteFromClipboard;

end;

procedure TMainFrm.SelAllEdMClick(Sender: TObject); // Процедура выделения всего текста в RichEd

begin

RichEd.SelectAll;

end;

procedure TMainFrm.SearchEdMClick(Sender: TObject); // Процедура открытия диалога поиска текста в RichEd

begin

FindDlg.Execute;

end;

procedure TMainFrm.FileListBox1DblClick(Sender: TObject); // Процедура открытия документа из FileListBox1

begin

MainFrm.Caption := 'APPostate - ' + FileListBox1.FileName;

RichEd.Lines.LoadFromFile(FileListBox1.FileName);

end;

procedure TMainFrm.FindDlgFind(Sender: TObject); // Процедура поиска текста в RichEd

var

StartPos, // Стартовая позация

ToPos, // Размерность поля поиска

FoundPos: Integer; // Найденная позация

Opt: TSearchTypes;

begin

StartPos := RichEd.SelStart + RichEd.SelLength;

ToPos := Length(RichEd.Text) - StartPos;

if frMatchCase in FindDlg.Options then

Opt := [stMatchCase];

if frWholeWord in FindDlg.Options then

Opt := Opt + [stWholeWord];

FoundPos := RichEd.FindText(FindDlg.FindText, StartPos, ToPos, Opt); // Вывод позации текста

if FoundPos <> -1 then // Обработка случая необнаружения текста

begin

RichEd.SelStart := FoundPos;

RichEd.SelLength := Length(FindDlg.FindText);

RichEd.SetFocus;

end

else

ShowMessage('Текст не найден!');

end;

procedure TMainFrm.ReplaceEdMClick(Sender: TObject); // Процедура открытия диалога замены текста в RichEd

begin

ReplaceDlg.Execute;

end;

procedure TMainFrm.ReplaceDlgFind(Sender: TObject); // Процедура поиска текста в RichEd для его замены (процедура аналогично процедуре FindDlgFind и работает на кнопку "Найти далее" в диалоге)

var

StartPos,

ToPos,

FoundPos: Integer;

Opt: TSearchTypes;

begin

StartPos := RichEd.SelStart + RichEd.SelLength;

ToPos := Length(RichEd.Text) - StartPos;

if frMatchCase in ReplaceDlg.Options then

Opt := [stMatchCase];

if frWholeWord in ReplaceDlg.Options then

Opt := Opt + [stWholeWord];

FoundPos := RichEd.FindText(ReplaceDlg.FindText, StartPos, ToPos, Opt); // Получение позации текста

if FoundPos<>-1 then

begin // Обработка случая необнаружения текста

RichEd.SelStart := FoundPos;

RichEd.SelLength := Length(ReplaceDlg.FindText);

RichEd.SetFocus;

end

else

ShowMessage('Текст не найден!');

end;

procedure TMainFrm.ReplaceDlgReplace(Sender: TObject); // Процедура замены текста в RichEd

var

i,

StartPos,

ToPos,

FoundPos: Integer;

Opt: TSearchTypes;

begin

if frMatchCase in ReplaceDlg.Options then

Opt := [stMatchCase];

if frWholeWord in ReplaceDlg.Options then

Opt := Opt + [stWholeWord];

i := 0;

repeat // Регулярный поиск текста и его изменение

StartPos := RichEd.SelStart + RichEd.SelLength;

ToPos := Length(RichEd.Text) - StartPos;

FoundPos := RichEd.FindText(ReplaceDlg.FindText, StartPos, ToPos, Opt);

if FoundPos <> -1 then

begin

RichEd.SelStart := FoundPos;

RichEd.SelLength := Length(ReplaceDlg.FindText);

RichEd.SelText := ReplaceDlg.ReplaceText;

RichEd.SetFocus;

inc(i);

end;

until (FoundPos = -1) or not(frReplaceAll in ReplaceDlg.Options); // Процедура продолжается до тех пока пока не достигнет конца документа или до случая, когда ничего не будет найдено

if i = 0 then

ShowMessage('Текст не найден!')

else

if frReplaceAll in ReplaceDlg.Options then

ShowMessage('Произведено ' + IntToStr(i) + ' замен');

end;

procedure TMainFrm.SpeedButton21Click(Sender: TObject); // Процедура выравнивания текста по правому краю

begin

RichEd.Paragraph.Alignment := taRightJustify;

end;

procedure TMainFrm.SpeedButton22Click(Sender: TObject); // Процедура выравнивания текста по центру

begin

RichEd.Paragraph.Alignment := taCenter;

end;

procedure TMainFrm.SpeedButton23Click(Sender: TObject); // Процедура выравнивания текста по левому краю

begin

RichEd.Paragraph.Alignment := taLeftJustify;

end;

procedure TMainFrm.SpeedButton24Click(Sender: TObject); // Процедура включения/отключения маркерованного списка

begin

case SpeedButton24.Tag of

0 : begin

SpeedButton24.Tag := 1;

RichEd.Paragraph.Numbering := nsBullet;

end;

1 : begin

SpeedButton24.Tag := 0;

RichEd.Paragraph.Numbering := nsNone;

end;

end;

end;

procedure TMainFrm.SpeedButton15Click(Sender: TObject); // Процедура включения/отключения жирного шрифта

begin

case SpeedButton15.Tag of

0 : begin

SpeedButton15.Tag := 1;

RichEd.SelAttributes.Style := RichEd.SelAttributes.Style + [fsBold];

end;

1 : begin

SpeedButton15.Tag := 0;

RichEd.SelAttributes.Style := RichEd.SelAttributes.Style - [fsBold];

end;

end;

end;

procedure TMainFrm.SpeedButton16Click(Sender: TObject); // Процедура включения/отключения курсивного шрифта

begin

case SpeedButton16.Tag of

0 : begin

SpeedButton16.Tag := 1;

RichEd.SelAttributes.Style := RichEd.SelAttributes.Style + [fsItalic];

end;

1 : begin

SpeedButton16.Tag := 0;

RichEd.SelAttributes.Style := RichEd.SelAttributes.Style - [fsItalic];

end;

end;

end;

procedure TMainFrm.SpeedButton17Click(Sender: TObject); // Процедура включения/отключения подчёркнутого шрифта

begin

case SpeedButton17.Tag of

0 : begin

SpeedButton17.Tag := 1;

RichEd.SelAttributes.Style := RichEd.SelAttributes.Style + [fsUnderline];

end;

1 : begin

SpeedButton17.Tag := 0;

RichEd.SelAttributes.Style := RichEd.SelAttributes.Style - [fsUnderline];

end;

end;

end;

procedure TMainFrm.SpeedButton14Click(Sender: TObject); // Процедура закрытия браузера

begin

WebBrowser1.Visible := False;

WebBrowser1.Hide;

Edit1.Text := 'Поиск в Google';

Edit1.Width := 169;

end;

//======== Процедуры ассоциации кнопок меню со SpeedButton-кнопками ============

procedure TMainFrm.SpeedButton10Click(Sender: TObject);

begin

ReplaceEdM.Click;

end;

procedure TMainFrm.SpeedButton11Click(Sender: TObject);

begin

FontFmtM.Click;

end;

procedure TMainFrm.SpeedButton12Click(Sender: TObject);

begin

ColorFmtM.Click;

end;

procedure TMainFrm.SpeedButton13Click(Sender: TObject);

begin

PrintM.Click;

end;

procedure TMainFrm.SpeedButton1Click(Sender: TObject);

begin

N8.Click;

end;

procedure TMainFrm.SpeedButton2Click(Sender: TObject);

begin

OpenFileM.Click;

end;

procedure TMainFrm.SpeedButton3Click(Sender: TObject);

begin

SaveFileM.Click;

end;

procedure TMainFrm.SpeedButton4Click(Sender: TObject);

begin

CutEdM.Click;

end;

procedure TMainFrm.SpeedButton5Click(Sender: TObject);

begin

CopyEdM.Click;

end;

procedure TMainFrm.SpeedButton6Click(Sender: TObject);

begin

InsertEdM.Click;

end;

procedure TMainFrm.SpeedButton7Click(Sender: TObject);

begin

SelAllEdM.Click;

end;

procedure TMainFrm.SpeedButton8Click(Sender: TObject);

begin

UndoEdM.Click;

end;

procedure TMainFrm.SpeedButton9Click(Sender: TObject);

begin

SearchEdM.Click;

end;

//==============================================================================

procedure TMainFrm.SpeedButton25Click(Sender: TObject); // Процедура перевода текста в одной раскладки на другую

var

s: string;

i: integer;

begin

if RichEd.SelText <> '' then

begin

s := RichEd.SelText;

for i := 1 to length(s) do

begin

case s[i] of // Библиотека перевода символа с кириллицы на латиницы методом CHAR

// Перевод с большого регистра английской раскладки на большой регистр русской раскладки

'F' : s[i] := 'А';

'<' : s[i] := 'Б';

'D' : s[i] := 'В';

'U' : s[i] := 'Г';

'L' : s[i] := 'Д';

'T' : s[i] := 'Е';

':' : s[i] := 'Ж';

'P' : s[i] := 'З';

'B' : s[i] := 'И';

'Q' : s[i] := 'Й';

'R' : s[i] := 'К';

'K' : s[i] := 'Л';

'V' : s[i] := 'М';

'Y' : s[i] := 'Н';

'J' : s[i] := 'О';

'G' : s[i] := 'П';

'H' : s[i] := 'Р';

'C' : s[i] := 'С';

'N' : s[i] := 'Т';

'E' : s[i] := 'У';

'A' : s[i] := 'Ф';

'{' : s[i] := 'Х';

'W' : s[i] := 'Ц';

'X' : s[i] := 'Ч';

'I' : s[i] := 'Ш';

'O' : s[i] := 'Щ';

'}' : s[i] := 'Ъ';

'S' : s[i] := 'Ы';

'M' : s[i] := 'Ь';

'"' : s[i] := 'Э';

'>' : s[i] := 'Ю';

'Z' : s[i] := 'Я';

// Перевод с малого регистра английской раскладки на малый регистр русской раскладки

'f' : s[i] := 'а';

',' : s[i] := 'б';

'd' : s[i] := 'в';

'u' : s[i] := 'г';

'l' : s[i] := 'д';

't' : s[i] := 'е';

';' : s[i] := 'ж';

'p' : s[i] := 'з';

'b' : s[i] := 'и';

'q' : s[i] := 'й';

'r' : s[i] := 'к';

'k' : s[i] := 'л';

'v' : s[i] := 'м';

'y' : s[i] := 'н';

'j' : s[i] := 'о';

'g' : s[i] := 'п';

'h' : s[i] := 'р';

'c' : s[i] := 'с';

'n' : s[i] := 'т';

'e' : s[i] := 'у';

'a' : s[i] := 'ф';

'[' : s[i] := 'х';

'w' : s[i] := 'ц';

'x' : s[i] := 'ч';

'i' : s[i] := 'ш';

'o' : s[i] := 'щ';

']' : s[i] := 'ъ';

's' : s[i] := 'ы';

'm' : s[i] := 'ь';

#39 : s[i] := 'э';

'.' : s[i] := 'ю';

'z' : s[i] := 'я';

// Перевод с большого регистра русской раскладки на большой регистр английской раскладки

'А' : s[i] := 'F';

'Б' : s[i] := '<';

'В' : s[i] := 'D';

'Г' : s[i] := 'U';

'Д' : s[i] := 'L';

'Е' : s[i] := 'T';

'Ж' : s[i] := ':';

'З' : s[i] := 'P';

'И' : s[i] := 'B';

'Й' : s[i] := 'Q';

'К' : s[i] := 'R';

'Л' : s[i] := 'K';

'М' : s[i] := 'V';

'Н' : s[i] := 'Y';

'О' : s[i] := 'J';

'П' : s[i] := 'G';

'Р' : s[i] := 'H';

'С' : s[i] := 'C';

'Т' : s[i] := 'N';

'У' : s[i] := 'E';

'Ф' : s[i] := 'A';

'Х' : s[i] := '{';

'Ц' : s[i] := 'W';

'Ч' : s[i] := 'X';

'Ш' : s[i] := 'I';

'Щ' : s[i] := 'O';

'Ъ' : s[i] := '}';

'Ы' : s[i] := 'S';

'Ь' : s[i] := 'M';

'Э' : s[i] := '"';

'Ю' : s[i] := '>';

'Я' : s[i] := 'Z';

// Перевод с малого регистра русской раскладки на малый регистр английской раскладки

'а' : s[i] := 'f';

'б' : s[i] := ',';

'в' : s[i] := 'd';

'г' : s[i] := 'u';

'д' : s[i] := 'l';

'е' : s[i] := 't';

'ж' : s[i] := ';';

'з' : s[i] := 'p';

'и' : s[i] := 'b';

'й' : s[i] := 'q';

'к' : s[i] := 'r';

'л' : s[i] := 'k';

'м' : s[i] := 'v';

'н' : s[i] := 'y';

'о' : s[i] := 'j';

'п' : s[i] := 'g';

'р' : s[i] := 'h';

'с' : s[i] := 'c';

'т' : s[i] := 'n';

'у' : s[i] := 'e';

'ф' : s[i] := 'a';

'х' : s[i] := '[';

'ц' : s[i] := 'w';

'ч' : s[i] := 'x';

'ш' : s[i] := 'i';

'щ' : s[i] := 'o';

'ъ' : s[i] := ']';

'ы' : s[i] := 's';

'ь' : s[i] := 'm';

'э' : s[i] := #39;

'ю' : s[i] := '.';

'я' : s[i] := 'z';

// Необходимо сделать знаковую библиотеку

end;

end;

RichEd.SelText := s;

end;

end;

end.

# 3.6 Вызов программы

Запуск программы производится по двойному нажатию на файл «APPostate.exe», который изображен на рисунке 8.

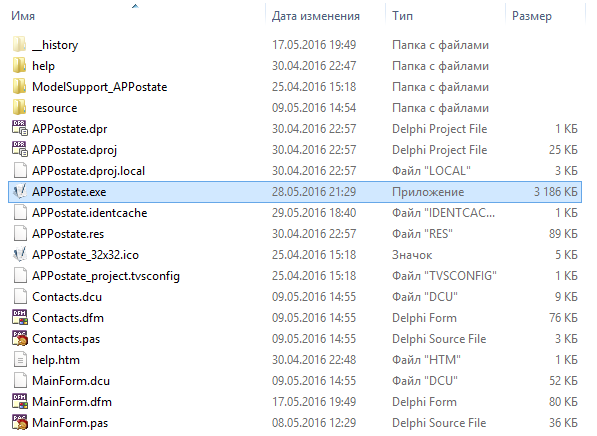


Рисунок 8 – Файл «APPostate.exe»

Для функционирования программы из оперативной памяти под неё выделяется приблизительно 8 Мб.

Окно программы с контрольным примером ввода данных представлено на рисунке 8.

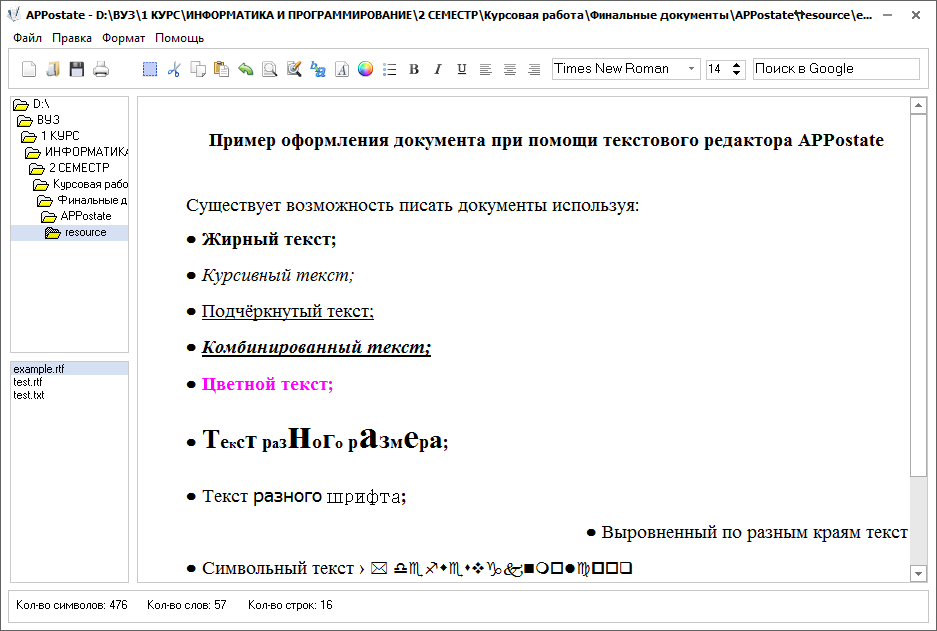


Рисунок 9 – Окно программы с примером ввода данных

# Заключение

В результате выполнения курсовой работы был изучен процесс работы с программными модулями для обработки текстовой информации и полученные теоретические знания были применены на практике.

Были достигнуты поставленные цели и задачи, а именно:

* была изучена предметная область решаемой задачи;
* был разработан программный модуль для обработки текстовой информации, отвечающий требованиям технического задания.

Список литературы

1. 1. Редактор WordPad [Электронный ресурс] / Microsoft [Офиц. сайт]. 2016. URL: [http://windows.microsoft.com/ru-ru/windows/using-wordpad](http://windows.microsoft.com/ru-ru/windows/using-wordpad#1TC=windows-7) (дата обращения: 12.04.2016).

   [↑](#endnote-ref-1)
2. 1. AkelPad [Электронный ресурс] / Библиотека бесплатных программ. [Офиц. сайт]. 2015. URL: <http://biblprog.org.ua/ru/akelpad> (дата обращения: 12.04.2016).

   [↑](#endnote-ref-2)
3. 1. Блокнот (программа) [Электронный ресурс] / Википедия. Свободная энциклопедия [Офиц. сайт]. 2015. URL: [https://ru.wikipedia.org/wiki/Блокнот\_(программа)](https://ru.wikipedia.org/wiki/Определитель) (дата обращения: 12.04.2016).
   2. ЕСПД. Общие положения [Текст] : ГОСТ 19.701-90. - Введ. 1980-01-01. – М. : ИПК Изд-во стандартов, 1980 (дата обращения: 12.04.2016).
   3. ЕСПД. Общие положения [Текст] : ГОСТ ГОСТ 19.105 - 78. - Введ. 1980-01-01. – М. : ИПК Изд-во стандартов, 1980 (дата обращения: 12.04.2016).
   4. ЕСПД. Основные надписи [Текст] : ГОСТ 19.104–78. - Введ. 1980-01-01. – М. : ИПК Изд-во стандартов, 1980 (дата обращения: 12.04.2016).
   5. ЕСПД. Общие положения [Текст] : ГОСТ 19.001–77. - Введ. 1980-01-01. – М. : ИПК Изд-во стандартов, 1980 (дата обращения: 12.04.2016).
   6. ИСО 9:1995 — действующий стандарт, принятый Международной организацией по стандартизации (дата обращения: 12.04.2016).
   7. ГОСТ 7.79-2000 — адаптация ИСО 9, принятая в России и некоторых странах СНГ (дата обращения: 12.04.2016).

   [↑](#endnote-ref-3)