

SECTION 2. Applied mathematics. Mathematical modeling.

Shevtsov Alexandr Nikolayevich
candidate of technical Sciences,
President, Theoretical & Applied Science, LLP,
associate Professor of the Department «Applied mathematics»
Taraz State University named after M.Kh. Dulati, Kazakhstan

Smailova Ylmeken Muhitovna
candidate of physical and mathematical Sciences, associate Professor,
Branch of JSC National centre of improvement of qualification of «Orleu» IPKPR regions of
Zhambyl, Kazakhstan

Shyrynkhanova Dinara Zhaksylykovna
1 year magistr of the speciality "Information systems "
Taraz State University named after M.Kh. Dulati, Kazakhstan

TIME CHARACTERISTICS OF SCRIPTS WORD IN DELPHI ENVIRONMENT

This article studies and develops methods of optimization of work processes and the choice of algorithms for processing of server scripts Word.

Keywords: optimization, Word, time.

ВРЕМЕННЫЕ ХАРАКТЕРИСТИКИ ВЫПОЛНЕНИЯ СКРИПТОВ WORD В СРЕДЕ ДЕЛЬФИ

В данной статье рассматриваются методы оптимизации процессов работы и выбора алгоритмов обработки скриптов сервера Word.

Ключевые слова: оптимизация, время.

Разработка программ зачастую связана с процессом оптимизации. Известно, что на этот этап разработки отводится больше всего времени. Рассмотрим программу в Delphi, основной функцией которой является обработка коллекций, их сравнение и формирование новой структуры данных в соответствии с поставленной блок схемой.

Тестирование будем вести на компьютере со следующими характеристиками (рис.1).

Для оценки времени выполнения отдельных скриптов используем алгоритм, привязанный к текущему времени системы.

```
Var  
d: TDateTime;  
-----  
d:=Now;  
  
-----Скрипт-----  
  
Label14.Caption:=FormatDateTime('hh:mm:ss:zzz', Now()-d);
```

Компонент	Подробно	Оценка	Общая оценка
Процессор	Intel(R) Core(TM)2 Duo CPU T6600 @ 2.20GHz	5,7	 <p>Определяется наименьшей оценкой</p>
Память (ОЗУ)	4,00 ГБ	5,7	
Графика	NVIDIA GeForce GT 240M (Microsoft Corporation - WDDM v1.2)	6,4	
Графика для игр	2798 МБ графической памяти доступно	6,4	
Основной жесткий диск	Свободно: 38GB (всего: 146GB)	5,8	
Windows 8 Профессиональная			

Система

Производитель	Acer
Модель	Aspire 5739G
Суммарный объем системной памяти	4,00 ГБ ОЗУ
Тип системы	64-разрядная операционная система
Число ядер процессора	2

Рисунок 1 – Параметры компьютера.

Рассмотрим начальные скрипты алгоритмы для соединения с сервером и настройки параметров ускорения обработки данных сервером Word 2007, и оценим их работу:

Таблица 1

	Скрипты	Время (сек.)
1.	WordApp:=CreateOleObject('Word.Application');	0,521
2.	WordApp.Documents.Open(FileName);	0,677
3.	WordApp.Visible:=false;	0,051
4.	WordApp.Options.CheckSpellingAsYouType:=False;	0,015
5.	WordApp.Options.CheckGrammarAsYouType:=False;	0,002
6.	WordApp.Options.CheckGrammarWithSpelling:= False;	0,001
7.	col:=WordApp.ActiveDocument.Paragraphs.Count;	0,055
8.	WordApp.Options.ContextualSpeller := False;	0,016
9.	WordApp.Options.ShowReadabilityStatistics := False;	0,002
10.	WordApp.ActiveDocument.ShowGrammaticalErrors := False;	0,006
11.	WordApp.ActiveDocument.ShowSpellingErrors := False;	0,007

12.	VarIsEmpty(WordApp);	0,000
13.	WordApp.Visible:=true;	0,019
14.	WordApp.ActiveDocument.Close(SaveChanges,EmptyParam,EmptyParam);	0,431
15.	WordApp.Quit(SaveChanges,EmptyParam,EmptyParam);	0,007

Теперь рассмотрим скрипты непосредственной обработки коллекций и данных содержащихся в них, а также возможные способы их оптимизации.

Таблица 2

	Скрипты	Время (сек.)
1.	OleContainer.copy;	0,003
2.	OleContainer1.paste;	0,101
3.	OleContainer.copy; OleContainer1.paste; OleContainer2.paste; OleContainer3.paste; OleContainer4.paste; OleContainer5.paste;	0,455
4.	application.ProcessMessages;	0,005
5.	v:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.start;	0,025
6.	WordApp.ActiveDocument.Range(v,v+1).Select;	0,015
7.	s0:=WordApp.selection.text;	0,011
8.	v:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.start; WordApp.ActiveDocument.Range(v,v+1).Select; s0:=WordApp.selection.text;	0,051
9.	WordApp.selection.delete;	0,002
10.	WordApp.ActiveDocument.Range(v,v+1).Select;	0,018
11.	s0:=WordApp.selection.text;	0,002
12.	while numb(s0) do begin WordApp.selection.delete; WordApp.ActiveDocument.Range(v,v+1).Select; s0:=WordApp.selection.text; end;	0,019
13.	for k := 1 to 3 do begin v:=WordApp.ActiveDocument.Paragraphs.Item(2).Range.start; WordApp.ActiveDocument.Range(v,v+2).Select; s0:=WordApp.selection.text; if num(s0)=0 then begin	0,089

	<pre> vk:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.end; WordApp.ActiveDocument.Range(vk-1,vk).Select; WordApp.selection.delete; end; end; </pre>	
14.	WordApp.ActiveDocument.Paragraphs.Item(1).Range.select;	0,021
15.	WordApp1.Selection.EndKey(wdStory,EmptyParam);	0,015
16.	WordApp1.Selection.InsertAfter(inttostr(n)+' ');	0,002
17.	WordApp1.Selection.EndKey(wdStory,EmptyParam); WordApp1.selection.paste;	0,866
18.	WordApp.Selection.EndKey(wdStory,EmptyParam); WordApp.selection.paste;	0,061
19.	<pre> WordApp1.Selection.EndKey(wdStory,EmptyParam); case z of 1:WordApp1.Selection.InsertAfter('A '); 2:WordApp1.Selection.InsertAfter('B '); 3:WordApp1.Selection.InsertAfter('C '); 4:WordApp1.Selection.InsertAfter('D '); 5:WordApp1.Selection.InsertAfter('E '); end; </pre>	0,010

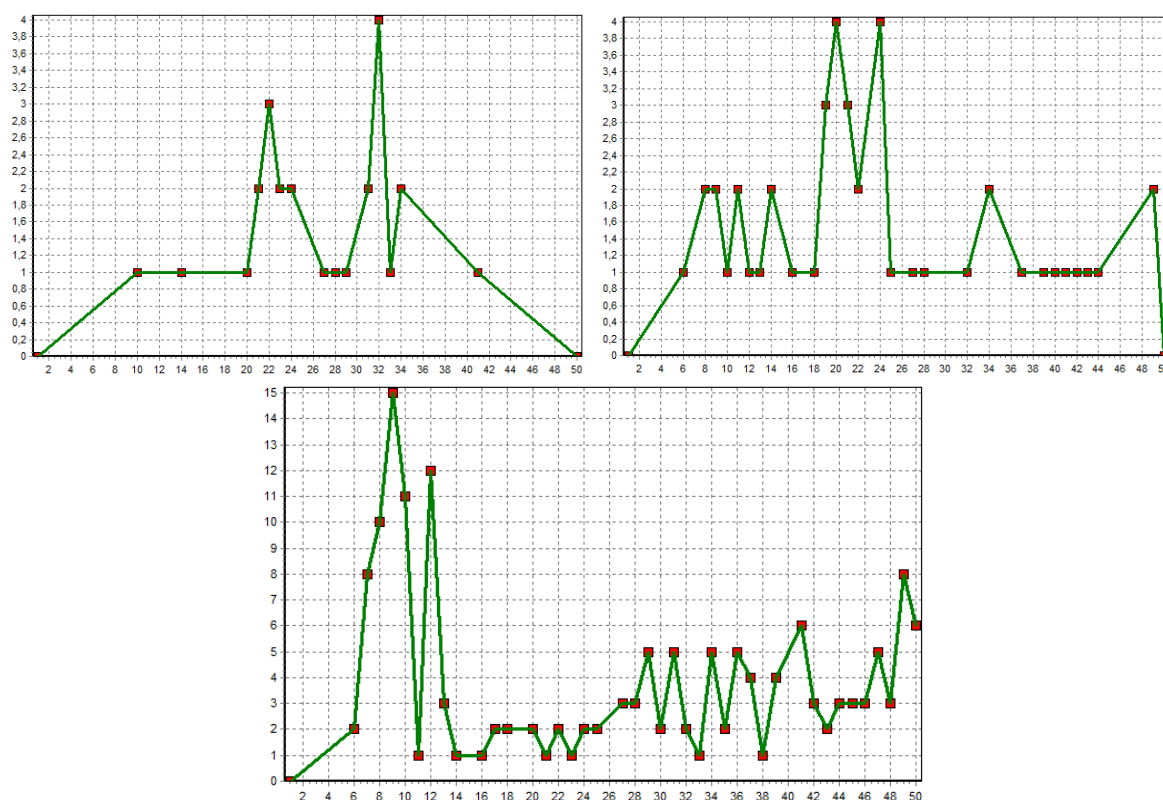
Интересные данные получим при выполнении скриптов внутри циклов, (табл. 2, пункт 19). Рассмотрим 5 вопросов, и 5 ответов, тогда получим временную матрицу:

Таблица 3

		Вопросы				
		1	2	3	4	5
Ответы	1	0,010	0,014	0,027	0,023	0,031
	2	0,022	0,021	0,032	0,032	0,034
	3	0,033	0,022	0,032	0,034	0,024
	4	0,020	0,023	0,022	0,028	0,024
	5	0,029	0,031	0,021	0,041	0,032

Аналогичным образом проходили апробацию ряд тестов различного объема, и с различным контентом. В нескольких случаях наблюдался сбой работы буфера обмена, а также прерывание программы в месте нахождения разрыва строки, в конце одного из ответов, хотя первым из алгоритмов анализа как раз и является отыскание разрывов (Shift+Enter) и преобразование их в обычный переход на следующую строку (#13#10).

Делалось предположение о принадлежности распределения времени для скриптов находящихся в цикле к одному из характерных и известных (нормальное, Пуассона, Лапласа и т.д.) но это не подтвердилось.



**Рисунок 1 – График функции распределения.
(5 вопросов, 10 вопросов, 70 вопросов)**

Выводы:

- Прослеживается уменьшение времени при выполнении повторяющихся операций, (табл.2, пункты 1-3).
- В отдельных случаях одинаковые скрипты выполняются с разным временем, скорее всего, это связано с объемом информации передаваемой через буфер обмена (табл.2, пункты 7 и 11).
- Время выполнения набора скриптов (табл.2, пункт 8) совпадает с суммарным временем выполнения отдельных скриптов (табл.2, пункты 5-7).
- Проверка условия (табл.2, пункт 12), и отказ от выполнения алгоритма, может занимать столько же системного времени как и сам вложенный исполняемый алгоритм.
- Расчеты в пределах одного сервера выполняются быстрее (табл.2, пункт 13), особенно это заметно при копировании информации через буфер обмена и вставке в параллельный сервер (табл.2, пункты 17 и 18), разница во времени составляет 0,805 секунды, что весьма значительно, и должно быть учтено при оптимизации.
- Построенная функция распределения не дает четкой зависимости времени от каких либо параметров, также не выявлена зависимость от размера объектов помещаемых в буфер.

Литература:

1. Шевцов А.Н., Шырынханова Д.Ж. Разработка алгоритмов и приложения компонентной модели для анализа и исправления ошибок экзаменационного теста. Theoretical & Applied Science. «Development of Applied Mathematics», ISPC, 30.05.2013, Taraz, Kazakhstan. - №5, 2013. -p.77-83.

2. ШЕВЦОВ А.Н., СМАЙЛОВА У.М., ШЫРЫНХАНОВА Д.Ж. Некоторые алгоритмы предварительной обработки теста. Theoretical & Applied Science. «Results & Perspectives», ISPC, 30.09.2013, Florence, Italy. - №9, 2013. -p.51-58.

Приложение 1

Текст программы

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Mask, Office_Tlb, word_tlb, ExtCtrls, StdCtrls, ComObj, ComCtrls, Menus,
  ShellAPI, ClipBrd, OleCtrls, Buttons;
type
  TForm1 = class(TForm)
    Button1: TButton;
    Edit1: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Edit2: TEdit;
    Edit3: TEdit;
    ProgressBar1: TProgressBar;
    ProgressBar2: TProgressBar;
    Edit4: TEdit;
    Memo2: TMemo;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    Edit5: TEdit;
    Edit6: TEdit;
    Label5: TLabel;
    Label6: TLabel;
    Button3: TButton;
    Label7: TLabel;
    Memo3: TMemo;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    OleContainer0: TOleContainer;
    Ed5: TEdit;
    OleContainer5: TOleContainer;
    OleContainer4: TOleContainer;
    Ed4: TEdit;
    Ed3: TEdit;
    OleContainer3: TOleContainer;
    OleContainer2: TOleContainer;
    OleContainer1: TOleContainer;
    Ed1: TEdit;
    Ed0: TEdit;
    Label9: TLabel;
    Label10: TLabel;
    CheckBox1: TCheckBox;
    Label8: TLabel;
```

```

Label4: TLabel;
Ed2: TEdit;
TabSheet2: TTabSheet;
Memo1: TMemo;
Label3: TLabel;
OleContainer: TOleContainer;
Memo4: TMemo;
CheckBox2: TCheckBox;
TabSheet3: TTabSheet;
Label11: TLabel;
CheckBox3: TCheckBox;
BitBtn1: TBitBtn;
Image1: TImage;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
DCOMConnection1: TDCOMConnection;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure N6Click(Sender: TObject);
procedure N7Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure E1;
  procedure FormDestroy(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);

private
  { Private declarations }
WordApp0, WordApp, WordApp1: OleVariant;
FData: OLEVariant;
  Procedure WordAppExcept(Sender: TObject; E: Exception);
procedure WMDropFiles (var Msg: TMessage); message wm_DropFiles;
function FindAndReplace(const FindText, ReplaceText: string): boolean;
function FindSelect(const FindText: string): boolean;
function FindSelectRed(const FindText: string): boolean;
protected
public
  { Public declarations }
end;

type
  TBlockHeader = packed record
    BlockID : Longint;
    BlockSize : Longint;
  end;
const

  bl_Sting = $0001;
  bl_Int = $0002;
  bl_DateTime = $0003;

Const
  ShablonFileName='тест.doc';
  NEWFileName='тест новый.doc';
  AB='ABCDE';
var
  Form1: TForm1;
  col, col1, col2: integer;
  E2b: boolean;

```

```

FileName,Enter:string;
tt:boolean;
d1: TDateTime;
a:array[1..500,0..5]of string;
implementation

{$R *.dfm}

Procedure TForm1.WordAppExcept(Sender:TObject; E:Exception);Begin Try WordApp.Quit(0); Except End;
End;

function numb(s:string):boolean;
var i:integer; b:boolean; s1,s2:string;
begin
s1:=copy(s,1,1); b:=false;
for I := 0 to 9 do if s1=inttostr(i) then b:=true;
numb:=b;
end;

function num(s:string):integer;
var i:integer; s1,s2:string;
begin
s1:=copy(s,1,1); s2:=copy(s,2,1);
i:=0;
if s1=' ' then s:=s2 else s:=s1;
if (s='A')or(s='a') then i:=1;
if (s='B')or(s='b') then i:=2;
if (s='C')or(s='c') then i:=3;
if (s='D')or(s='d') then i:=4;
if (s='E')or(s='e') then i:=5;
num:=i;
end;

function nump(s:string):boolean;
var i:integer; b:boolean; s1,s2:string;
begin
s1:=copy(s,1,1);
b:=false;
if s1=')' then b:=true;
if s1='.' then b:=true;
nump:=b;
end;

function TForm1.FindAndReplace(const FindText,ReplaceText:string):boolean;
const wdReplaceAll = 2;
begin
WordApp1.Selection.Find.MatchSoundsLike := False;
WordApp1.Selection.Find.MatchAllWordForms := False;
WordApp1.Selection.Find.MatchWholeWord := False;
WordApp1.Selection.Find.Format := False;
WordApp1.Selection.Find.Forward := True;
WordApp1.Selection.Find.ClearFormatting;
WordApp1.Selection.Find.Text:=FindText;
WordApp1.Selection.Find.Replacement.Text:=ReplaceText;
FindAndReplace:=WordApp1.Selection.Find.Execute(Replace:=wdReplaceAll);
end;

function TForm1.FindSelect(const FindText:string):boolean;
begin
WordApp1.Selection.Find.MatchSoundsLike := False;

```



```

WordApp1.Selection.Find.MatchAllWordForms := False;
WordApp1.Selection.Find.MatchWholeWord := False;
WordApp1.Selection.Find.Format := false;
WordApp1.Selection.Find.Forward := True;
WordApp1.Selection.Find.ClearFormatting;
WordApp1.Selection.Find.Wrap := 1;
WordApp1.Selection.Find.Text:=FindText;
while WordApp1.Selection.Find.Execute do
WordApp1.selection.Range.HighlightColorIndex:= wdYellow;//brush.color:=cllime;
end;

function TForm1.FindSelectRed(const FindText:string):boolean;
begin
WordApp1.Selection.Find.MatchSoundsLike := False;
WordApp1.Selection.Find.MatchAllWordForms := False;
WordApp1.Selection.Find.MatchWholeWord := False;
WordApp1.Selection.Find.Format := false;
WordApp1.Selection.Find.Forward := True;
WordApp1.Selection.Find.ClearFormatting;
WordApp1.Selection.Find.Wrap := 1;
WordApp1.Selection.Find.Text:=FindText;
while WordApp1.Selection.Find.Execute do
WordApp1.selection.Range.HighlightColorIndex:= wdRed;//brush.color:=cllime;
end;

procedure TForm1.E1;
Var
s:array[0..5]of String;
s0,st,sn,z1,z2:string;
I,j,p0,n,k,z,zzn,zz,zzn1: Integer;
v,vk: OleVariant;
b0,z0:boolean;
im1,im2:array[1..5]of tbitmap;
begin
for I := 1 to 5 do
begin
im1[i]:= TBitmap.Create;
im2[i]:= TBitmap.Create;
end;
n:=1;
col2:=WordApp.ActiveDocument.Paragraphs.Count;
// Анализ вопроса
while col2>7 do BEGIN
OleContainer.copy;
OleContainer1.paste;
OleContainer2.paste;
OleContainer3.paste;
OleContainer4.paste;
OleContainer5.paste;
application.ProcessMessages;

v:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.start;
WordApp.ActiveDocument.Range(v,v+1).Select;
s0:=WordApp.selection.text;
while numb(s0) do
begin
WordApp.selection.delete;

WordApp.ActiveDocument.Range(v,v+1).Select;
s0:=WordApp.selection.text;

```



```

v:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.start;
vk:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.end;
WordApp.ActiveDocument.Range(v,vk).Select;
WordApp.selection.copy;

```

```

if checkbox2.checked then

```

```

  case z of
  1:OleContainer1.paste;
  2:OleContainer2.paste;
  3:OleContainer3.paste;
  4:OleContainer4.paste;
  5:OleContainer5.paste;
  end;

```

```

WordApp1.Selection.EndKey(wdStory,EmptyParam);

```

```

case z of
1:WordApp1.Selection.InsertAfter('A ');
2:WordApp1.Selection.InsertAfter('B ');
3:WordApp1.Selection.InsertAfter('C ');
4:WordApp1.Selection.InsertAfter('D ');
5:WordApp1.Selection.InsertAfter('E ');
end;

```

```

WordApp1.Selection.EndKey(wdStory,EmptyParam);

```

```

if numb(s0)and (z<5) then

```

```

begin
WordApp1.Selection.InsertAfter(' //Нет ответа//'+#13#10);
WordApp.Selection.HomeKey(wdStory,EmptyParam);
WordApp.Selection.Insertbefore('A ')'+#13#10);
v:=WordApp.ActiveDocument.Paragraphs.Item(2).Range.start;
WordApp.ActiveDocument.Range(v,v).Select;
WordApp.Selection.Insertbefore('A ');

```

```

v:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.start;
vk:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.end;
WordApp.ActiveDocument.Range(v,vk).Select;
WordApp.selection.delete;

```

```

end

```

```

else

```

```

begin

```

```

v:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.start;
vk:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.end;
WordApp.ActiveDocument.Range(v,vk).Select;
WordApp1.selection.paste;

```

```

if z=1 then

```

```

begin

```

```

z1:= WordApp.selection.text;
zzn:=WordApp.Selection.InlineShapes.count;

```

```

if zzn>0 then

```

```

for zz := 1 to zzn do

```

```

begin

```

```

WordApp.Selection.Range.InlineShapes.Item(zz).Select;//CopyAsPicture;
WordApp.Selection.Copy;

```

```

Im1[zz].Assign(Clipboard);

```

```

end;

```

```

end;

```

```

    if z>1 then
    begin
    v:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.start;
    vk:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.end;
    WordApp.ActiveDocument.Range(v,vk).Select;
    z2:= WordApp.selection.text;
    zzn1:=WordApp.Selection.InlineShapes.count;

    if zzn>0 then
    for zz := 1 to zzn1 do
    begin
    WordApp.Selection.Range.InlineShapes.Item(zz).Select;
    //CopyAsPicture;
    // WordApp.InlineShapes.Item(zz).Range.CopyAsPicture;
    WordApp.Selection.Copy;
    Im2[zz].Assign(Clipboard);
    end;
    end;
    v:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.start;
    vk:=WordApp.ActiveDocument.Paragraphs.Item(1).Range.end;
    WordApp.ActiveDocument.Range(v,vk).Select;
    WordApp.selection.delete;
    end;

    z0:=false;
    if z1=z2 then
    if zzn=0 then
    z0:=true
    else
    begin
    z0:=true;
    // im1[zz].savetofile(inttostr(n)+inttostr(z)+'.bmp');
    for zz := 1 to zzn do
    if im1[zz]<>im2[zz] then z0:=false;
    end;

    if z0 then
    begin
    WordApp1.ActiveDocument.Range(
    WordApp1.ActiveDocument.Paragraphs.Item(WordApp1.ActiveDocument.Paragraphs.Count-
    1).Range.start,
    WordApp1.ActiveDocument.Paragraphs.Item(WordApp1.ActiveDocument.Paragraphs.Count-
    1).Range.end).Select;
    WordApp1.selection.Range.HighlightColorIndex:= wdPink;//brush.color:=cllime;
    WordApp1.Selection.endKey(wdStory,EmptyParam);
    end;

application.ProcessMessages;
end;

WordApp1.Selection.InsertAfter(#13#10);

// Конец вопроса

col2:=WordApp.ActiveDocument.Paragraphs.Count;
end;
WordApp1.Selection.HomeKey(wdStory,EmptyParam);
FindAndReplace(Chr(11),' ');
WordApp1.Selection.HomeKey(wdStory,EmptyParam);
FindAndReplace(#9,' '); // Tab

```

```

WordApp1.Selection.HomeKey(wdStory,EmptyParam);
FindAndReplace(' ');

WordApp1.Selection.HomeKey(wdStory,EmptyParam);
FindSelect('Her');
FindSelect('Да');
FindSelect('Незнаю');
FindSelect('Затрудняюсь ответить');
FindSelect('Все ответы верны');
FindSelect('Все ответы неверны');
FindSelect('Все ответы не верны');
FindSelect('Нет правильного ответа');
FindSelect('Все перечисленное');
FindSelect('Все перечисленные');
FindSelect('Всё перечисленное');
FindSelect('Всё перечисленные');

FindSelect('Ия');
FindSelect('Жок');
FindSelect('Білмеймін');
FindSelect('Дұрыс жауабы жок');
FindSelect('Барлық жауабы дұрыс');
FindSelect('Аталғандардың бәрі');
FindSelectRed('//Нет ответа//');

for I := 1 to col2 do
if (i-1) mod 7=0 then
begin
v:=WordApp1.ActiveDocument.Paragraphs.Item(i).Range.start;
vk:=WordApp1.ActiveDocument.Paragraphs.Item(i).Range.end;
WordApp1.ActiveDocument.Range(v,vk).Select;

WordApp1.selection.Range.HighlightColorIndex:= wdNoHighlight;//brush.color:=cllime;
end;

v:=WordApp1.ActiveDocument.Range.start;
vk:=WordApp1.ActiveDocument.Range.end;
WordApp1.ActiveDocument.Range(v,vk).Select;
WordApp1.Selection.Font.Size := 12;

WordApp1.Selection.ParagraphFormat.SpaceBeforeAuto:= False;
WordApp1.Selection.ParagraphFormat.SpaceAfterAuto:= False;
WordApp1.Selection.ParagraphFormat.FirstLineIndent:= 0;
WordApp1.Selection.HomeKey(wdStory,EmptyParam);
WordApp1.Visible:=true;

application.ProcessMessages;
for I := 1 to 5 do
begin
im1[i].free; im2[i].free;
end;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
Var
sss,sss1:String;
N,I,j: Integer;
v,vk: OleVariant;
d: TDateTime;

```

```

begin
  d:=Now;
  tt:=true;
  BitBtn1.enabled:=false;

  form1.Left:=10;
  form1.Top:=10;
  memo1.Clear;
  label4.Caption:=(‘max=’+inttostr(trunc(col/7)));
  col1:=1;
  col2:=WordApp.ActiveDocument.Paragraphs.Count;
  E1;
  application.ProcessMessages;

  col2:=WordApp.ActiveDocument.Paragraphs.Count;

  memo1.lines.add(‘Общее количество ошибок = ’+inttostr(memo1.lines.count));
  label3.caption:=‘Общее количество ошибок = ’+inttostr(memo1.lines.count-1);
  label10.caption:=‘ N = ’+inttostr(col2 div 7);
  memo1.lines.savetofile(‘Статистика ошибок теста – ’+extractfilename(filename) +’.txt’);
  WordApp.ScreenUpdating := true;
  WordApp1.ScreenUpdating := true;
  Label12.Caption:=FormatDateTime(‘hh:mm:ss’, Now()-d);
end;

procedure TForm1.Button1Click(Sender: TObject);
Var
  tt:String;
  N:Integer;
  m2,w,z, I,k,m: Integer;
  a,b,c,d,e: OleVariant;
  Un, x: Olevariant;
  label m3;
begin
  memo1.Clear;

  x:= wdPageBreak;
  Un:=wdParagraph;
  ProgressBar1.max:=strtoint(edit4.Text);
  ProgressBar2.max:=strtoint(edit1.Text);

  form1.Left:=10;
  form1.Top:=10;

  for w := 1 to strtoint(edit1.Text) do
  begin
  //запишем текст в шаблон
  a:=‘A’ ;b:=‘B’ ;c:=‘C’ ;d:=‘D’ ;e:=‘E’ ;
  randomize;
  memo2.Clear;
  WordApp1.Selection.EndKey(wdStory,EmptyParam);
  WordApp1.ActiveDocument.Range.InsertAfter(‘ Вариант № ’+inttostr(strtoint(edit2.Text)+w)+#13#10);
  //WordApp.font.style:=“;
  for I := 1 to strtoint(edit4.Text) do
  begin
  m3: n:=random(strtoint(edit3.Text));
  for m2 := 0 to memo2.Lines.Count - 1 do
  if n=strtoint(memo2.Lines.Strings[m2]) then goto m3;
  memo2.Lines.Add(inttostr(n));
  ProgressBar1.Position:=i;

```

```

ProgressBar2.Position:=w;

k:=n*7+1;
//WordApp.Selection.HomeKey(wdStory,EmptyParam);

WordApp.ActiveDocument.Paragraphs.Item(k).Range.copy;
WordApp1.Selection.EndKey(wdStory,EmptyParam);
WordApp1.ActiveDocument.Range.InsertAfter(inttostr(i)+' ');
WordApp1.Selection.EndKey(wdStory,EmptyParam);
WordApp1.selection.paste;
WordApp1.Selection.EndKey(wdStory,EmptyParam);
//WordApp1.ActiveDocument.Range.InsertAfter(#13#10);

//kop:=col;//WordTables1.Item(1).Rows.Count;

z:=1+random(5);

tt:=tt+copy(AB,z,1);

WordApp.ActiveDocument.Paragraphs.Item(k+z).Range.copy;
WordApp1.Selection.EndKey(wdStory,EmptyParam);
WordApp1.ActiveDocument.Range.InsertAfter('A '); WordApp1.Selection.EndKey(wdStory,EmptyParam);
WordApp1.selection.paste;

for m := 2 to 5 do
begin
if z=m then
begin
WordApp.ActiveDocument.Paragraphs.Item(k+1).Range.copy;
WordApp1.Selection.EndKey(wdStory,EmptyParam);
case m of
2: WordApp1.ActiveDocument.Range.InsertAfter('B ');
3: WordApp1.ActiveDocument.Range.InsertAfter('C ');
4: WordApp1.ActiveDocument.Range.InsertAfter('D ');
5: WordApp1.ActiveDocument.Range.InsertAfter('E ');
end;
WordApp1.Selection.EndKey(wdStory,EmptyParam);
WordApp1.selection.paste;

end
else begin
WordApp.ActiveDocument.Paragraphs.Item(k+m).Range.copy;
WordApp1.Selection.EndKey(wdStory,EmptyParam);
case m of
2: WordApp1.ActiveDocument.Range.InsertAfter('B ');
3: WordApp1.ActiveDocument.Range.InsertAfter('C ');
4: WordApp1.ActiveDocument.Range.InsertAfter('D ');
5: WordApp1.ActiveDocument.Range.InsertAfter('E ');
end;
WordApp1.Selection.EndKey(wdStory,EmptyParam);
WordApp1.selection.paste;
end;
end;

WordApp1.Selection.EndKey(wdStory,EmptyParam);
WordApp1.ActiveDocument.Range.InsertAfter(#13#10);
end;

memo1.lines.add(inttostr(strtoint(edit2.Text)+w)+' '+tt); tt:= "";

```

```
memo1.lines.add("");
WordApp1.Selection.EndKey(wdStory,EmptyParam);
WordApp1.Selection.InsertBreak(x);
end;
memo1.SelectAll;
memo1.CopyToClipboard;
WordApp1.selection.paste;
ProgressBar1.Position:=0;
ProgressBar2.Position:=0;
end;

procedure TForm1.FormCreate(Sender: TObject);
Var i:integer;
begin
for I := 1 to 5 do
case i of
1:Ed1.Text:='A';
2:Ed2.Text:='B';
3:Ed3.Text:='C';
4:Ed4.Text:='D';
5:Ed5.Text:='E';
end;

pagecontrol1.tabindex:=0;
tt:=false;
DragAcceptFiles(Handle,True);
end;

procedure TForm1.FormDestroy(Sender: TObject);
var
// для параметров
SaveChanges: olevariant;
begin
if not VarIsEmpty(WordApp1) then
WordApp1.Visible:=true;
// если Word не закрыт
if not VarIsEmpty(WordApp) then
begin
WordApp.Visible:=true;
//изменения не сохранять
SaveChanges:=false;
// то закрыть сначала документ
WordApp.ActiveDocument.Close(SaveChanges,EmptyParam,EmptyParam);
// а потом и ворд
WordApp.Quit(SaveChanges,EmptyParam,EmptyParam);
end;
end;

procedure TForm1.Button3Click(Sender: TObject);
Var
ss:String;
begin
ss:=label6.caption;
WordApp:=CreateOleObject('Word.Application');
//получаем путь к шаблону
FileName:=label5.caption;
//открываем документ
WordApp.Documents.Open(FileName);
WordApp.Visible:=false;
col:=WordApp.ActiveDocument.Paragraphs.Count;
```



```

edit5.Text:=inttostr(col);
label4.Caption:=('max='+inttostr(trunc(col/7)));
BitBtn1.enabled:=true;
WordApp1:=CreateOleObject('Word.Application');
//получаем путь к шаблону//открываем документ
WordApp1.Documents.Add;
WordApp1.Visible:=checkbox1.checked;//True;
WordApp1.Options.CheckSpellingAsYouType:=False;
WordApp1.Options.CheckGrammarAsYouType:=False;
// '3. Отключение проверки правописания.
WordApp.Options.CheckSpellingAsYouType := False ;
WordApp.Options.CheckGrammarAsYouType:= False;
WordApp.Options.CheckGrammarWithSpelling:= False;
WordApp.Options.ContextualSpeller := False ;
WordApp.Options.ShowReadabilityStatistics := False;
WordApp.ActiveDocument.ShowGrammaticalErrors := False;
WordApp.ActiveDocument.ShowSpellingErrors := False ;
// '3. Отключение проверки правописания.
WordApp1.Options.CheckGrammarAsYouType:= False;
WordApp1.Options.CheckGrammarWithSpelling:= False;
WordApp1.Options.ContextualSpeller := False ;
WordApp1.Options.CheckSpellingAsYouType := False ;
WordApp1.Options.ShowReadabilityStatistics := False;
WordApp1.ActiveDocument.ShowGrammaticalErrors := False;
WordApp1.ActiveDocument.ShowSpellingErrors := False ;

if checkbox3.checked then BitBtn1.click;
WordApp.Visible:=true;
WordApp1.Visible:=true;
end;

procedure TForm1.WMDropFiles(var Msg: TMessage);
Var Filename: array[0..256] of char;
begin
DragQueryFile(THandle(Msg.WParam),0,Filename,SizeOf(Filename));
// Теперь в переменной Filename будет находиться путь
// к перетаскиваемому файлу. Далее вы можете выполнять с этим файлом, зная
// его путь, все что угодно.
label5.caption:=FileName;
label6.caption:=extractfilename(label5.caption);
label11.caption:=copy(extractfilename(label5.caption),1,length(extractfilename(label5.caption))-4);
Button3.Click;//BitBtn1.Click;
//Сообщаем об окончании претаскивания
DragFinish(THandle(Msg.WParam));
end;

procedure TForm1.N4Click(Sender: TObject);
begin
application.Terminate;
end;

procedure TForm1.N6Click(Sender: TObject);
begin
showmessage('Программа предназначена для предварительного анализа и автоматического
исправления ошибок экзаменационных тестов...');
end;

end.

```