

Практическая работа № 13, ОПРЕДЕЛИТЕЛЬ

Постановка задачи

Создайте программу, которая в зависимости от величин N (количество строк и столбцов) создает матрицу размером N×N. Программа предоставляет возможность ввести значения вручную или считать информацию из текстового файла. Результатом выполнения программы является подсчет определителя.

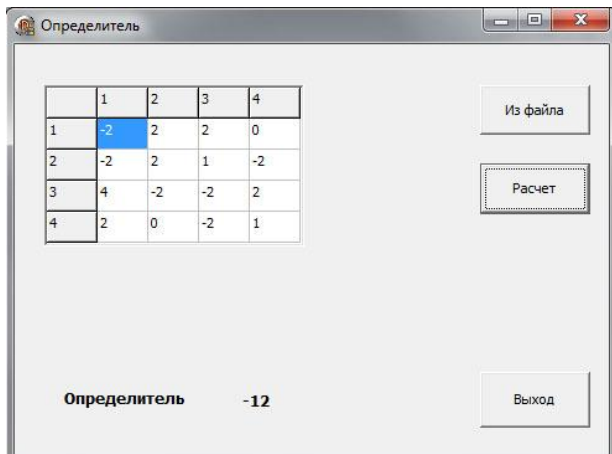


Рис.1

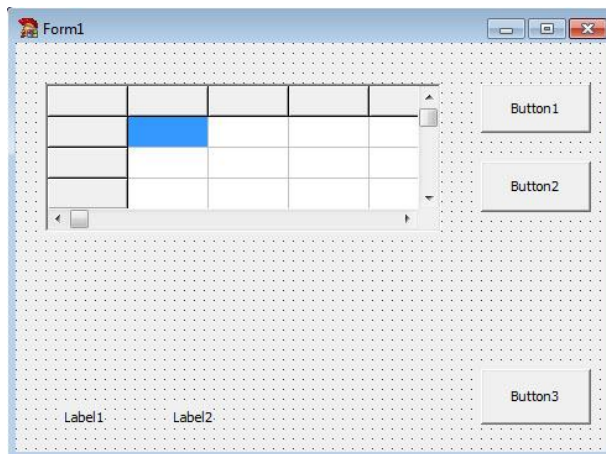


Рис.2

Новым в этой работе являются:

- использование метода Гаусса для вычисления определителя матрицы.

План разработки программы

1. Откройте новый проект.
2. Разместите в форме объекты в соответствии с рис.2.
3. Установите свойства компонент на вкладке **Object Inspector**:

Выделенный объект	Имя свойства	Значение
Button1	Caption	Из файла
Button2	Caption	Расчет
Button3	Caption	Выход
Label1	Caption	Определитель
Label2	Caption	Удалить название объекта
StringGrid1	Name	MATR
Form1	Caption	Определитель

4. Сохраните код программы и проект под именами, например, **Unit_13.pas** и **Pr_13.dpr**.

5. Разместите в блоке реализации после слова **implementation** описание переменных:

```
Const    N=10;
Type     T_Mas=Array[1..N,1..N] Of Real;
Var      Mas      :T_Mas; // Массив
          Kol_N    :Byte; // N-количество строк, столбцов
          Determ   :Real; // Определитель
```

6. Создайте следующие процедуры обработки событий:

Выделенный объект	Имя событие	Текст процедуры
Button3 «Выход»	OnClick	<pre>procedure TForm1.Button3Click(Sender: TObject); begin Close; end;</pre>
Form1	OnCreat	<pre>procedure TForm1.FormCreate(Sender: TObject); MATR.Visible:=False; Button2.Enabled:=False; end;</pre> <p>Комментарий</p> <p>При создании формы установим компонент StringGrid невидимым (новое имя этого компонента MATR), т.к. в начале каков размер матрицы. Кроме этого до определения размера таблицы установим недоступными кнопку «Расчет».</p>

7. Создадим процедуру формирования матрицы с помощью текстового файла **Test.txt**. Информация в этом текстовом файле представлена следующим образом:

1-я строка – размер матрицы

2-я и все последующие строки содержат значения элементов матрицы строки, разделенные не менее одним пробелом.

Пример содержания файла **Test.txt**

```
4
-2 2 2 0
-2 2 1 -2
4 -2 -2 2
2 0 -2 1
```

Для компонента **Button1** («Из файла») перейдите на вкладку **Events Инспектора объектов (Object Inspector)**, найдите событие **OnClick**, справа от него дважды щелкнуть левой кнопкой мыши. Попад в код программы, надо написать следующий код:

```
procedure TForm1.Button1Click(Sender: TObject);
Var I,J :Integer;
    myFile : TextFile;
    text : string;
begin
    Button2.Enabled:=True;
    // Считывание информации из файла
    AssignFile(myFile, 'Test.txt');
    Reset(myFile);
    ReadLn(myFile, Kol_N);
    For I:=1 To Kol_N Do
        begin
            For J:=1 To Kol_N Do
                Read(myFile, Mas[I, J]);
            ReadLn(myFile);
        end;
    CloseFile(myFile);
    // Формирование MATR
```

```
MATR.DefaultColWidth:= 40;  
MATR.RowCount:=Kol_N+1;  
MATR.ColCount:=Kol_N+1;  
MATR.Height:=  
  (MATR.DefaultRowHeight+2) * (Kol_N+1);  
MATR.Width:=  
  (MATR.DefaultColWidth +2) * (Kol_N+1);  
MATR.Visible:=True;  
For I:=1 to Kol_N do  
  MATR.Cells[0,I]:=IntToStr(I);  
For J:=1 to Kol_N do  
  MATR.Cells[J,0]:=IntToStr(j);  
For I:=1 to Kol_N+1 do  
  For J:=1 to Kol_N+1 do  
    MATR.Cells[j,I]:=FloatToStr(Mas[I,J]);  
end;
```

Комментарий

В начале устанавливается доступная кнопка «Расчет», а затем считывается необходимая информация (**Kol_N** – размер матрицы, **Mas[I,J]** – элементы матрицы) из текстового файла **Test.txt**.

На следующем этапе программно задаем свойства компонента **StringGrid (MATR)** и устанавливаем ее видимой. Заполняем фиксированную часть таблицы значениями номеров строк и столбцов и значениями элементов массива **Mas**.

Немного теории

Свойства определителей

1. Определитель, имеющий нулевой столбец (строку), равен нулю.
2. При перестановке местами любых двух столбцов (строк) знак определителя меняется на противоположный.
3. Если элементы строки (столбца) разделить (умножить) на число, то определитель увеличится (уменьшится) на это число.
4. Если к элементам какой-нибудь строки (столбца) прибавить (вычесть) соответствующие элементы другой строки (столбца), то определитель не изменится.
5. Определитель треугольной матрицы равен произведению элементов, стоящих на главной диагонали.

Алгоритм нахождения определителя

Дана матрица $A[n,n]$

$$\begin{pmatrix} A[1,1] & A[1,2] & A[1,3] & \dots & A[1,n] \\ A[2,1] & A[2,2] & A[2,3] & \dots & A[2,n] \\ \dots & & & & \\ A[n,1] & A[n,2] & A[n,3] & \dots & A[n,n] \end{pmatrix}$$

Если $N = 1$, то **det = A[n,n]** → **конец**

1 (к)-ый этап

1. Найдем FMAX – абсолютное максимальное значение в 1 (к) столбце. NSTR – номер строки, в которой находится абсолютное максимальное значение.
2. Если FMAX = 0, то **det = 0** → **конец**
3. NSTR < > 1 (к), то меняем местами 1-ую (к-ую) строку и строку с номером NSTR.
det := det × (-1)
4. Разделить все элементы 1-ой (к-ой) строки на A[1,1] (A[k,k]).

$A[1,1]/A[1,1]$	$A[1,2]/A[1,1]$...	$A[1,n]/A[1,1]$
A[2,1]	A[2,2]	...	A[2,n]
...
A[n,1]	A[n,2]	...	A[n,n]

Получаем:

1	A[1,2]	...	A[1,n]
A[2,1]	A[2,2]	...	A[2,n]
...
A[n,1]	A[n,2]	...	A[n,n]

det := det × A[1, 1] (det := det × A[k, k])

5. Обнуляем все элементы 1-го (к-го) столбца, кроме A[1,1] (A[k,k]), который равен уже 1. Для этого из i -ой строки вычитаем 1-ую (к-ую) строку умноженную на число, равное A[i,1] (A[k,1]) и записываем на место i-ой строки, при этом определитель не изменится.

1	A[1,2]	...	A[1,n]
$A[2,1]-A[2,1]$	$A[2,2]-A[1,2] \times A[2,1]$...	$A[2,n]-A[1,n] \times A[2,1]$
...
$A[n,1]-A[n,1]$	$A[n,2]-A[1,2] \times A[n,1]$...	$A[n,n]-A[1,n] \times A[n,1]$

Получаем:

1	A[1,2]	...	A[1,n]
0	A[2,2]	...	A[2,n]
...
0	A[n,2]	...	A[n,n]

Теперь будем работать с матрицей без 1-ого (к-ого) столбца и 1-ой (к-ой) строки по тому же алгоритму.

1	A[1,2]	...	A[1,n]
0	A[2,2]	...	A[2,n]
...
0	A[n,2]	...	A[n,n]

Всего таких этапов N-1 (**for K:=1 to N-1**), пока не останется матрица из одного элемента A[n,n].

В конце **det := det × A[n,n]**

Пример

Найти определитель матрицы:

$$\begin{pmatrix} -2 & 2 & 2 & 0 \\ -4 & 2 & 1 & -2 \\ 4 & -2 & -2 & 2 \\ 2 & 0 & -2 & 1 \end{pmatrix}$$

1 шаг

-2	2	2	0
-2	2	1	-2
4	-2	-2	2
2	0	-2	1

i=1
FMax=4, NSTR=3

Меняем местами 1-ю и 3-ю строки.

Знак определителя меняется на противоположный.

4	-2	-2	2
-2	2	1	-2
-2	2	2	0
2	0	-2	1

Det := Det × (-1)

Все элементы 1-ой строки делим на FMax (A[1,1]).

Значение определителя умножаем на FMax.

4/4	-2/4	-2/4	2/4
-2	2	1	-2
-2	2	2	0
2	0	-2	1

Det := Det × (4) = -4

В результате получаем:

1	-0.50	-0.50	0.50
-2	2	1	-2
-2	2	2	0
2	0	-2	1

Обнуляем все элементы 1-го столбца:

1	-0.50	-0.50	0.50
-2+2	2-(-2×-0.50)	1-(-2×-0.50)	-2-(-2×0.50)
-2+2	2-(-2×-0.50)	2-(-2×-0.50)	0-(-2×0.50)
2-2	0-(2×-0.50)	-2-(2×-0.50)	1-(2×0.50)

В результате получаем:

1	-0.50	-0.50	0.50
0	1	0	-1
0	1	1	1
0	1	-1	0

2 шаг

1	-0.50	-0.50	0.50
0	1	0	-1
0	1	1	1
0	1	-1	0

$i=2$

$F_{Max}=1, NSTR=2$

1	-0.50	-0.50	0.50
0	1	0	-1
0	1-1	1-0×1	1-(1×-1)
0	1-1	-1-0×1	0-(1×-1)

$Det:=Det \times (1)=-4$

3 шаг

1	-0.50	-0.50	0.50
0	1	0	-1
0	0	1	2
0	0	-1	1

$i=3$

$F_{Max}=1, NSTR=3$

1	-0.50	-0.50	0.50
0	1	0	-1
0	0	1	2
0	0	-1+1	1-(-1×2)

4 шаг

1	-0.50	-0.50	0.50
0	1	0	-1
0	0	1	2
0	0	0	3

$Det:=Det \times A[4,4]=-4 \times 3=-12$

Пояснения к программному коду

Описанный алгоритм можно представить как повторяющийся **к-ый этап**, который состоит из основных блоков:

<p>Найдем FMAX – абсолютное максимальное значение в k-ом столбце. NSTR – номер строки, в которой находится абсолютное максимальное значение.</p>	<pre>Fmax:=A[K,K]; NSTR:=K; for I:=K+1 to N do If ABS(A[I,K])>ABS(Fmax) then begin Fmax:=A[I,K];NSTR:=I; end;</pre>
<p>Если FMAX = 0, то det = 0 → конец</p>	<pre>If Fmax=0.0 then begin Det:=0; Exit; end;</pre>
<p>NSTR <> k, то меняем местами k-ую строку и строку с номером NSTR. det:=det×(-1)</p>	<pre>If NSTR<>K Then begin For I:=1 to N do begin R:=A[K,I]; A[K,I]:=A[NSTR,I]; A[NSTR,I]:=R; end; Det:= -Det; end;</pre>
<p>Разделить все элементы k-ой строки на A[k,k].</p>	<pre>For J:=K to N do A[K,J]:=A[K,J]/Fmax; For I:=K+1 to N do begin R:=A[I,K]; For J:=K to N do A[I,J]:=A[I,J]-A[K,J]*R; end; Det:=Det*FMax;</pre>

Всего таких этапов N-1 (**for K:=1 to N-1**), пока не останется матрица из одного элемента A[n,n].

В конце **det := det × A[n,n]**

8. Для расчета определителя создадим самостоятельную процедуру, которая может выглядеть так:

```
Procedure DeterminantMatrix(A:T_Mas; N:integer; Var Det:Real);
var
  K,I,J, L,M,NSTR:integer;
  R, Fmax:Real;
Begin
  Det:=1;
  for K:=1 to N-1 do
    begin
      //FMAX - абсолютное максимальное значение в k-ом столбце
      //NSTR - номеру строки этого элемента
      Fmax:=A[K,K]; NSTR:=K;
      for I:=K+1 to N do
        If ABS(A[I,K])>ABS(Fmax) then
          begin
```

```
      Fmax:=A[I,K];NSTR:=I;
    end;
  If Fmax=0.0 then
    begin Det:=0; Exit; end;
  //Меняем местами строки
  If NSTR<>K Then
    begin
      Det:=-Det;
      For I:=1 to N do
        begin
          R:=A[K,I]; A[K,I]:=A[NSTR,I]; A[NSTR,I]:=R;
        end;
      end;
  //Разделить все элементы k-ой строки на A[k,k]
  For J:=K to N do A[K,J]:=A[K,J]/Fmax;

  For I:=K+1 to N do
  begin
    R:=A[I,K];
    For J:=K to N do A[I,J]:=A[I,J]-A[K,J]*R;
  end;
  Det:=Det*FMax
end;
Det:=Det*A[N,N];
end;
```

9. Для компонента **Button2** («Расчет») перейдите на вкладку **Events Инспектора объектов (Object Inspector)**, найдите событие **OnClick**, справа от него дважды щелкните левой кнопкой мыши. Попад в код программы, надо написать следующий код:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  DeterminantMatrix(Mas,Kol_N,Determ);
  Label2.Caption:=FloatToStr(Determ);
end;
```

10. Сохраните проект окончательно, запустите и протестируйте его.

Задание для самостоятельного выполнения

1. Дополните программу возможностью выбора текстового файла, который содержит информацию о матрице.
2. Дополните программу возможностью ввода информации вручную (см. Практическую работу №11 «МАТРИЦА»).