

```

1  unit ProcessMatrix;
2
3  interface
4
5
6  uses
7  Windows, Messages, SysUtils, Classes, Graphics, Controls,
8  Forms, Dialogs, StdCtrls, ExtCtrls;
9
10 type
11     TMatrix = array of array of Real;
12
13 procedure CreateMatrix(var Matrix:TMatrix; AxisY,AxisX: integer);
14 procedure GetCoord(var AxisY,AxisX: integer);
15 procedure InputMatrix(var Matrix:TMatrix);
16     procedure RandomInputMatrix(var Matrix:TMatrix);
17     procedure UserInputMatrix(var Matrix:TMatrix);
18
19 procedure OpenTextFile(var TF:TextFile);
20     procedure ODOpenFile(var TF:TextFile);
21     procedure PathOpenFile(var TF:TextFile);
22
23 procedure WriteMatrixInTextFile(var TF:TextFile; Matrix:TMatrix);
24 procedure WriteIntMatrixInTextFile(var TF:TextFile; Matrix:TMatrix);
25
26 procedure ProcessingFlip(var Matrix:TMatrix;y,x:integer);
27 procedure ProcessingSpreading(var Matrix:TMatrix;n:integer);
28
29 { Dock
30 Input : User choice -> Random or Manual input
31 Output: User choice -> OpenFileDialog File or PopupPath File
32 }
33
34
35 implementation
36
37 procedure GetCoord(var AxisY,AxisX: integer);
38 const
39     IBTitle = 'Создание матрицы';
40     IBMessage1 = 'Ведите число строк: '; {строки - ось y}
41     IBMessage2 = 'Ведите число столбцов: '; {столбцы - ось x}
42     IBDefaultValue = '2';
43
44 begin
45     AxisY:= StrToInt(InputBox(IBTitle,IBMessage1,IBDefaultValue));
46     AxisX:= StrToInt(InputBox(IBTitle,IBMessage2,IBDefaultValue));
47     if (AxisY<2) and (AxisX<2) then
48     begin
49         ShowMessage('Значения '+IntToStr(AxisY)+':y или '+IntToStr(AxisX)+':x
50 не коректны (y<2 или x<2)');
51         AxisY:=2;AxisX:=2;
52     end;
53
54 end;

```

```

55
56 procedure CreateMatrix(var Matrix:TMatrix; AxisY,AxisX: integer);
57 begin
58     SetLength(Matrix,AxisY,AxisX);
59 end;
60
61 procedure InputMatrix(var Matrix:TMatrix);
62 const
63     IBTitle = 'Создание матрицы';
64     IBMessage = 'Способ заполнения, Вручную-N, Автоматически-Y';
65     IBDefaultvalue = 'Y';
66 var
67     flag:string;
68 begin
69     flag:= InputBox(IBTitle,IBMessage,IBDefaultvalue);
70
71     if (flag = 'Y')or(flag = 'y')or(flag = ' ')or(flag = '') then
RandomInputMatrix(Matrix)
72     else UserInputMatrix(Matrix);
73 end;
74
75 procedure RandomInputMatrix(var Matrix:TMatrix);
76 var i,j: integer;
77 begin
78     Randomize;
79     for i := 0 to length(Matrix)-1 do
80         for j := 0 to length(Matrix[1])-1 do
81             begin
82                 Matrix[i,j]:= Random(10);
83                 Matrix[i,j]:= Matrix[i,j] + Random;
84                 Matrix[i,j]:= Round(matrix[i,j]*10000)/10000;
85             end;
86         end;
87
88 procedure UserInputMatrix(var Matrix:TMatrix);
89 const
90     IBTitle = 'Создание матрицы(2/2)';
91     IBDefaultvalue = '1.0';
92 var
93     i,j: integer;
94     IBMessage: string;
95 begin
96     for i := 0 to length(Matrix)-1 do
97         for j := 0 to length(Matrix[1])-1 do
98             begin
99                 IBMessage:='Введите matrix['+IntToStr(i)+'']['+IntToStr(j)+''] элемент;';
100                 Matrix[i,j]:=StrToFloat(InputBox(IBTitle,IBMessage,IBDefaultvalue));
101             end;
102         end;
103
104 procedure OpenTextFile(var TF:TextFile);
105 const
106     IBTitle = 'Создание матрицы';
107     IBMessage = 'Способ выбора файла, Путь файла-N, Окно выбора-Y';
108     IBDefaultvalue = 'Y';
109 var
110     s,Path,flag: string;
111     i:integer;

```

```

112 begin
113     flag:= InputBox(IBTitle,IBMessage,IBDefaultValue);
114
115     if (flag = 'Y')or(flag = 'y')or(flag = ' ')or(flag = '') then
116         ODOpenFile(TF)
117     else PathOpenFile(TF);
118 end;
119
120 procedure writeMatrixInTextFile(var TF:TextFile; Matrix:TMatrix);
121 var i,j: integer;
122 begin
123     writeln(TF, ' ');
124     writeln(TF, 'x:= '+IntToStr(length(Matrix[1]))+', y:=
'+IntToStr(length(Matrix)));
125     for i := 0 to length(Matrix)-1 do
126         for j := 0 to length(Matrix[1])-1 do
127             begin
128                 if (j <> length(Matrix[1])-1) then
129                     write(TF,Matrix[i,j]:8:4, ' ')
130                 else
131                     writeln(TF,Matrix[i,j]:8:4)
132                 end;
133             end;
134         end;
135     end;
136
137 procedure writeIntMatrixInTextFile(var TF:TextFile; Matrix:TMatrix);
138 var i,j: integer;
139 begin
140     writeln(TF, ' ');
141     writeln(TF, 'x:= '+IntToStr(length(Matrix[1]))+', y:=
'+IntToStr(length(Matrix)));
142     for i := 0 to length(Matrix)-1 do
143         for j := 0 to length(Matrix[1])-1 do
144             begin
145                 if (j <> length(Matrix[1])-1) then
146                     write(TF,Matrix[i,j]:1:0, ' ')
147                 else
148                     writeln(TF,Matrix[i,j]:1:0)
149                 end;
150             end;
151         end;
152     end;
153
154 procedure ODOpenFile(var TF:TextFile);
155 var
156     Path:string;
157     openDialog : TOpenDialog;
158 begin
159     openDialog:=TOpenDialog.Create(nil);
160     if openDialog.Execute then
161         if FileExists(openDialog.FileName) then
162             Path:=openDialog.FileName
163         else raise Exception.Create('File does not exist.');
```

```

167 procedure PathOpenFile(var TF:TextFile);
168 var Path:string;
169 begin
170   Path:=InputBox('Введите путь','', 'A:\Desktop\');
171   AssignFile(TF,Path);
172   if FileExists(Path) then Append(TF) else Rewrite(TF);
173 end;
174
175
176 procedure ProcessingFlip(var Matrix:TMatrix;y,x:integer);
177 var centerX,centerY,i,j:integer;
178     TempVar:real;
179 begin
180   centerX:=((x-1) div 2);
181   centerY:=((y-1) div 2);
182   for i:=0 to centerY do
183     for j:=0 to centerX do
184       begin
185         TempVar:=Matrix[i,j];
186         Matrix[i,j] :=Matrix[centerY+i+1,centerX+j+1];
187         Matrix[centerY+i+1,centerX+j+1]:=TempVar;
188       end;
189
190   for i:=0 to centerY do
191     for j:=0 to centerX do
192       begin
193         TempVar:=Matrix[centerY+i+1,j];
194         Matrix[centerY+i+1,j] :=Matrix[i,centerX+j+1];
195         Matrix[i,centerX+j+1]:=TempVar;
196       end;
197   end;
198
199 procedure ProcessingSpreading(var Matrix:TMatrix;n:integer);
200 var i,j:integer;
201 begin
202   for i := 0 to length(Matrix)-1 do
203     for j := 0 to length(Matrix[1])-1 do
204       begin
205         if i=j then Matrix[i,j]:=1;
206         if i<j then Matrix[i,j]:=j-i+1;
207         if i>j then Matrix[i,j]:=-j+i+1
208       end;
209   end;
210
211 end.
212

```