

,

,

. . .

i

i

i

DELPHI

“

”

2

15 09.04.2002 .

2002

004.43

2002 .

: . , . . , . . , . .

— „

Object Pascal

Delphi. ()

” “ ”.

,

,

“

,

.

,

“I ”;

,

,

Delphi.

— . . .

7

1

Object Pascal.

2

2.1.

8; c, 1 (8) ; 256 (0,1 ,..., 255).

7.1.

7.1 –

♥	00000011	3
♦	00000100	4
	11111110	254
	11111111	255

ANSI- '0' '9' ANSI - 1, 2.4.1

```

Object Pascal
: '+', 'a', 'R', '5', '(
),
#.# : #13 ("Enter"), #27 ("Esc").
- char (
ansichar). :
Var c1,c2 : char;
name : array[1..20] of char;
Cmass : array[1..3,1..3] of char;
1 2 -
name -
Cmass -
'Z' 'X' 'Y'
'1' '3' '5'
'/' '+' '?'
:
1:='x';
(a a , i a i i i ),
Edit, Memo, ListBox,
StringGrid , name -
Edit :
k:=Edit1.GetTextLen;
for i:=1 to k do
name[i]:=Edit1.text[i];
k
Edit. GetTextLen
a a . a i ,
a " i " ( a i i i ),
a a a a p pa .
p " " i , a a " pa "
, p . i
" i " a ( k<20), a p a pa p
for j := k+1 to 20 do name[j] := ' ';
( )
Edit, Memo, ListBox, StringGrid .

```

2.2

Char

1)

Function Chr(**n**:integer):char;**n**

:

Chr(97)='a'; Chr(148)='A'; Chr(32)=' '().

2)

Function Ord(**c**:char):integer;**c**

:

Ord(' ') = 97; Ord('A') = 148; Ord('♦') = 4.

3)

ifunction UpCase(**c** : char) : char ;

p a i a i i p i . a p a , i a
 pa pi

c1 := 'b';

c2 := UpCase(c1);

a i c2 'B'. C i a ' a , i i
 a i , pi a a i p, p a a i a i p
 p i. a a UpCase, a i , i p ' ', p a
 a a a i pa ' ' (a ' ').

2.3

: <, >, =, <=, >=,

<>.

:

if c1>c2 then c1:=c2;

1 > 2 (True),
 2.

1 -
 -
 . -
 -

,

:

('0'<=c) and (c<='9').

a ia p a 15 a i i a a a
 a a i a a i p a i. , i p a
 , a a a a i i pa a i a ANSI- p .
 :

procedure TForm.Button1Click(Sender:Tobject);

Var a : array[1..15] of char; max : char; i, n : integer;

begin

```

  n := Memo1.Lines.Count; {
  if n=0 then begin ShowMessage(' a
  if n>15 then n:=15; {
// Memo
  for i:=1 to n do
  a[i]:=Memo1.lines[i][1];
//
  max := Chr(32); {
  for i:= 1 to n do
  if ( a[i]>='a')and( a[i]<='z') {
    and ( a[i] > max ) {
    then maxL := a[i];
//
  Edit1.text:=' a a a i . i pa a : '+ max;
end;
  a a , , p i a a -
  a a i p , .

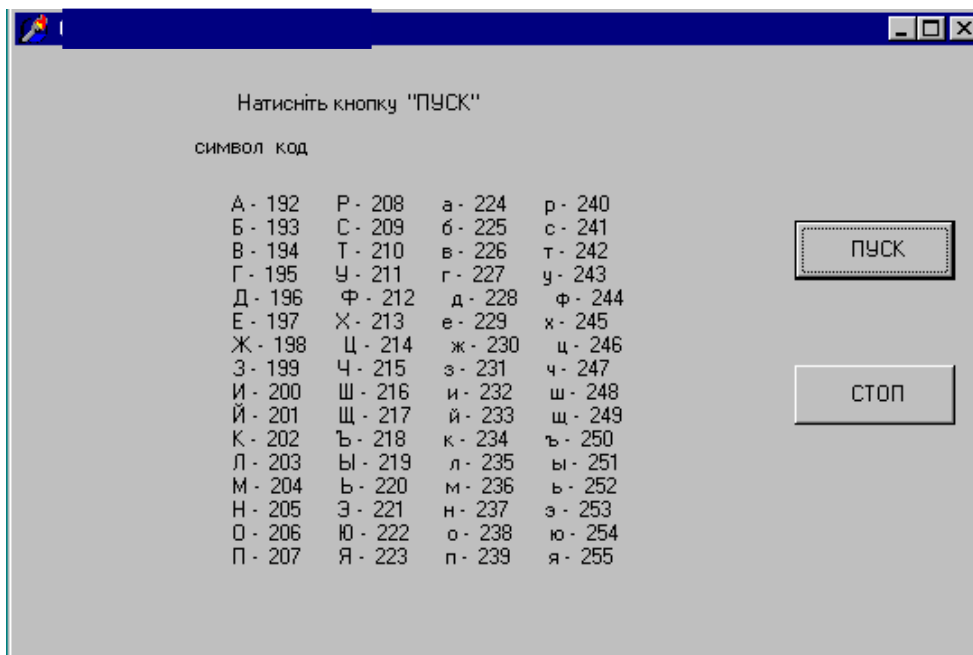
```

3

3.1 1

• (192). Label

. 7.1.



```

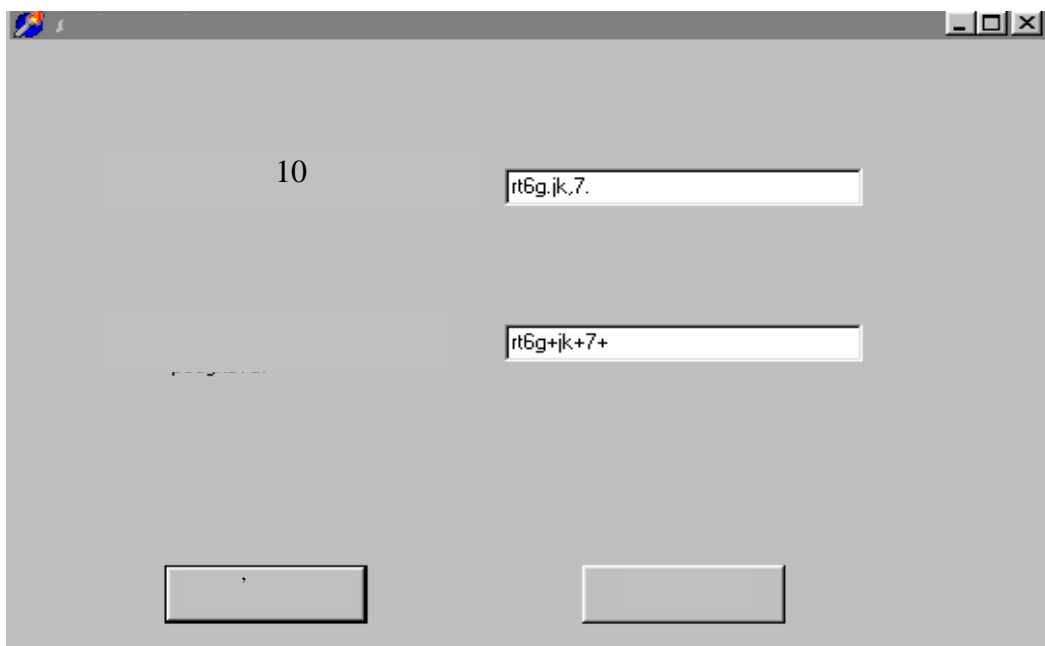
procedure TForm1.Button1Click(Sender: TObject);
  Var ki:byte; {
    i,k:integer; s:string;
begin
  for i:=0 to 15 do { 16 }
  begin
    ki:=i+192;
    for k:=1 to 4 do { 4 }
    begin
      s:=s+Chr(ki)+' - '+IntToStr(ki)+' ';
      ki:=ki+16;
    end;
    s:=s+#13; {
  end;
  label2.caption:=s;
end;

```

3.2 2

10-
 ‘;’ () ‘+’.
 Edit.

.7.2.



```

:

Unit unit1;
.....
implementation

{$R *.DFM}

Type tchararr=array[1..10] of char;

//
procedure zamina (k:integer; s1: tchararr; var s2: tchararr);
var i:integer;
begin
    for i:=1 to k do
        if (s1[i]='.') or (s1[i]=' ') then s2[i]:='+' else s2[i]:=s1[i];
end;

//
procedure TForm1.Button1Click(Sender: TObject);
var str1,str2:tchararr; i,k:integer;
begin
{
    k:=Edit1.GetTextLen;
    if k=0 then begin ShowMessage( ' a '); exit; end;
    if k>10 then n:=10;
    for i:= 1 to k do
        str1[i]:=edit1.text[i];
{
    zamina(k,str1,str2);
{
    Edit2.Text:='';
    for i:=1 to k do
        edit2.text:= edit2.text+str2[i];
end;
.....
end.

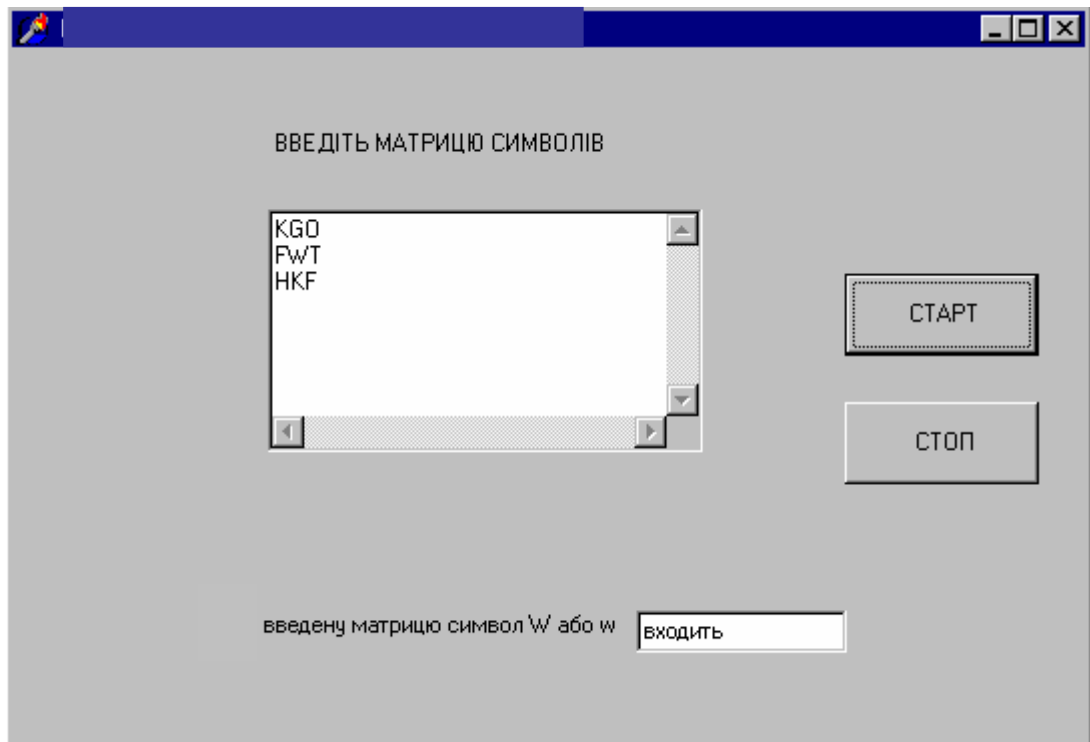
```

3.3 3

3 3 , W w. -

Memo, - Edit.

7.3.



7.3 –

3

```

.....
implementation
{$R *.DFM}

Type matr=array[1..3,1..3] of char;

//                                     W      w
procedure Wwsimv(n,m:integer;mas:matr;var vidp:boolean);
Var i,j:integer;
begin
  vidp:=false;
  for i:=1 to n do
  for j:=1 to m do
  if (mas[i,j]='W') or (mas[i,j]='w') then
  begin
    vidp:=true; exit
  end;
end;

//                                     "      "
procedure TForm1.Button1Click(Sender: TObject);
var a:matr;i,j:integer;
    vidpov:boolean;
    s:string;

```

```

begin
{
Memo}
for i:=1 to 3 do
for j:=1 to 3 do
a[i,j]:= memo1.lines[i-1] [j];
{
Wwsimv}
Wwsimv(3,3,a,vidpov);
{
Edit}
if vidpov=true then edit1.text:='
else edit1.text:='
';
end;

```

4

```

1 Object Pascal?
2 ?
3 , 256 .
4 256- 255?
5
32.
6 00000101.
7 , '%'
8 , Edit -
9 ANSI- 13. '&',
10 Edit.
11 a i pa p, a a i i i S a a
" i ".
12 a pa p pa :
...
var p : array[1..10] of char; ...
begin
Edit1.text:=p;
...
13 a i pa p a pa i? i i p .
a ia p a p p ip 3*4. i , a p a
" a " i, a p a a a ENTER
i i p a.
14 a i pa p a pa p , a p a a
pa i a p p ip 4*5. i , a p a a a
pa i " a " , a i 4- p i 5 i

```

```

15   a   pa   p   pa   :
      ... var i, j : integer; C_arr : array[1..2,1..3] of char;
      begin
      for i:=1 to 2 do
      for j:=1 to 3 do C_arr[i,j] := CHR(i+j) ;

```

5

1	Pascal	(Object	-
2	DELPHI.			
3		,		
1	ANSI-	,	m	
2		,	9-	
3		,	12-	-
4		,	10-	-
5		,	9-	
6		,	14-	-
7		,	11-	-
8		,	19-	-
9	ANSI -	,	17-	-
10		,	18-	-
11		,	12-	-

12	,			11-	-
13	,			14-	-
14	,			“+”	11-
15	,			‘ # ‘	2 3
16	,		R r.		
5-17	,				
17	,			13-	-
18	,			15-	-
19	,				6-
20	,				3 3
21	,				-
(32 255) 10-					-
ANSI-					
22	,			10-	-
23	,			‘_‘	9
24	,				3 4
25	,				3 2
26	,				3 3
27	,				3 4
28	,				3 3
29	,		ANSI-		
10-					
30	,				3 4

8

1

) ' (-
Object Pascal.

2

2.1.

- () - .
:
, ,
, ' string. -
,
,
,
name1:='Pascal', name2:=name1[4] 'c'.
,
0 () ,
.

```
Var name:string[15];
...
Begin
...
name:='Object Pascal';
...
name:='Delphi';
...
end;
```

15.

(name-1)

'Object'

Pas al' (name-2) - 'Delphi' - 6
). , , name 16
 .8.1.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
name-1	13	O	b	j	e	c	t		P	a	s	c	a	l		
name-2	6	D	e	l	p	h	i									

8.1 -

```

0
name ( - 13, - 6).
)
name " "
:
name:="";
255.
:
Var s:string;
.
' ' 2 255
shortstring string[255],
Delphi) : longstring, widestring, ansi-string, widechar.
    
```

2.2 string-

1) () char
 string, - string- s1,s2 s3 string, t char i
 ,
 t:='-';
 s1:=' ';
 s2:=' - ';
 s3:=s1+c+s2;
 s3 ' - - '.

2) `(...)`. -
s1 `. C` *s1*
`s2,`
`s2` - ANSI-
`'1' '<>'1$'` True, `'1' '='1$'` -
 False.

2.3 string-

1) **Copy** –

```
Function Copy(s:string; i:integer; n:integer):string;
s – ;
– ;
n – ( ) .
```

```
Var str1,str2:string;
begin
str1:='Object Pascal';
str2:=Copy(str1,8,6);
...
str2 str1, 8- -
6. , Str2='Pascal'.
```

2) **Pos** –

:

```
Function POS(p:string;s:string):byte;
```

```
s , – byte – s; -
0 255. , POS('Pascal','Object Pascal') -
```

8;

3) **Length** –

:

```
Function Length(s:string):byte;
```

string-

:

```
Var str:string[30];dlina:integer;
begin
str:='';
dlina:=Length(str); // dlina 11
...

```

3) **StrToInt(s)** –4) **IntToStr(n)** –5) **StrToFloat(s)** –6) **FloatToStr(x)** –

2.4

1) Delete –

```

Procedure Delete(var s:string; k:integer; n:integer);
    s      n      ,      ,
    s      k.      s
n      .
:
    ...
s:= 'Object Pascal';
Delete(s,1,6); //      'Pascal'

```

2) Insert –

```

Procedure Insert(p:string; var s:string; k:integer);
    string-      s,      -
k      s.
:
    ...
str1:='Object';
str2:='Pascal';
str1:=str1+'';
Insert(str1,str2,1); //      str2='Object Pascal'

```

3) Val –

```

Procedure Val(s:string; var x:<      >; var ier:integer);
    s –      , –      real integer,
ier –      ,      ier
.
:
    ...
var s1,s2: string; f:integer; h:real; ier:integer;
    ...
begin s1:='125'; s2:='-1.253E2';
Val(s1,f,ier); //      f      125
Val(s2,h,ier); //      h      -125,3

```

4) Str –

```

Procedure Str (x:<      >; var s:string);
    s.      ,
Str(12,s) string-      s      '12'.

```

3

3.1 1

string- 20-

.8.2.

8.2 –

1

```

:
.....
implementation
{$R *.DFM}

Type tstr=string[20];

//
procedure Spsimv(s1,s2:tstr;var s:tstr);
  var i,dlina,po,ps:integer;
begin
  s:='';
  dlina:=length(s1);
  for i:=1 to dlina do
    begin
      ps:=pos(s1[i],s2);
      if ps<>0 then
        begin
          po:=pos(s1[i],s);
          if po=0 then s:=s+s1[i];
        end;
      end;
    end;
end;

```

```

//
procedure TForm1.Button1Click(Sender: TObject);
var s1,s2,s3,s12,s13,s23,s123:tstr;
begin
//
s1:=edit1.text;
s2:=edit2.text;
s3:=edit3.text;
//
Spsimv(s1,s2,s12); edit4.text:=s12;
Spsimv(s1,s3,s13); edit5.text:=s13;
Spsimv(s2,s3,s23); edit6.text:=s23;
//
Spsimv(s12,s23,s123); edit7.text:=s123;
end;
.....
end.

```

3.2 2

).

Memo (

.8.3.

8.3

2

```

:

unit Unit1;
.....
implementation
{$R *.DFM}

    Type s60=string[60];

//          -
function del_n(s1:s60):s60;
begin
    while (pos(' ',s1)=1)and (length(s1)>0) do
        delete(s1,1,1);
    del_n:=s1;
end;

//          -          "          "
function slovo(s:s60):s60;
var p:integer;
begin
    p:=pos(' ',s);
    if p=0 then slovo:= s
        else slovo:= copy(s,1,p-1) ;
end;

//          -          "          "
procedure dol_gr(sdol:s60; koef:real; var sgr:s60);
var dol,gr:real; ier:integer;
begin
    delete(sdol,1,1); //          ' $'
    val(sdol,dol,ier); //
    gr:=dol*koef; //
    str(gr:10:2,sgr); //
    sgr:=sgr+' .';
end;

//          "          "
procedure TForm1.Button1Click(Sender: TObject);
var sv,sr,sg:s60; k,dd:integer;
begin sv:=edit1.text; //
    k:=strtofloat(Edit2.text); //
    sv:=del_n(sv); //

```

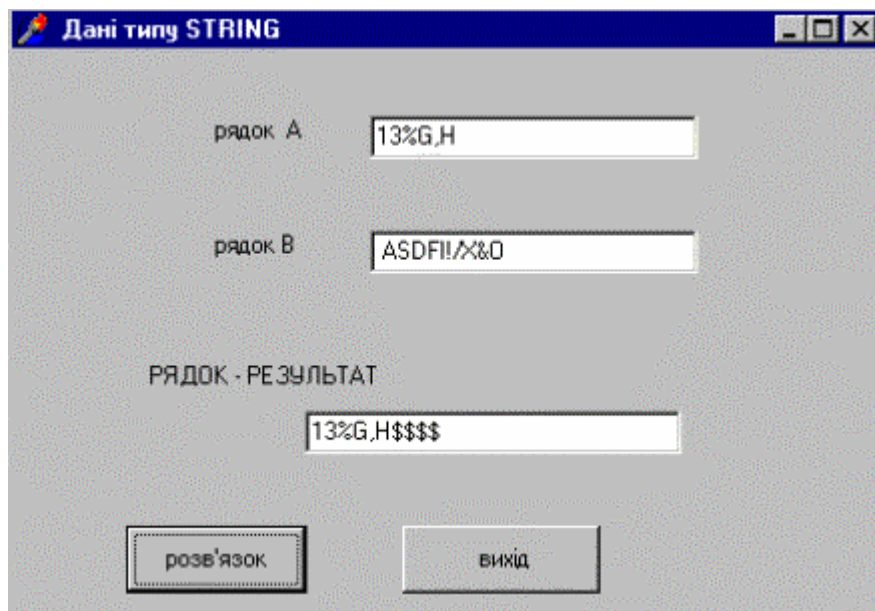
```

while length(sv)>0 do
begin
  sr:=slovo(sv); //
  dol_gr(sr,k,sg); //
  memo1.lines.add(sg); //
  delete(sv,1,length(sr)); //
  sv:=del_n(sv); //
end;
end;
end.

```

3.3 3

string-
 ‘#’
 ‘\$’
 .8.4.



8.4 –

3

```

:
.....
implementation
{$R *.DFM}

//
procedure TForm1.Button1Click(Sender: TObject);
  Var a,b:string[10];
  dla,dlb,i:integer;

```

```

begin
//
a:=edit1.text;
b:=edit2.text;
//
dla:=length(a);
dlb:=length(b);
if dla>dlb then
//                                     ‘#’
  for i:=dlb+1 to dla do
    begin
      b:=b+'#';
      edit3.text:=b;
    end
else if dla<dlb then
//                                     ‘$’
  for i:=dla+1 to dlb do
    begin
      a:=a+'$';
      edit3.text:=a;end;
end;

end.

```

4

1		?		?		
2			,	string[5]?		
3			,			8
4	.		,			S2
5	S1,	4-		“ ”		-
6	,		,	“ ”		k -
7	?	.	,			D -
8			,	S?		.
9	,		string-	,		? -
	.		,			
	,			S,		?
	.					

10

-

?

11

Length('')?

12

,

?

5

1

Object

Pascal

-

Delphi.

2

3

1

,

10-

2

,

12-

, ,

'+',

3

,

13-

-

4

,

10-

5

,

16-

6

(

,

).

10-

7

,

12-

'+'.

8

,

14-

, ,

,

9

,

10-

,

,

,

,

10

,

9-

,

,

,

“

”

-

11	,	12-
12	;	10-
13	,	8-
14	,	10-
15	,	11- '#'
16	,	10-
17	'45', '5', '4'.	12-
18	,	16-
19	,	11-
20	,	10-
21	,	18-
22	,	10-
23	,	8-
'fdg#3gk'	'fdg#3gk'.	

24 , 9- -

25 , 8- -

(: 5.5 , 7.5 , 2.5 , ..., 9.5).

26 , 6- -

(5-) -

ANSI- .

27 , 5- -

(6-), -

'\$' (, \$125) .

(, 625) ,

28 , 8- -

/ .

29 , 13- -

30 , 10- -

(12.5)

; ,

(, 1.25 1). , .

```
Procedure, '
( ' )
.
, , , .
, ' begin end, end ' .
( ) :
' ( );
, , - .
, ' ; " ; " (
).
:

Procedure ex1 ( n1, n2, ..., nN : 1;
              m1, m2, ..., mM : 2;
              k1, k2, ..., kK : 3 );

, .
( ).
:
• - ;
• - ( Var);
• - (
Procedure);
• - ( Function).
.
, ,
- ,
.
, ,
- ,
,
-
,
.
Var-
```

```

:
Procedure ex2 ( a, b, c : real; var x, y : real );
Var z : real;
begin
  z := a+b+c;
  x := sqr ( z );
  y := sqrt ( z );
end;

ex2 ( 1. 7, 5. 5, 1. 8, t, q );

: = 1.7, b = 5.5, c = 1.8.
x y,
t q.
: t = 81. 0, q = 3.0.
ex2
:

f := 5. 5; k := 1. 1;
ex2 ( 1. 7, f, k+0. 7, i, w );

: a = 1. 7, b = 5. 5, c = 1. 8.

( ):
.....
Const n = 11;
Type mas = array [ 1.. n ] of real;
.....
Procedure ex3 ( k : integer; var x : mas );
{
..... }
end;
.....
{ }
.....
Var a : mas;
.....
ex3 ( n, a ); //
.....

```

```

= 11,
      k
      -
      .
      .
      ,
      -
      (
      ).
      .
      ,
      .

```

2.1.2

```

      -
      :
Function ' F (
      {
      }
      ) : -
      ;
.....
begin
      {
      }
.....
      ' F :=
      i
      ;
end;
      ,
      ( ' F )
      (
      ,
      ).
      -
      ,
      i
      ,
      -
      ,
      ,
      ,
      ,
      ,
      ,
      ,
      ,

```

$$\text{sign}(x) = \begin{cases} -1, & x < 0, \\ 0, & x = 0, \\ 1, & x > 0 \end{cases}$$

```

i
-
:
Function fsign(x : real):integer;
begin
  if x < 0 then fsign := -1
  else if x = 0 then fsign := 0
  else fsign := 1;
end;

```

i

-

i

```

Function fact(n : integer) : real;
var k : integer;  p : real;
begin  p := 1;
      for k := 1 to n do
        p := p*k;
      fact := p;
end;

```

```

      fact          real,
n,                7,          i          32767

```

```

i          -          (          -
,          ),

```

```

          ,          i          -          .          -

```

```

          (          Unit)

```

```

{$F+}.

```

```

Z = xy*sign ( x )

```

```

, fsign,

```

```

.....
var x,y,z : real;
                                begin
.....
      x := -2;  y := -2.557;
      z :=x*y*fsign(x);
.....
end;

```

$$a = \frac{m!}{(n-m)!}$$

- fact

```

.....
var m, n : integer;  a : real;
.....
begin
  m := 5; n := 2;
  a := fact (m) /fact(m - n);
.....
end;

```

2.2

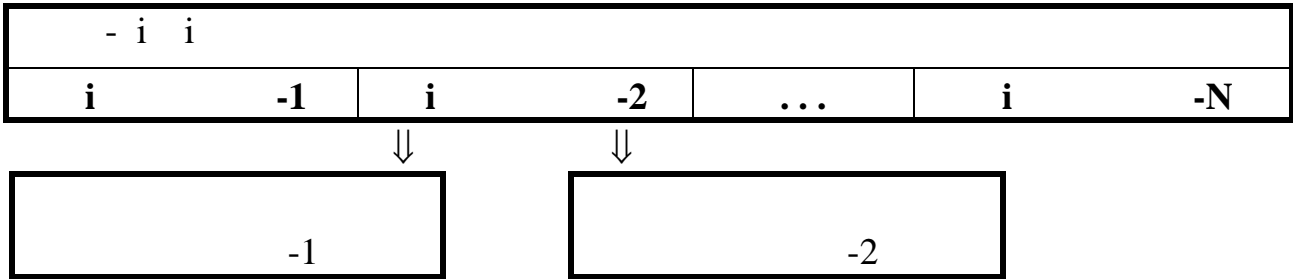
2.2.1

9.1

```

i i , i i i i -
i , i
i i - " i " .
9.1 , ( i -2) -
i i , i i
i i , i , -
i " i " .

```



9.1 - ,

```

i i i , i -
Delphi, i i i ( i i ), i -
. i i .
,
Delphi, uses c -
:
uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, ExtCtrls;
i i i i ,
HELP- Delphi.
i i exe- i i i i
i i i .

```

2.2.2

```

( ), -
i :
uses i ' i i -1, i ' -2, ..., i ' -N;

```

```

implementation.
    Pslau,
    Pvm ( c:\students\library\pvm, pvm.inf
pslau.par):
    Procedure pslau( n : integer; a : matr; var x : vect );
    ( n - i, a - i i i, x - -
      i i Pvm i i i :
    const maxn = 20 ;
    type matr : array[ 1..maxn, 1..maxn+1 ] of real ;
         vect : array[ 1..maxn ] of real ;
    i, Pslau " i " .,
implementation
uses Pvm ;
.....
procedure TForm1.Button1Click(Sender: TObject);
var m : matr; v : vect;
.....
begin
.....
Pslau( 5, m, v );
...
end;
    Pslau - i i m v -
    i matr vect i .
    i i i
Pvm:
// PSlau
uses pvm ;

```

```
const maxn = 20 ;
type matr : array[ 1..maxn, 1..maxn+1 ] of real ;
      vect : array[ 1..maxn ] of real ;
var a : matr; x : vect;
... begin

Pslau( 5, a , x );
...
```

Pascal, $\int_0^1 \frac{1+x}{\sqrt{1-k^2 \sin^2 x}}$

```
( k = 0.75471), Pvm Pm Psimps Psimps :
```

procedure Psimps(a,b:real; f : Ftype; eps : real; var : real; var ier : integer);

```
a, b, eps, ier, Ftype, Psimps ( ier = 0), Ftype -
```

Type Ftype = function(x : real) : real;

:

```
{$F+}
Unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, ExtCtrls;
.....
implementation
{$R *.dfm}
```

Uses PVM ;


```

Function fun ( x : real ) : real;
  Const k = 0.75471;
  Begin
    fun:=1/sqrt(1-sqr(k*sin(x)))
  end;
  //
procedure TForm1.Button1Click(Sender: TObject);
  Var Eps, R: real; IER : integer;

      Begin
.....

      Psimps( 0, 1, fun, eps, r, ier );
.....
End;

```

```

      i
      i ( ) Object Pascal. i
      i i i i ' i fun, i
      i i . i i , i .
Psimps. i
      i i . - i i i '
      i i i { $F+ } .1) , i -

```

2.2.3

```

Pascal- i i :
unit <i ' i i >;
interface
< i i , i i i
( i i i i ),
i i , i i i >
implementation
< i i i i i interface
i i i >

```

¹⁾ Ic i i , { \$... } (*\$...*)
. i i i , i , i i i .
, i ()
).

```

begin
  <      (      ), i      i
                -      i i      i      >
end.
                                i i i      :
unit <i ' i i >.      i      -      -
  i '      -      i      i i      i      ,
  i      i      .
                                i      i i      i (      -
.pas),i '      i      i      ,      i unit.

  i i      i      interface      implementation
  i i      (      interface      "      " -
    i i      "      i i"      "      " i
i i      ).      i      implementation      begin
  i      i i (implementation      " i      ").      i,
  i      begin      end -      i i i i i i.
  i (      !      i i      i      -i      -
    begin,      end. (      )      '      .
      i,      i i      '      .
      i i      i      ,
      i      .      ,      i      -
i,      i i      i
.      Pascal "      "      (      -
i ) Randomize.      i      ,      c      i
      i i      ,      i      i i
i i i i i i i :
  unit ...
  interface
  .....
  implementation
  .....
  { i      i      implementation }
  begin
  randomize
      end.
      i      i i      i i      i i.
      :      i i      i i      (      i i      i i
      i      i      i i      ) ?      ,
  i i      i      -      i i      -
      i      ,      i i      i      " i      " -
      i i i      .      i i      " i      " i
implementation      i      , "      "

```

```

i      i i      .      ,      i      i      i      i      i      i
i      i i      .      i i      ,      i      i      i      i      ,
i      i      , i      ,      i      i      .
i      i i      ,      ,      i
uses    i      i      i i      .      i      interface
implementation –      i i      ,      i i
"      i      i" i i      i      i.
.
i      i i      i      i,
i i implementation.      (
)      i      i      i
i i      .

```

2.2.4

```

1
  File/New.
2      Unit
  ,
  :
      Unit Unit2;
      interface
      implementation
      end;
3
4      ,
      .
      ' Unit2      ' .

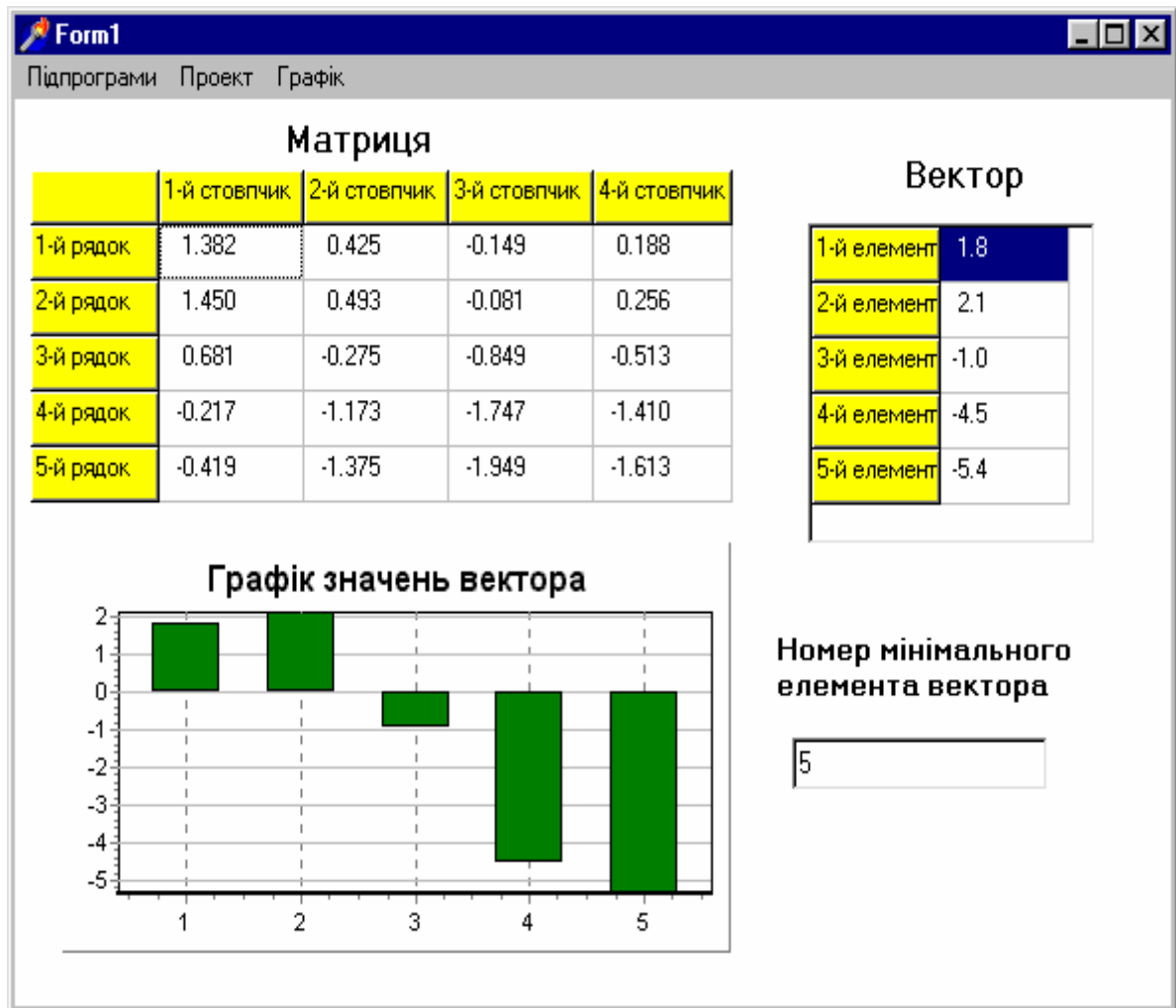
```

3

```

1
      :
      ai,j = sin i + cos j,      i = 1,2, ...,4; j = 1,2, ...,5.
2
3      ,
      -
      .

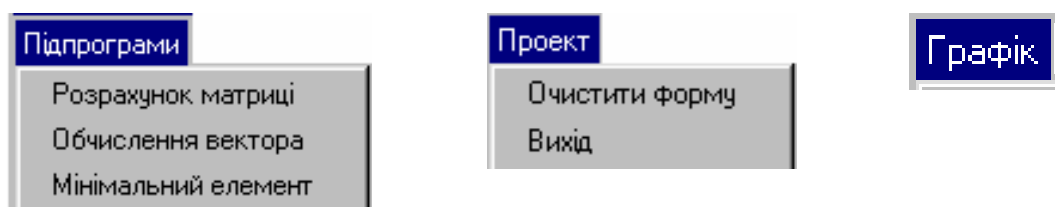
```



9.2 –

MainMenu.

.9.3.



9.3 –

unit Unit1;**interface**

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, Grids, StdCtrls, Menus, TeEngine, TeeFunci, Series, TeeProcs, Chart, **bibl**;

type

TForm1 = class(TForm)

Label1: TLabel; Label2: TLabel; Label3: TLabel;

StringGrid2: TStringGrid; StringGrid1: TStringGrid;

Edit1: TEdit;

MainMenu1: TMainMenu;

N1: TMenuItem;

.....

N8: TMenuItem;

procedure FormCreate(Sender: TObject);

procedure N2Click(Sender: TObject); procedure N3Click(Sender: TObject);

procedure N4Click(Sender: TObject); procedure N6Click(Sender: TObject);

procedure N7Click(Sender: TObject); procedure N8Click(Sender: TObject);

Chart1: TChart;

TeeFunction1: TAddTeeFunction;

Series1: TBarSeries;

.....

implementation

{ \$R *.DFM }

const n=5; m=4;

var a:matr; :vect; { }

//

procedure TForm1.FormCreate(Sender: TObject);

begin

n3.visible:=false;

n4.visible:=false;

end;

```

//          "          /          "
procedure TForm1.N2Click(Sender: TObject);
  var i,j:integer; s:string;
begin
  {          }
    elem(n,m,a);
  {          }
  for i:=1 to n do
    StringGrid1.Cells[0,i]:=IntToStr(i)+'-';
  for j:=1 to m do
    StringGrid1.Cells[j,0]:=IntToStr(j)+'-';
  {          }
  for i:=1 to n do
    for j:=1 to m do
      begin
        str(a[i,j]:8:3,s);
        StringGrid1.Cells[j,i]:=s;
      end;
  {          }
  n3.visible:=true;
  n4.visible:=false;
end;

//          "          /          "
procedure TForm1.N3Click(Sender: TObject);
  var i,j:integer; s:string;
begin
  {          }
    sum(n,m,a, );
  {          StringGrid}
  for i:=1 to n do
    begin
      StringGrid2.Cells[0,i-1]:=IntToStr(i)+'-';
      str( [i]:5:1,s); StringGrid2.Cells[1,i-1]:=s;
    end;

```

. . .

```

{
    n4.visible:=true;
end;

//
procedure TForm1.N4Click(Sender: TObject);
var i:integer; nom:integer;
s:string;
begin
{
    nom:=nom_min(x,n);
{
    edit1.text:=IntToStr(nom);
end;

//
procedure TForm1.N6Click(Sender: TObject);
var i,j:integer;
begin
    for i:=1 to n do
        for j:=1 to m do
            StringGrid1.Cells[j,i]:=”;
            for i:=1 to n do
StringGrid2.Cells[1,i-1]:=”;
                edit1.text:=”;
{
n3.visible:=false;
n4.visible:=false;
end;

//
procedure TForm1.N7Click(Sender: TObject);
begin
close
end;

```

```
//
//
procedure TForm1.N8Click(Sender: TObject);
var i:integer; gr:array [1..n] of TPoint;
begin
{
}
  for i:=1 to n do
    Series1.Addxy(i,x[i],”,cITeeColor);
end;

end.
```

unit bibl;

interface

```
Type matr=array[1..10,1..10] of real;
      vect=array[1..10] of real;
procedure elem(n,m:integer; var c:matr);
procedure sum(n,m:integer; c:matr; var v:vect);
function nom_min(v:vect; k:integer):integer;
```

implementation

```
//
  procedure elem(n,m:integer;var :matr);
var i,j:integer;
begin for i:=1 to n do
  for j:=1 to m do
    [i,j]:=sin(i)+cos(j);
end;

//
  procedure sum (n,m:integer; :matr; var c:vect);
var i,j:integer;
```



```

begin
  for i:=1 to n do
    begin c[i]:=0;
      for j:=1 to m do
        v[i]:=v[i]+ c[i,j];
      end;
    end;
end;

//          -

function nom_min(v:vect; k:integer):integer;
var i:integer; min:real; nmin:integer;
begin min:=v[1]; nmin:=1;
  for i:=2 to k do
    if v[i]<min then
      begin min:=v[i];nmin:=i; end;
  nom_min:=nmin;
end;

end.

```

4.

```

1          -          ?
2          -          ?
3          -          ?
4          ?
5          -
6          ?
7          ?
8          Object Pascal          -
9          , -          ?
10         -          ?
11         ?
12         ,          -          ?
13         ?

```

```

14      i i      ?
15      i          i      i i      ?      i          i
i      .
16          i          i          i i      ?
17          i          i i      mylib.      -
      i i          ?
18      i i      i uselib      i          subr          :
      type vtype = array[1..10] of integer;
      procedure subr( A : vtype; var B : vtype );
      .....
i      p1          subr          :
      Unit p1;
      interface
      uses
      Windows, Messages, SysUtils, Classes, Graphics,
      Controls, Forms, Dialogs, StdCtrls, ExtCtrls;
      .....
      implementation
      {$R *.dfm}
      procedure TForm1.Button1Click(Sender: TObject);
      var x, y : array[1..10] of integer;
      begin
      .....
      subr( x, y );
      .....
      ?          - i,          ?
19      i          {$F+}?          i          ,          -
      i      .
20      i i          i          i ?
21          Pascal-          i          i i          .
22      i i          i          Pascal- i i          i?          i          i i          ' -
i,      i- i?
23      i i          i          i i interface      implementation.
24          i          i i i          ,
-      i i          i i?          i          .
25      i i          i Pvm          matr.          i          i
i i          MyLib,          i          i          i          -          i
RangMatr,          matr i          (          -
      )          integer.

```

5.

$(m -$, $n -$ $k \times k,$ $k = (m+n) \bmod 3+3,$
 $)$.

$$i,j = (k/2 - i + 0,3) \cdot (j - k + 5 \cdot n) \cdot \sin j,$$

$i=1, \dots, k; j=1, \dots, k.$

1 $i \ i$, i i :
 • ;
 • , $i \ i$ $i \ i$ $i \ i$.
 2 $i \ i$ $i \ i$.

1
 2
 3
 4
 5
 6
 7
 8
 9
 10 ,
 11
 12
 13
 14 ,
 15
 16
 17
 18
 19
 20
 21 -
 .
 22
 23
 24
 25 $k-$ $1-$
 26
 27 $(k-2)-$ $(k-1)-$
 28 $1- \ k-$
 29 $1- \ k-$
 30 ,

10

-

1

i " " i i -
i i , ? . -

2 i i i

2.1

i i , i i i i i -
i i . i ' i Pascal -

- , i i i -
i i i -

10.1 -
(.10.1).

/	I.						
		4	4	4	5	12	1983
1	I.						
2	.	5	5	5	27	10	1985

	.						
3	.	3	3	4	31	1	1984
	.						
4	.	4	5	5	10	7	1985
	.						

i : *i* *i* *i* *i* - *i* -
) ;
) *i* , *L* . - *i* ;
) *i* - *i* ;
) ():

- *i* *i* 1 31;
i - *i* *i* 1 12;
i - *i* .

“ ”.

2.2 ” ”

“ ”

record **end** Type:

Type *NameR* = **record**
NameP1 : *TypeP1*;
NameP2 : *TypeP2*;
 ...
NamePn : *TypePn*;
end;

NameR - *i* ' ;
NameP1, *NameP2*, ..., *NamePn* - *i* ;

```

TypeP1, TypeP2, . . . ,TypePn – (Real, Integer,
Boolean, Char i , i i Type).
(Var)
, “ ”.
i i i i .I i -
i i . “ -
”
- , :
Type tovar = record
    nazva : string [15];
    cena : real;
    kilk : integer;
end;
Var t : tovar; mt : array[1.. 50] of tovar;
t mt ( 50 ) “ -
” (tovar).
, , “. ”.
i i , .10.1, :
Type student = record
    nom : integer;
    pib : string[20] ;
    ocink : array [1..3] of integer;
    datar = record
        day : 1..31;
        month : 1..12;
        year : integer;
    end;
end;
Var st: student; gr : array [1..4] of student;
i st – ( i), i gr – i (
i).
2.3 “ ”
i ’ , i ( i ’ ). i i i -
i i . , i i -
st . pib = ’ . .’;
pib i st ’ . .’;
gr[4] . ocink[2]:=5;

```

```

?      5      ocink      o      -
i gr ;

      st.datar.month:= 4 ;
      month      datar      i      st      4.
i      i      .      i ,
      "      "      ,      -
      i      :

Type complex = record
      re,im : real
end;
Var x,y,z : complex;

i      i      i      i      i      i      "      "; re      im -
      i ( i      i      i      ) ; x, y, z - i      i
      omplex .

      x.re:=5.3;
      x.im:= -8.29e -2;

      i      i      i
(x = 5,3 - 0,0879 i );

      z.re:= x.re + .re;
      z.im:= x.im + y.im;

      z      x      y.

2.4      ?      .      i

      i

,

      with nv do
      begin
      <      i      i      >
      end;

nv - i '      i      "      "      .
      i      i .      i      i      i

```

i , i -

i i :

```
) with x do
  begin
    re:=5.3;
    im:=-8.29e -2
  end;
```

```
) with st do
  st . pib = ' . .';
```

```
) with st . datar do
  begin
    month := 4;
    if year < 1972 then k:= k+1
  end;
```

Object Pascal

i i ,

, " " . -

i i i . -

, i i i

i i : , i , i , ,

, i i ; i , i , ,

i :

//

```
type pol = (mug, gen);
  people = record
{
  fam : string[15];
  gr : 1800..2000;
  mg : pol;
{
  case pol of
    mug : ( voen : boolean; spec : string[9] );
    gen : ( zamug : boolean; child : integer);
  end;
```

```
{ }
```

```
var chel1, chel2: people;
  ved : array[1..1000] of people;
```

B i ase -

, i i (ase
end, end).

i i i (i), i i

3

(20) , : , -
 , . , -
 .

10.1.

10.1 –

```

:
ComboBox1 (
:
Items;
-
– Text);
RadioGroup1 (
:
Items;
– ItemIndex);
Edit1, Edit2, Edit3.

```

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
    ComboBox1: TComboBox;
    Label1: TLabel; Label2: TLabel; Label3: TLabel; Label4: TLabel;
    RadioGroup1: TRadioGroup;
    Button1: TButton; Button2: TButton; Button3: TButton;
    Button4: TButton; Button5: TButton;
    Memo1: TMemo; Memo2: TMemo;
    Edit1: TEdit; Edit2: TEdit; Edit3: TEdit;
    procedure TForm1.FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    .....
implementation
    {$R *.DFM}

    Type abitur = record
      country: string[10];
      oblast: string[15];
      family: string[12];
      sr_bal: real;
      medal: char;
    end;

    {
      var m_zap:array[1..20]of abitur; {
        k:integer; {
    }

    //
    procedure TForm1.FormCreate(Sender: TObject);
    begin
    k:=0;
    end;

    //
    procedure TForm1.Button1Click(Sender: TObject);
    begin

```

```

        k:=k+1;
    with m_zap[k] do
    begin
        country:=ComboBox1.text;
        oblast:=Edit1.text;
        family:=Edit2.text;
        sr_bal:=StrToFloat(Edit3.Text);
        Case RadioGroup1.ItemIndex of
            0: medal:= ' ';
            1: medal:= ' ';
            2: medal:= ' ';
        end;
    end;
end;

//          "          "
procedure TForm1.Button2Click(Sender: TObject);
begin
    k:=0;
    ComboBox1.Clear;
    Edit1.Clear; Edit2.Clear; Edit3.Clear;
    Memo1.Clear;
end;

//          "          "
procedure TForm1.Button3Click(Sender: TObject);
var sbal,dan:string; smed:string; i:integer;
begin
    Memo1.Clear;
    Memo1.Lines.Add('          |          |          |          |          ');
    for i:=1 to k do
        with m_zap[i] do
        begin
            case medal of
                ' ': smed:= ' ';
                ' ': smed:= ' ';
                ' ': smed:= ' ';
            end;
            str(sr_bal:5:2,sbal);
            dan:=family+' | '+country+' | '+oblast+' | '+sbal+' | '+smed;
            Memo1.Lines.Add(dan);
        end;
    end;
end;

//          "          "

```

```

procedure TForm1.Button4Click(Sender: TObject);
var sbal,dan:string; smed:string;i:integer;
begin
    memo2.Lines.Add(' ');
    for i:=1 to k do
        with m_zap[i] do
            if (medal=' ') Or (medal=' ') then
                begin
                    case medal of
                        ' ': smed:=' ';
                        ' ': smed:=' ';
                    end;
                    dan:=family+'|'+country+'|'+smed;
                    memo2.Lines.Add(dan);
                end;
    end;

// " "
procedure TForm1.Button5Click(Sender: TObject);
begin
    close
end;

```

end.

```

4. i
1 i i i Object Pascal?
2 ' ?
3 ,
4 i i? i ?
5 i i ?
6 ? i " "?
7 , i , i i : -
8 , , i i .

5

1 . i ,
i i , RadioGroup
ComboBox.
2 i i i i ,
. 10.2, i i i . i i i
5.

```

10.2 –

i	I	i
1...6	<i>i i</i>	: i , i , ,

...

7...12	, i : i , , i , i i , ,
13...18	i i: , i , i i , i -
19...24	, , i, : CD ROM, ' i' -
25...30	i i i: , i ic i - i , i

3 , (. 2) “ -
”. : Edit, StringGrid,
ListBox, RadioGroup ComboBox (). -
4 i i i , i
I i i
1 i i i
2 i i
i i . i i i
3 i i i
4 i i i i i i 300 . -
5 i i i i , 60 i , i
i i . i i i i i
6 i i . i i i i i
i i . i i
7 i i
i i . i , “5”, -
8 i i . i i i ,
9 “5”, i i i i , i
ic “5”, i i i i i ,
10 “4” “5”.
11 i i i , i 19 i ,
ci i i . i i i i i
12 , i i i i .
i i i i , i , -
13 i i i . i i i
14 i i i
15 . ic i i
i i .

16 i i , -
 , i i i i .
 17 i i i , i ,
 i i
 18 i i i i i -
 i , i i i i . ,
 19 i i , ' i i
 10 i i i i .
 20 , i , -
 21 i . , i i i
 22 , i
 i i . ,
 23 , CD ROM, i
 i i .
 24 i i i , i
 400 i , i i .
 25 i i i i i i i -
 i i i .
 26 i , i i , i,
 i , i i i i ,
 27 i .
 i i i i .
 28 i i i i
 i (i i i i i i .
 29 i i i i , i i .
 i i (i i i i), i
 i i i .
 30 i , i i i , i ,
 i i , -
 .

11

1

i . i

i .

2. i i i

2.1

i i -

i i i . -

i Object Pascal i . -

— i i -

. i i i , i i -

i i , i i i , i i . -

i i i , (i i) -

, , . i i -

, , i , . i i -

i i , i i -

i i i i i i i i i -

, . i ,

i , i . -

i Object Pascal i i: i i, -

i .

2.2

i

i :

Type *NameTyp* = **file of** *ElemTyp*;**Var** *NameV* : *NameTyp*;**Var** *NameV* : **file of** *ElemTyp*;

NameTyp – i ' ; *NameV* – i ' i ;

ElemTyp – (*ElemTyp* - , i

file). , *ff* ,

i i ,

Var *ff*: file of integer;

```

Type kadr = record
    pib : string[12] ;
    year : integer;
    oklad : real
end;
fpk = file of kadr;

```

```

Var ct : fpk;

```

```

    i    i                i                “ -
”      .
.      i                i                “ ”.
      .I i                ,                -
“      ”                i                i
      .                i                i (EOF).
      i                i                ,                i                -
i      i                i                i                i,                -
      i                “                -                ”.

```

2.3

1) **AssignFile**(NameV, File_Name); –

```

    i    NameV,                , “                ” i
    ,                i ’ File_Name.    File_Name –
    ,    i                i ’                i    i ’
    .    -                i                .
    ,

```

```

    AssignFile(h, 'c: \myfile.dat');

```

```

    i    h    i                ’myfile.dat’,    i
    i,                ’ c.

```

2) **CloseFile**(NameV); –

```

?      i    i                i i i    i    i .
      i                i                i                i
i .

```

3) **Erase**(NameV); –

4) **Read**(NameV, a1,a2, ..., aN); –

```

    i                a1,a2, ..., aN.
    i    i    , i                i .
    ,

```

5) **RenameFile**(Old_Name,New_Name); –

```

i      i ’ ’New_Name’.

```


-
- 6) **Reset**(NameV); – i i . -
- 7) **Rewrite**(NameV); – NameV.
i , i .
- 8) **Seek**(NameV, n); – NameV n- .
0, - 1 i . .
- 9) **Write**(NameV, A1, A2, ... , An); – NameV i i
i . i i , i i
i i i . i
i i . (i , i -
reset) i .

2.4 i

- 1) **Eof**(NameV); – True, NameV -
i i , i False -
- 2) **FilePos**(NameV); – , i -
- 3) **FileSize**(NameV); – i i i i.
i . i i i i

2.5

- i i i
:
- 1) i , AssignFile i
Reset;
- 2) i , Read, Seek;
- 3) , CloseFile.
i i
- :
- 1) i , AssignFile,
Rewrite, Reset;
- 2) i , Write, Seek;
- 3) , CloseFile.
-

2.6

i

```

      i      i      -      i      -
 1)      FileExists(File_Name).
      {$I±}
      i      i      -      i      i      -
      -i      i      i      i      i      -
      .      i      i      i      i      -
i      ,      i      i
      -      .      i      ,
      i      IOresult.      i      0,
      i      i      -      ,      i      -
      ( )      i      .      i      i      i
      .
      ,      '      i      ,
      i      '      ,      i      i      i      ,
      i      i      i      '      .
i      ,      i      i      .
.....
Var s : string[20]; b : boolean; f : file; ...
begin
  b:=false;
  repeat
    ShowMessage ('      '      Edit1');
    Edit1.Text:= s;
    AssignFile(f,s);
    {$I-} reset(f); {$I+}
    if IOresult=0 then b:=true
      else ShowMessage('      ');
  until b;
  ShowMessage ('      : ', FileSize(f) );
  CloseFile(f);
end;
2)      FileExists(File_Name); -      True,
      i      ,      i      '      File_Name, i False -
      ,      '      .doc'
      ,
      ,
      .
If FileExists ('      .doc') then reset(f)
      else begin
          rewrite(f);
          ShowMessage('      ');
      end;

```

3.

StringGrid.

. 11.1.

Робота с типізованими файлами

Вихід

Очистити все
Закрити проект

Відомості про товари

Додати новий запис

Товари на складі

Найменування

Кількість

Ціна

Рік випуску

Сортування за ціною

Найменування	Кількість	Ціна	Рік випуску
Телевізор	15	850.00	1999
Муз. центр	5	1150.00	2000
Пральна маши	9	1580.00	2001
Відео	25	2300.00	201
Комп'ютер	30	2950.00	2002

Загальна кількість =

Товар з міні ціною

11.1 –

MainMenu.

StringGrid1.

OpenDialog (

OpenDialog1.Execute

True,

OpenDialog1.FileName

:

unit Unit1;**interface****uses**Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, ExtCtrls, Menus, Grids;**type**

```
TForm1 = class(TForm)
  MainMenu1: TMainMenu;
  OpenDialog1: TOpenDialog;
  Button1: TButton; Button2: TButton; Button3: TButton;
  Button4: TButton; Button5: TButton; Button6: TButton;
  Edit1: TEdit; Edit2: TEdit; Edit3: TEdit;
  Edit4: TEdit; Edit5: TEdit; Edit6: TEdit;
  Label1: TLabel; Label2: TLabel; Label3: TLabel;
  Label4: TLabel; Label5: TLabel;
  StringGrid1: TStringGrid;
  N1: TMenuItem;
  N2: TMenuItem;
  N3: TMenuItem;
  procedure TForm1.FormCreate(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure Button3Click(Sender: TObject);
  procedure Button4Click(Sender: TObject);
  procedure Button5Click(Sender: TObject);
  procedure N2Click(Sender: TObject);
  procedure N3Click(Sender: TObject);
  .....
```

implementation

{\$R *.DFM}

type tovar=**record**

```
  name:string[12];
  kol:integer;
  cena:real;
  god:integer;
end;
```

```

//
var f:file of tovar; {
    z:tovar; {
        namefile:string; {
//
procedure TForm1.FormCreate(Sender: TObject);
begin
    if OpenDialog1.Execute then namefile:=OpenDialog1.FileName;
    AssignFile(F,namefile);
    if FileExists(namefile) then
        begin
            reset(f);
            ShowMessage('
        end
        else
        begin
            rewrite(f); ShowMessage('
        end;
end;

//
procedure TForm1.Button1Click(Sender: TObject);
begin
{
    if (length(edit1.text)=0) or (length(edit2.text)=0) or
        (length(edit3.text)=0) or (length(edit4.text)=0) then
        begin
            showmessage('
        exit;
        end ;
{
    reset(f);
    seek(f,FileSize(f)); {<-
    with z do
        begin
            name:=edit1.text;
            kol:=StrToInt(edit2.text);
            cena:=StrToFloat(edit3.text);
            god:=StrToInt(edit4.text);
        end;
    write(f,z);
{
    edit1.Clear; edit2.Clear; edit3.Clear; edit4.Clear;
    closefile(f);
end;

```

```

//                                     “                 ”
procedure TForm1.Button2Click(Sender: TObject);
var scena:string;
    i:integer;
begin
reset(f);
seek(f,0); {<-                               }
i:=0;
StringGrid1.Cells[0,0]:=’           ’;
StringGrid1.Cells[1,0]:=’           ’;
StringGrid1.Cells[2,0]:=’           ’;
StringGrid1.Cells[3,0]:=’           ’;
while not eof(f)do
begin
    read(f,z);
    i:=i+1;
    with z do
        begin
            str(cena:1:2,scena);
            StringGrid1.Cells[0,i]:=name;
            StringGrid1.Cells[1,i]:=IntToStr(kol);
            StringGrid1.Cells[2,i]:=scena;
            StringGrid1.Cells[3,i]:=IntToStr(God);
        end;
    closefile(f);
end;
end;

//                                     “                 ”
procedure TForm1.Button3Click(Sender: TObject);
var k:integer;
begin
reset(f);
seek(f,0); {<-                               }
k:=0;
    while not eof(f)do
        begin
            read(f,z);
            k:=k+z.kol;
        end;
edit5.text:=IntToStr(k);
closefile(f);
end;

```

```

//          “          min          ”
procedure TForm1.Button4Click(Sender: TObject);
var zmin:tovar; min:real; s:string;
begin
  reset(f);
  seek(f,0); {<-          }
  read(f,z);
  zmin:=z; min:=z.cena;
  while not eof(f)do
    begin
      read(f,z);
      with z do
        if cena<min then
          begin min:=cena; zmin:=z;end;
        end;
    with zmin do
      s:=name+'|'+IntToStr(kol)+'|'+floatToStr(cena)+'|'+IntToStr(God);
    edit6.text:=s;
  closefile(f);
end;

//          “          ”
procedure TForm1.Button5Click(Sender: TObject);
var k,i,j:integer; zi,zj:tovar; s:string;
begin
  reset(f);
  seek(f,0); {<-          }
  Memo1.Clear;
  k:=FileSize(F); {<-          }
  {C          }
  for i:=0 to k-2 do
    begin
      for j:=i+1 to k-1 do
        begin
          seek(f,j);read(f,zi);
          seek(f,i); read(f,zj);
          if zi.cena>zj.cena then
            begin
              seek(f,j); write(f,zi);
              seek(f,i); write(f,zj);
            end;
          end;
        end;
      end;
    end;
end;

```

```

{                               Memo}
seek(f,0); {<-                               }
while not eof(f) do
  with z do
    begin
      read(f,z);
      s:=name+'|'+IntToStr(kol)+'|'+floatToStr(cena)+'|'+IntToStr(God);
      Memo1.Lines.Add(s);
    end;
  closefile(f);
end;

//                               “ /                               ”
procedure TForm1.N2Click(Sender: TObject);
begin
  {                               }
  rewrite(f); closefile(f);
  {                               }
  memo1.Clear;
  edit1.Clear; edit2.Clear;edit3.Clear;
  edit4.Clear;edit5.Clear;edit6.Clear;
  end;

//                               “ /                               ”
procedure TForm1.N3Click(Sender: TObject);
begin
  close;
end;

end.

```

5. i

```

1  i           ?
2
3  i           i           i           ?
4  i           i           i
?
5  i           ?
6  i           i           i           i           'kot.dat' -
i           , n, m.
7  i           i           i           i           i
d  'dom.dat'.

```


1 i i i i i , . 11.1,
i i i . i i i i - i 7 10.

11.1 –

i	I i
1...6	i i: , i ic i i , - i
7...12	i i : i , i , ,
13...18	i: i , i , i i , ,
19...24	i i: , i , i i , i - ,
25...30	' : , ' , ' i, CD ROM, i -

2
.1 i i i i . -
3 i , - ,
i .
I i i
1 i i i i i i i -
i i i . i i
2 , i i i i ,
i .
3 i i i i -
4 i i i .
i (i i) i i .
5 i i i (i i) , i
6 i i i , i i
i , i . i
7 i i i i .
8 i i
i i .

9 i i i

10 i i i i i 300 . -

i i i i .

11 i i i , 60 i , i

i i .

12 i i i i i

i i .

13 i i

i i .

14 i i , "5", -

i i .

15 i i i , i

ic "5", i i .

16 i i i i i ,

"4" "5".

17 i i i , i 19 i ,

ci i i .

18 i i i i i i .

i , i i i .

19 , i i i , -

i i i .

20 i i i

. ic i i

21 i i .

22 i i ,

, i i i .

23 i i i , i ,

i i .

24 i , i i i i i -

25 i i , , , i

i 10 i i i .

26 , i i , -

27 , i i -

. i i

28 , i

i i .

29 , CD ROM, i

i i .

30 i i i , i

400 i , i i .

12

c

1

i

2

i

i

i

i

2.1

Var NameV: TextFile;

NameV = ' ' ; TextFile =

AssignFile,
AssignFile

AssignFile

11.

Rewrite.

Reset.

Append.

CloseFile.

2.1.1

Write WriteLn.

Write(NameV, spisok);

Writeln(NameV, spisok);

spisok –

.
.

, f TextFile,
1 2

Writeln(f, ' ', 1, 2);

Write Writeln

Writeln

"

".

Var MyTextFile: TextFile; s: string; i:integer;

begin

AssignFile(MyTextFile, 'Dan.Txt');

rewrite(MyTextFile);

for i:=1 to 3 do

begin

s:= ' ';

writeln(MyTextFile, s, i);

end;

closefile(MyTextFile);

end;

"Dan.Txt"

:

1

2

3

2.1.2

,

Append,

Writeln

Write.

"Dan.Txt"

Var MyTextFile: TextFile; S: String; i:Integer;

begin

AssignFile(MyTextFile, 'Dan.Txt');

Append(MyTextFile);

for i:=4 to 5 do

begin

s:= ' ';

Writeln(MyTextFile, S, i);

```

    end;
    CloseFile(MyTextFile);
end;
        "Dan.Txt"
            1
            2
            3
            4
            5

```

2.1.3

```

i      i      :      i      -
1) i      ,      AssignFile i
Reset;
2)      i      ,
      Read(NameV, spisok)
      Readln(NameV, spisok);
      spisok -
3)      ,      CloseFile.
      "Dan.Txt":

```

```

Var MyTextFile:TextFile;
      S:string[12]; j:integer;
begin
    AssignFile(MyTextFile, 'Dan.Txt');
    reset(MyTextFile);
    while not eof(MyTextFile) do
    begin
      readln(MyTextFile,S,j);
      Memo1.Lines.Add(S+IntToStr(j));
    end;
    CloseFile(MyTextFile);
end;

```

```

S      ,
      str ng[12].
S -

```

```

      j.

```

Memo.

2.1.4

- 1 **AssignFile**(NameV, File_Name) – Name V
File_Name
- 2 **Append**(NameV) –
- 3 **CloseFile**(NameV) –
- 4 **DeleteFile**(File_Name) –
- 5 **Eof**(NameV) – ;
True, False -
- 6 **Erase**(NameV) - , ' .
- 7 **FileExists**(File_Name) – File_Name.
True, File_Name , False -
- 8 **IoResult**(NameV) –
- 9 **Read**(NameV, a1,a2, ..., aN), **Readln**(Name V, a1,a2, ..., aN) -
- 10 **RenameFile**(Old_Name,New_Name) –
Old_Name New_Name.
- 11 **Reset**(NameV) –
- 12 **Write**(NameV, a1,a2, ..., aN), **Writeln**(Name V, a1,a2, ..., aN) –

3

c

3.1

1

- 1)
- 2)
- 3)
- 4)

StringGrid

“ ”
OnActivate.

```

procedure TForm1.FormActivate(Sender: TObject);
var i:integer;
begin
for i:=0 to ComponentCount-1 do
begin
if components[i] is TLabel then (components[i] as TLabel).Visible:=false;
if components[i] is TEdit then (components[i] as TEdit).Visible:=false;
if components[i] is TButton then (components[i] as TButton).Visible:=false;
if components[i] is TStaticText then (components[i] as
TStaticText).Visible:=false;
end;
with strg1 do
begin
cells[0,0]:= ' ';
cells[1,0]:= ' . ' ;
cells[2,0]:= ' ' ;
cells[3,0]:= ' . ' ;
Visible:=False;
end;
App_Fil.Enabled:=false;
Viw_Fil.Enabled:=false;
N2.Enabled:=false;
Memo1.Hide;
end;

```

Implementation

Btn2:

OnClick

implementation

{ \$R *.DFM }

```

var f:TextFile;
    FName:String[18];
    nbtn2:integer;

```

“ ” — “ ”
“ ” :

```

procedure TForm1.RewFilClick(Sender: TObject);
begin
  Lbl5.Show; Btn2.Show; Ed4.Show; Btn2.Enabled:=false;
  Ed4.setFocus; nbtn2:=1;
end;

procedure TForm1.RedFilClick(Sender: TObject);
begin
  Lbl5.Show; Btn2.Show; Ed4.Show; Btn2.Enabled:=false;
  Ed4.setFocus; nbtn2:=2;
end;

```

```

, , Ed4 ,
, " Ed4 !". -

```

```

procedure TForm1.Btn2Click(Sender: TObject);
begin
  if Ed4.Text="" then
    begin
      showmessage(' ', !+#13+ ');
      Exit;
    end;
  fname:=Ed4.Text;
  assignfile(f,fname);
  if nbtn2=1 then rewrite(f) else
    begin
      {$I-}
      reset(f);
      {$I+}
      if IOResult<>0 then
        begin
          showmessage (' !'+#13+ ');
          Ed4.Clear; Ed4.SetFocus;
          exit;
        end;
    end;
  closefile(f);
  Ed4.Clear; Lbl5.Hide;
  Ed4.Hide;
  Btn2.Hide;
  App_Fil.Enabled:=true;
  Viw_Fil.Enabled:=true;
  N2.Enabled:=true;
end;

```

OnChange Ed4:

```
procedure TForm1.Ed4Change(Sender: TObject);
begin Btn2.Enabled:=true; end;
```

```
procedure TForm1.App_FilClick(Sender: TObject);
begin
  Lbl1.Show; Lbl2.Show; Lbl3.Show; Lbl4.Show;
  Ed1.Show; Ed2.Show; Ed3.Show;
  Ed1.SetFocus; Ed2.Enabled:=False; Ed3.Enabled:=false;
  Btn1.Show; Btn1.Enabled:=false;
end;
```

```
procedure TForm1.Viw_FilClick(Sender: TObject);
var snobl,snGor:string[20];snkol:string[10];
begin
  Lbl6.Show; Memo1.Show; Btn3.Show;
  reset(f); N1.Enabled:=false; N2.Enabled:=false;
  Memo1.Width:=320;
  While not eof(f) do
    begin
      Readln(f,snobl,snGor,snkol);
      Memo1.Lines.Add(snobl+snGor+snkol);
    end;
  closefile(f);
end;
```

```

    ”,
... Ed3,           ”,           ”,           ”,
                   OnChange.   Ed1
                   Ed3,         OnKeyPress.

```

```
procedure TForm1.Ed1Change(Sender: TObject);
begin Ed2.Enabled:=true; end;
procedure TForm1.Ed2Change(Sender: TObject);
begin Ed3.Enabled:=true; end;
procedure TForm1.Ed3Change(Sender: TObject);
begin Btn1.Enabled:=true; end;
```

```
procedure TForm1.Ed3KeyPress(Sender: TObject; var Key: Char);
begin if not (key in ['0'..'9']) then key:=Chr(0); end;
```

```
“ “ ” ’
“ ” , -
```

```
procedure TForm1.Btn3Click(Sender: TObject);
begin
  Lbl6.Hide; Memo1.Clear; Memo1.Hide; Btn3.Hide;
  N2.Enabled:=true; N1.Enabled:=true;
end;
```

```
’
: “ ’ ”; “ ”; “
:
:
```

```
procedure TForm1.Kol_FilClick(Sender: TObject);
var s_kol:integer; skol:integer;
    snobl,snGor:String[20];
begin
  Lbl9.Show; St2.Show;
  Btn4.Show; Lbl9.Left:=90; St2.Top:=230; Lbl9.Top:=230;
  St2.Left:=300; St2.Width:=160; Btn4.Left:=145;
  s_kol:=0;
  reset(f);
  while not eof(f) do
    begin
      readln(f,snobl,snGor,skol);
      s_kol:=s_kol+skol;
    end;
  closefile(f);
  St2.Caption:=IntToStr(s_kol);
end;
```

```
procedure TForm1.Mil_FilClick(Sender: TObject);
var kol:word; skol:integer;
    snobl,snGor:String[20];
begin
  Lbl7.Show; StrG1.Show;
  Btn4.Show; Lbl7.Left:=168; Lbl7.Top:=40;
  Strg1.Top:=60; Strg1.Left:=100;
  reset(f); kol:=1; Btn4.Left:=145;
```

```
while not eof(f) do
begin
  readln(f,snobl,snGor,skol);
  if skol>1000000 then
    begin
      Strg1.Cells[0,kol]:=IntToStr(kol);
      Strg1.Cells[1,kol]:=Trim(snobl);
      Strg1.Cells[2,kol]:=Trim(snGor);
      Strg1.Cells[3,kol]:=InttoStr(skol);
      kol:=kol+1;
      Strg1.RowCount:=kol;
      if Strg1.Height<130 Then Strg1.Height:=Strg1.RowCount*29;
    end;
  end;
closefile(f);
end;
```

```
procedure TForm1.Max_FilClick(Sender: TObject);
var max_kol:integer; skol:integer;
snobl,snGor,smax:string[20];
begin
  Lbl8.Show; St1.Show;
  Btn4.Show; Lbl8.Left:=90; St1.Top:=210; Lbl8.Top:=210;
  St1.Left:=300; St1.Width:=160; Btn4.Left:=145;
  max_kol:=-1000000;
  reset(f);
  while not eof(f) do
    begin
      readln(f,snobl,snGor,skol);
      if skol>max_kol then
        begin max_kol:=skol; smax:=snGor; end;
    end;
  closefile(f);
  St1.Caption:=Trim(Smax)+' '+IntToStr(Max_kol);
end;
```

Btn4

-

```
procedure TForm1.Btn4Click(Sender: TObject);
begin
  Lbl7.Hide; StrG1.Hide; Btn4.Hide; lbl8.Hide; St1.Hide;
  lbl9.Hide; St2.Hide; N1.Enabled:=true;
end;
```


“ ”,
TabSheet. *Name* , *Capt on*
Tbs1 ... Tbs4.
PageControl ,
 , *Button*,
 , , .12.2.

12.2 –

Button (), ' *Label* (*Ed t* ()).
 12.3.

12.3 –

2

Form	Caption Position	PoScreenCenter
Label1 ... Label5	Caption Name	Lbl1 ... Lbl5
Button	Caption Name	Dtn1
Edit1 ... Edit4	Name Text	Ed1 ... Ed4

```

OnAct vate,
-
-
Enabled      Btn1,      Ed2 ... Ed4,
False.      ,      Act vePage      PageControll
      Tbs1,      ,
      “      ”.      “
      ”,      Enabled      Button,
      False. ,      ,
      Str ngGr d,      “ ,
”,      2,      ,
      .
      F leEx sts      ,
      ,
      ( . . 2.1.5).      onAct vate
:

```

```

Procedure TForm1.FormActivate(Sender: TObject);
begin
  assignfile(f,'Dan.txt');
  if FileExists('Dan.txt') then reset(f) else rewrite(f);
  closefile(f);
  Btn1.Enabled:=false;
  Ed2.Enabled:=false;
  Ed3.Enabled:=false;
  Ed4.Enabled:=false;
  PageControl1.ActivePage:=Tbs1;
  Btn5.Enabled:=False;
  lbl7.Hide; lbl8.Hide;
  Strg1.RowCount:=2;
  StrG2.RowCount:=2;
  StrG1.Hide; StrG2.Hide;
end;

```


*Ed2...Ed4**: OnKeyPress**, OnChange*

:

```
procedure TForm1.Ed1Change(Sender: TObject);
begin Ed2.Enabled:=True; end;
```

```
procedure TForm1.Ed2Change(Sender: TObject);
begin Ed3.Enabled:=True; end;
```

```
procedure TForm1.Ed3Change(Sender: TObject);
begin Ed4.Enabled:=True; end;
```

```
procedure TForm1.Ed4Change(Sender: TObject);
begin Btn1.Enabled:=True; end;
```

```
procedure TForm1.Ed2KeyPress(Sender: TObject; var Key: Char);
begin
  f not (key in ['0'..'9']) then key:=Chr(0);
end;
```

*Ed2KeyPress**Ed3 Ed4.**implementat on*

:

“ ”,

Type spisok=**record**

fio:string[16];

mat,fiz,inf:word;

srbal:real;

end;**var** st:spisok;

f:TextFile;

sw:array[1..5]of string[50];

“

”,

“

”.

```
procedure TForm1.Btn1Click(Sender: TObject);
```

begin

f (Ed1.Text=' ') or (Ed2.text=' ') or (Ed3.text=' ') or (Ed4.Text=' ') then

begin

showmessage('

!'+#13+'

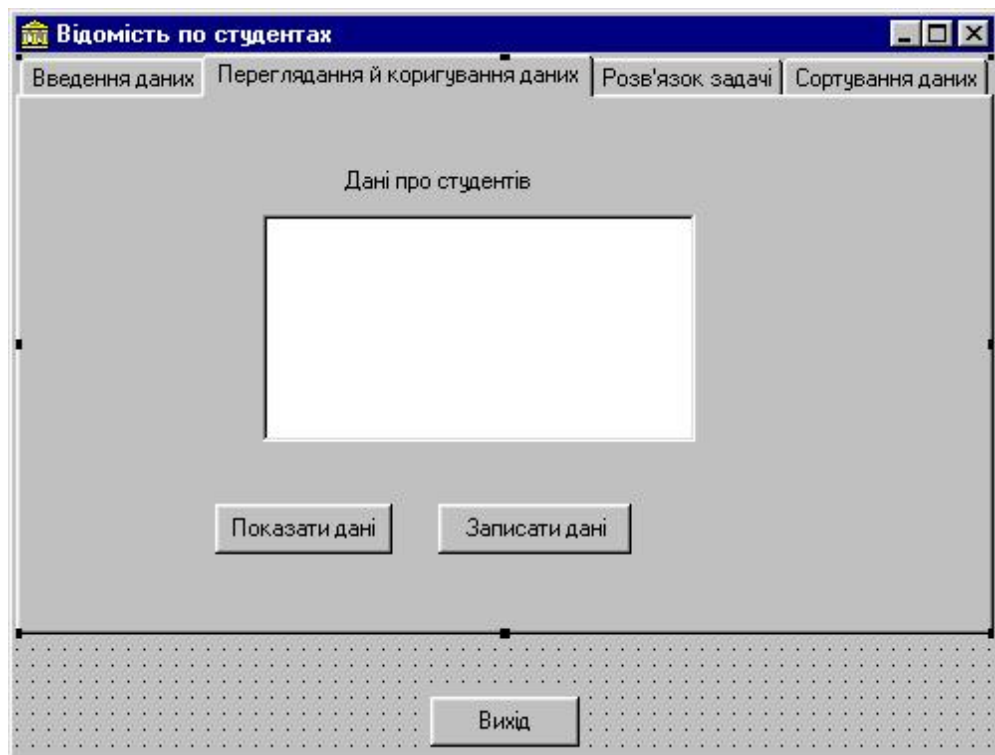
!');

```

Ed1.SetFocus;
Exit;
end;
append(f);
with st do
begin
fio:=Ed1.text; mat:=StrToInt(Ed2.Text);
fiz:=StrToInt(Ed3.Text); inf:=StrToInt(Ed4.Text);
srbal:=(mat+fiz+inf)/3;
writeln(f,fio,' ',mat,' ',' ',fiz,' ',inf,' ',srbal:1:2);
closefile(f);
Ed1.Clear; Ed2.Clear; Ed3.Clear; Ed4.Clear;
Ed1.SetFocus; ed2.Enabled:=false; Ed3.Enabled:=false;
Btn1.Enabled:=false; ed4.enabled:=false;
end;
end;

```

. 12.3.



12.3 –

: *Memo1*,

Button

Label.

Memo.

Memo

12.4.

12.4 –

2

Memo1	ScrollBars	SsVertical
Label	Caption Name	Lbl6
Button4	Name Caption	Btn4
Button5	Name Caption	Btn5

```
procedure TForm1.Btn4Click(Sender: TObject);
```

```
var s:String;
```

```
begin
```

```
  Memo1.Clear;
```

```
  reset(f);
```

```
  while not eof(f) do
```

```
    begin
```

```
      readln(f,s);
```

```
      Memo1.Lines.Add(s);
```

```
    end;
```

```
  closefile(f);
```

```
    showmessage('!'+#13+'
```

```
'+#13+' -
```

```
'+#13+'
```

```
,'+#13+'
```

```
,
```

```
!');
```

```
  Btn5.Enabled:=false;
```

```
end;
```

```
procedure TForm1.Btn5Click(Sender: TObject);
```

```
var s,sn:string; i,j,k,m:word; spr:array[1..4] of byte;
```

```
  err:array[2..4]of integer;
```

```
  buf:TextFile;
```

```
begin
```

```
  assignfile(buf,'Dat.Txt');
```

```
  rewrite(buf);
```

```
  for k:=0 to Memo1.Lines.Count-1 do
```

```
    begin
```

```
      s:=Memo1.Lines[k];
```

```
      if s="" then continue;
```

```

i:=0;
s:=s+' ';
sn:="";
while not (s[i] in ['0'..'9']) do
begin
  sn:=sn+s[i];
  i:=i+1;
end;
sw[1]:=trimleft(sn);
delete(s,1,i-1); s:=trimleft(s);
for i:=2 to 5 do
begin
  sw[i]:=copy(s,1,pos(' ',s)-1);
  delete(s,1,Pos(' ',s)-1);
  if i<5 then
  begin
    spr[i]:=0;
    for m:=1 to length(s) do
    begin
      if s[m]=' ' then
        begin spr[i]:=spr[i]+1; end
      else break;
    end;
    s:=trimleft(s);
  end;
  sw[i]:=trim(sw[i]);
end;
with st do
begin
  val(sw[2],mat,err[2]);
  val(sw[3],fiz,err[3]);
  val(sw[4],inf,err[4]);
  for j:=2 to 4 do
  if err[j]<>0 Then
  begin
    showmessage('
    ,
    !'+#13+
    !'+#13+'
    !');
    CloseFile(buf);
    Erase(buf);
    exit;
  end;
  srbal:=(mat+fiz+inf)/3;
  write(buf,sw[1],mat);
  sn:="";
  for j:=1 to spr[2] do

```

```

        sn:=sn+' ';
    write(buf,sn,fiz);
    sn:=";
    for j:=1 to spr[3] do
        sn:=sn+' ';
    write(buf,sn,inf);
    sn:=";
    for j:=1 to spr[4] do
        sn:=sn+' ';
    writeln(buf,sn,srbal:5:2);
end;
end;
closefile(buf);
rewrite(f);
reset(buf);
Memo1.Clear;
while not eof(buf) do
    begin
        readln(buf,s);
        Memo1.Lines.Add(s);
        writeln(f,s);
    end;
closefile(f);
closefile(buf);
erase(buf);
Btn5.Enabled:=false;
end;

```

(Bt1)

Memo.

(Bt2)

Memo,

”,

*OnChange**Memo,*

-

```

procedure TForm1.Memo1Change(Sender: TObject);
begin Btn5.Enabled:=true; end;

```

Memo

-

*spr:array[1..4] of**byte*

”

,

.

,

,

-

.

-

.

```

        : L stBox;
    ”, “
    : Btn6, Btn7, Btn8.
        Button.
        ”.
        Name
        : “
        -
        OnClick

```

```

procedure TForm1.Btn6Click(Sender: TObject);
var s:string;
begin
    reset(f);
    ListBox1.Sorted:=false; ListBox1.clear;
    While not eof(f) do
        begin
            Readln(f,s);
            ListBox1.Items.Add(s);
        end;
    closefile(f);
end;

```

```

procedure TForm1.Btn7Click(Sender: TObject);
begin Listbox1.Sorted:=true; end;

```

```

procedure TForm1.Btn8Click(Sender: TObject);
var s:string; k:word;
begin
    rewrite(f);
    for k:=0 to ListBox1.Items.Count-1 do
        begin
            s:=ListBox1.Items[k];
            writeln(f,s);
        end;
    closefile(f);
    ListBox1.Sorted:=false;
end;

```

```

L stBox,
    ,
    0
    :
    Button,
    (Name=Strg1)
    (Name=Strg2) -
    .
    Caption
        Sorted
        true.
        “
        ,
        ”.
        StringGrid,
        -
        StringGrid
        .
        ,
        Label
        -
        Caption
        : “

```



```

begin
  k1:=k1+1;
  strg1.Cells[0,k1]:=IntToStr(k1);
  for i:=1 to 5 do
    Strg1.Cells[i,k1]:=sw[i];
  Strg1.RowCount:=k1+1;
  if Strg1.Height<=90 then
    Strg1.Height:=strg1.RowCount*23;
  end;
mat:=StrToInt(sw[2]);
fiz:=StrToInt(sw[3]);
inf:=StrToInt(sw[4]);
if (mat<60) or (fiz<60) or (inf<60) then
  begin
    k2:=k2+1;
    Strg2.Cells[0,k2]:=IntToStr(k2);
    for i:=1 to 5 do
      Strg2.Cells[i,k2]:=sw[i];
    Strg2.RowCount:=k2+1;
    if Strg2.Height<=90 then
      strg2.Height:=Strg2.RowCount*23;
    end;
  end;
end;
closefile(f);
end;

```

```

procedure TForm1.Btn3Click(Sender: TObject);
var i,j:word;
begin
for i:=1 to Strg1.RowCount do
  for j:=1 to 5 do
    Strg1.Cells[j,i]:=”;
for i:=1 to Strg2.RowCount do
  for j:=1 to 5 do
    Strg2.Cells[j,i]:=”;
Strg1.RowCount:=2;
Strg2.RowCount:=2;
Strg1.Height:=47;
Strg2.Height:=47;
Strg1.Hide;
Strg2.Hide;
Lbl7.Hide;
Lbl8.Hide;
end;

```


4 i

```

1      ?
2      i          i          i ?
3      i          i          i ?
4      F.DAN.      i      -
          i          ( i          )      i
i      i          i “%”
5      i          i          i
      ‘STUD. T’.
6      i          i          i
i      ‘REZ.DAT’.
7      i          :
      )          WRITELN(F1,'INGA');
          i
WRITE(F1,'IN'); WRITELN(F1,'GA');
      )          WRITELN(F1,'INGA');
          i
WRITELN(F1,'IN'); WRITELN(F1,GA');
    
```

5

```

1      i          i          i ,          -
      ( . 12.5), i          i . i i
      - i 5 7.
2
-      (“ ” ’ Text Delphi).
3      Delph
.
4      i          i          i          ( . . 12.5),          i      -
          i          .
5      i          ,          -
          i          ,          -
          ,          i .
    
```

12.5 –

1	: , - , ,	- - -

2	: , , , -	, , -
3	: , , -	“ ” -
4	: , , -	, 1967 - 10
5	: , , -	, , -
6	: - , , (: 1- , 0-) -	, 1 -
7	: - , (), , -	10- , 17- , -
8	: , , -	, 3 -
9	: , , -	, - -
	(:)	

10	<p>， : ， - (- -)， : - () : : :)</p>	<p>， 10 .， - - - 22:40 22:42 .</p>
11	<p>: ， (- -)，</p>	<p>， ， -</p>
12	<p>- ， (- -))， ， -</p>	<p>- -</p>
13	<p>: (- -)， ，</p>	<p>， 0 ， - - 5</p>
14	<p>: - ， ， - ， ， -</p>	<p>- - - - 5-</p>
15	<p>: ， ， - ， ， - ，</p>	<p>Ford, 50000 2</p>
16	<p>: ， ， - ， (- -)，</p>	<p>， 5000 ， -</p>

17	: , , (, . - , . -),	- , (55)
18	: , , ,	- , 10 40 45
19	, : , , ,	- , 350 , - 32
20	: , , ,	- , - , 25
21	: , , ,	- , 95 , 100
22	: , , ,	- , “ ”, (10)
23	: , , ,	- ,

24	:	-
25	:	-
26	:	-
27	:	-
28	:	-
29	:	-
30	:	-

-
- 1 . . . Delphi: . . . -
: , 1999.
 - 2 . . . Delphi.
1. - : , 2001.
 - 3 . Delphi 5: . - .: , 2001.
 - 4 . Delphi 4: Object Pascal. - .: , 1999.
 - 5 . . Delphi 4: . - .: “ ”, 1998.
 - 6 . . . TURBO-
PASCAL 7.0. - M.: - , 1996.
 - 7 . . . -
: , 1997.
 - 8 . . Delphi . -
. - .: , 2000.
 - 9 . . Delphi 5. - .: , 2000.
-

i

73
813
9 i i25
10 -44
1154
1266
.....93

DELPHI

“ ”

2

: . . . , . . . , . . . ,
 . .
 . .
 . .
 , . .
 . .

