

Практическая работа № 9, МАТРИЦА

Постановка задачи

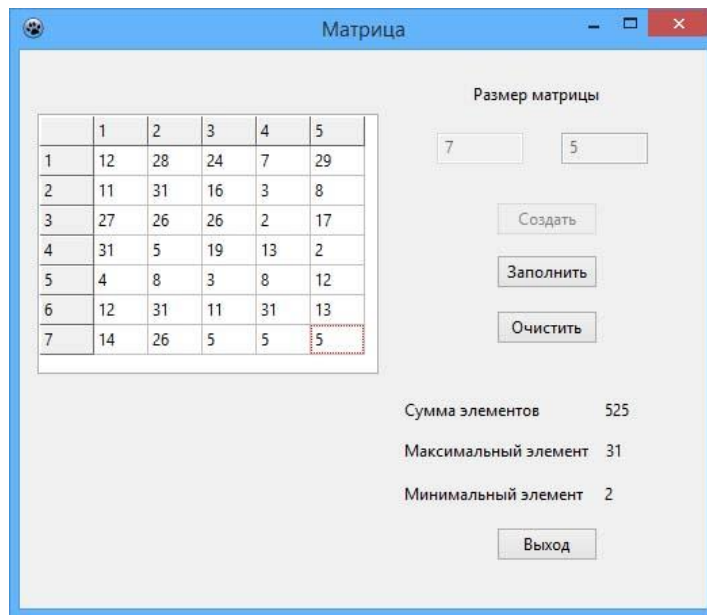


Рис.9.1

Создать программу, которая в зависимости от величин N (количество строк) и M (количества столбцов) формирует матрицу размером $N \times M$, размер которой будет определен после заполнения полей «Размер матрицы».

Программа предоставляет возможность заполнить матрицу с помощью случайных чисел или ввести значения вручную. Кроме этого можно подсчитать сумму элементов матрицы, определить максимальный и минимальный элементы матрицы.

Новым в этой работе являются:

- компонент таблица строк **StringGrid** вкладки палитры компонентов **Additional**.

План разработки программы

1. Откройте новый проект.
2. Разместите в форме объекты в соответствии с рис.9.2.

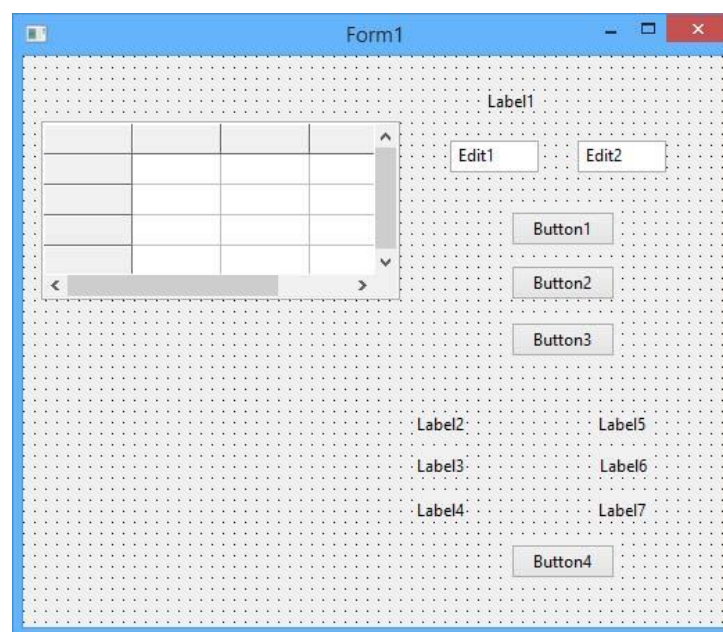


Рис.9.2

3. Установите свойства компоненты на вкладке **Object Inspector**:

Выделенный объект	Имя свойства	Значение
StringGrid1	Name	MATR

4. Первоначальные свойства других компонент зададим непосредственно в программе. Для этого создадим процедуру обработки событий при создании формы Form1.

Выделенный объект	Вкладка окна Инспектор объектов	Имя свойства/ Имя события	Значение/Действие
Form1	Events	OnCreat	<pre> procedure TForm1.FormCreate(Sender: TObject); begin Label1.Caption:='Размер матрицы'; Label2.Caption:='Сумма элементов'; Label3.Caption:='Максимальный элемент'; Label4.Caption:='Минимальный элемент'; Label5.Caption:=''; Label6.Caption:=''; Label7.Caption:=''; Button1.Caption:='Создать'; Button2.Caption:='Заполнить'; Button3.Caption:='Очистить'; Button4.Caption:='Выход'; Edit1.Text:=''; Edit2.Text:=''; Form1.Caption:='Матрица'; end; </pre>

5. Сохраните код программы и проект под именами, например, **fMain.pas** и **Matrix.lpr**.

Немного теории

Компонента **StringGrid** (вкладка палитры компонентов **Additional**) предназначен для создания таблицы.

Свойству компоненты **Cells** соответствует двухмерный массив ячеек, каждая из которых может содержать произвольный текст. Содержание ячейки массива, находящегося на пересечении строки с номером **Row** и столбца с номером **Col** определяется элементом **StringGrid1.Cells [Col, Row]**. Содержимое ячейки можно редактировать.

Двумерный массив состоит из двух частей: фиксированной и рабочей.

Фиксированная часть служит для показа заголовков строк (столбцов) и для ручного управления их размерами. Обычно фиксированная часть занимает крайний левый столбец и самую верхнюю строку таблицы. С помощью свойств **FixedCols** и **FixedRows** можно задать другое количество фиксированных столбцов и строк. Если свойства **FixedCols** и **FixedRows** имеют значение 0, то таблица не содержит фиксированной зоны.

Рабочая часть – это остальная часть таблицы. Она может содержать произвольное количество строк и столбцов, которое можно изменять в ходе выполнения программы.

Если рабочая часть не умещается целиком в пределах окна компоненты, то автоматически появляются полосы прокрутки. При прокрутке рабочей области фиксированная область не исчезает.

В таблице приведен перечень часто используемых свойств компоненты **StringGrid**.

ColCount	Количество столбцов таблицы
RowCount	Количество строк таблицы
FixedCols	Количество столбцов фиксированной зоны
FixedRows	Количество строк фиксированной зоны
DefaultRowHeight	Высота строки таблицы
Height	Высота всей таблицы
DefaultColWidth	Ширина столбца таблицы
Width	Ширина всей таблицы
GridLineWidth	Ширина линий, ограничивающих ячейки таблицы
Font	Шрифт, используемый для отображения содержимого ячеек таблицы
Options.goAlwaysShowEditor	Разрешает (True) редактировать ячейку сразу при выделении. Если (False), то для редактирования нужно нажать Enter или F2.
Options.goEditing	Признак допустимости редактирования содержимого ячеек таблицы. True – редактирование разрешено, False – запрещено
Options.goTabs	Разрешает (True) или запрещает (False) использование клавиши «Tab» для перемещения курсора в следующую ячейку таблицы

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

```
Const R = 10;           //Максимальный размер массива
Type T_Mas = Array[1..R,1..R] of integer; //Тип двумерного массива
Var
    N, M: Integer;     //N-количество строк, M-количество столбцов
    MATR_MAX,          //Значение максимального элемента таблицы
    MATR_MIN,          //Значение минимального элемента таблицы
    MATR_SUM: Integer; //Значение суммы элементов таблицы
    MAS: T_Mas;        //Рабочий массив
```

7. Основная задача проекта – создать таблицу, размер которой будет определен после заполнения полей **Edit1** и **Edit2**.

Создадим следующие процедуры обработки событий:

Выделенный объект	Имя событие	Текст процедуры
Button4 «Выход»	OnClick	<pre>procedure TForm1.Button4Click(Sender:TObject); begin Close; end;</pre>
Form1	OnCreat (дополнит ь уже существую щую)	<pre>MATR.Visible:=False; Button2.Enabled:=False; Button3.Enabled:=False;</pre> <p>Комментарий При создании формы установим компоненту StringGrid невидимой (новое имя этой компоненты MATR), т.к. в начале неизвестно сколько же строк и столбцов в таблице. Кроме этого до определения размера таблицы установим недоступными кнопки «Заполнить» и «Очистить».</p>
Button1 «Создать таблицу»	OnClick	<pre>procedure TForm1.Button1Click(Sender:TObject); Var I,J:Byte; begin If (Edit1.Text<>'') and (Edit2.Text<>'') Then begin Edit1.Enabled:=False; Edit2.Enabled:=False; Button1.Enabled:=False; Button2.Enabled:=True; Button3.Enabled:=True; N:=StrToInt(Edit1.Text); M:=StrToInt(Edit2.Text); MATR.DefaultColWidth:= 40; MATR.RowCount:=N+1; MATR.ColCount:=M+1; MATR.Height:=(MATR.DefaultRowHeight+2) * (N+1); MATR.Width:= (MATR.DefaultColWidth +2) * (M+1); MATR.Visible:=True; For I:=1 to N do MATR.Cells[0,I]:=IntToStr(I); For J:=1 to M do MATR.Cells[J,0]:=IntToStr(j); end; //Настройка свойства Options With MATR do Options:=Options+[goEditing]+ [goAlwaysShowEditor]+[goTabs]; end;</pre> <p>Комментарий Данная процедура будет выполняться только при условии, что введены размеры матрицы, т.е Edit1.Text и Edit2.Text не являются «пустым символом». В начале устанавливаются недоступными компоненты ввода, определяющие размер таблицы, и кнопка «Создать», а недоступные ранее кнопки «Заполнить» и «Очистить» становятся доступными. После этого переходим к формированию таблицы. Для этого устанавливаем значения переменных N и M, определяющих количество строк и столбцов таблицы –</p>

	<p>переводим из текстовой информации значения полей Edit1 и Edit2 в числовые значения. После этого программно задаем свойства компоненты StringGrid (MATR) и устанавливаем ее видимой. В конце заполняем фиксированную часть таблицы значениями номеров строк и столбцов.</p> <p>Последний оператор With позволяет внести программно изменения в свойства компоненты StringGrid (MATR) goEditing, goAlwaysShowEditor и goTabs установив их значения в True.</p>
<p>Button3 «Очистить таблицу»</p>	<pre> procedure TForm1.Button3Click(Sender:TObject); Var I,J:Byte; begin For I:=1 to N do For J:=1 to M do MATR.Cells[J,I]:= ''; Label5.Caption:= ''; Label6.Caption:= ''; Label7.Caption:= ''; Edit1.Text:= ''; Edit2.Text:= ''; Matr.Visible:=False; Button2.Enabled:=False; Button3.Enabled:=False; Button1.Enabled:=True;; Edit1.Enabled:=True; Edit2.Enabled:=True; end; </pre> <p>Комментарий</p> <p>Все ячейки таблицы заполняется пустой строкой, а затем очищаются компоненты Label5, Label6, Label7, Edit1.Text, Edit2.Text. Компонента StringGrid (MATR) устанавливается невидимой, кнопки «Заполнить» и «Очистить» становятся недоступными, кнопка «Создать» и компоненты Edit1 и Edit2 – доступными.</p>

8. Сохраните полученные изменения и проверьте правильность выполнения программы.

9. Для дальнейшей работы создадим три процедуры: заполнение рабочего массива нулями, заполнение рабочего массива случайными числами и подсчет суммы элементов массива, поиска максимального и минимального элемента.

```

procedure ZAPOLN_0(N,M:Integer; Var MAS:T_MAS);
// заполнение рабочего массива нулями
Var I,J : Byte;
begin
For I:=1 to N do
For J:=1 to M do
MAS[I,J]:=0;
end;

procedure ZAPOLN_1(N,M:Integer; Var MAS:T_MAS);
// заполнение рабочего массива случайными числами
Var I,J : Byte;
begin
Randomize;
For I:=1 to N do
For J:=1 to M do
MAS[I,J]:=Random(N*M);
end;
    
```

```

procedure MAS_SUM
(N,M:Integer; MAS:T_MAS; Var MATR_MAX,MATR_MIN,MATR_SUM:Integer) ;
// подсчет суммы элементов массива, поиска максимального и
минимального элемента
Var I,J : Byte;
begin
MATR_MAX:= MAS[1,1];
MATR_MIN:= MAS[1,1];
MATR_SUM:=0;
For I:=1 to N do
  For J:=1 to M do
    begin
      MATR_SUM:=MATR_SUM+MAS[I,J];
      IF MAS[I,J]>MATR_MAX Then MATR_MAX:=MAS[I,J];
      IF MAS[I,J]<MATR_MIN Then MATR_MIN:=MAS[I,J];
    end;
  end;
end;
    
```

Разместите процедуры в блоке реализации после {TForm1}.

10. Продолжим разработку программы и создадим следующую процедуру обработки событий:

Выделенный объект	Имя событие	Текст процедуры
Button2 «Заполнить таблицу»	OnClick	<pre> procedure TForm1.Button2Click(Sender: TObject); Var I,J:Byte; begin ZAPOLN_1(N,M,MAS); For I:=1 to N do For J:=1 to M do MATR.Cells[J,I]:=IntToStr(MAS[I,J]); MAS_SUM(N,M,MAS,MATR_MAX,MATR_MIN,MATR_SUM); Label5.Caption:=IntToStr(MATR_SUM); Label6.Caption:=IntToStr(MATR_MAX); Label7.Caption:=IntToStr(MATR_MIN); end; </pre> <p>Комментарий</p> <p>В начале обращаемся к процедуре ZAPOLN_1, в которой элементы рабочего массива MAS заполняются случайными числами в диапазоне от 0 до (N*M-1), после этого элементы массива переписываются ячейки компоненты StingGrid.</p> <p>Затем обращаемся к процедуре MAS_SUM, с помощью которой определяем максимальное и минимальное числа и сумма элементов таблицы.</p> <p>В конце полученные значения выводятся на экран. Для этого используются компоненты Label5, Label6, Label7.</p>

11. Программа готова. Но если в поля ввода «Размерность матрицы» мы по ошибке вместо числа введем букву или какой-либо другой символ, то программа будет прервана. Для того чтобы это избежать создадим процедуры обработки события **KeyPress** для компоненты **Edit1**.

```

procedure TForm1.Edit1KeyPress(Sender: TObject; var Key:Char);
    
```

```
begin
case Key of
  '0'..'9': ;
  #8:      ;
  #13: Edit2.SetFocus;
  else Key := Chr(0);
end;
end;
```

Комментарий

В зависимости от нажатой клавиши программа выполнит следующие действия:

- нажата любая цифровая клавиша или клавиша «**Back Space**» (код клавиши #8) – программа ничего не изменит,
- нажата клавиша «**Enter**» (код клавиши #13) – курсор перейдет в окно редактора ввода **Edit2**,
- во всех остальных случаях – введенный символ будет удален (присвоение значения пустого символа Chr(0)).

Для компоненты **Edit2** самостоятельно создайте процедуру, обрабатывающую ввод. Она будет отличаться тем, что при нажатии клавиши «**Enter**», курсор должен перейти на кнопку «Создать таблицу» (**Button1**).

12. Созданная программа не позволяет редактировать элементы таблицы. Внесем изменения в свойства компонента **StringGrid (MATR)**. Это можно сделать двумя способами:

- на вкладке **Object Inspector** свойству **Options.goEditing** установить значение **True**;
- программным способом. Для этого в уже созданную процедуру **TForm1.Button1Click** добавить оператор:

```
with MATR do
```

```
Options:=Options+[goEditing]+[goAlwaysShowEditor]+[goTabs];
```

Процедура, обрабатывающая нажатие клавиши в поле компонента **StringGrid (MATR)** будет выглядеть так:

```
procedure TForm1.MatrKeyPress(Sender: TObject; var Key: char);
  Var I,J:Byte;
begin
  case Key of
    '0'..'9': ;
    #8: MATR.Cells[MATR.COL,MATR.ROW] := ''; //BackSpace
    else Key:=Chr(0);
  end;
  For I:=1 to N do
    For J:=1 to M do
      If MATR.Cells[MATR.COL,MATR.ROW] <> '' Then
        MAS[I,J] := StrToInt(MATR.Cells[J,I]);
  MAS_SUM(N,M,MAS,MATR_MAX,MATR_MIN,MATR_SUM);
  Label5.Caption:=IntToStr(MATR_SUM);
```

```
Label6.Caption:=IntToStr (MATR_MAX) ;
Label7.Caption:=IntToStr (MATR_MIN) ;
end;
```

Комментарий

Обработка нажатия допустимой клавиши рассмотрена в п.11. После внесения изменений переписывается информация рабочий массив MAS.

Дальше идет обращение к процедуре подсчета максимального, минимального элемента и суммы элементов в таблице, после внесения изменений.

13. Сохраните проект окончательно, запустите и протестируйте его.

Задание для самостоятельного выполнения

1. Дополните программу, вставив блок определения суммы по каждому столбцу матрицы.

Подсказка. Необходима еще одна таблица (компонент **StringGrid**), в которой будут находиться подсчитанные суммы по столбцам. Формировать эту таблицу нужно в момент формирования основной таблицы. Для этой таблицы необходимо установить свойства **FixedCols** и **FixedRows** равные 0, т.е. таблица не будет содержать фиксированные зоны, при этом нумерация колонок и строк будет начинаться с 0.

Продолжение работы

Доработать программу так, чтобы можно было бы заполнять матрицу различными способами из предложенных или каким-то своим способом:

1 способ

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

2 способ

1	5	9	13	17
2	6	10	14	18
3	7	11	15	19
4	8	12	16	20

3 способ

1	2	3	4	5
10	9	8	7	6
11	12	13	14	15
20	19	18	17	16

4 способ

16	17	18	19	20
15	14	13	12	11
6	7	8	9	10
5	4	3	2	1

Для выбора способа заполнения можно воспользоваться компонентой **ListBox** (см. Практическая работа №4, СПРАВОЧНИК).

Для заполнения выделите объект **Listbox1**, найдите свойство **Items**, щелкните на кнопке с тремя точками, расположенной справа от него. В появившемся окне встроенного редактора **String List Editor** введите названия способов заполнения, каждый на новой строке. Например:

Случайными числами По горизонтали По вертикали Змейкой слева Змейкой справа Зигзагом Спиралью

Комментарий

- а) Свойство **Items** содержит элементы списка.
- б) Список может быть создан при создании формы или во время работы программы.
- в) Свойство **ItemIndex** определяет номер элемента, выбранного из списка. Первый элемент имеет номер 0. Если не выбран ни один из элементов, то значение свойства **ItemIndex** равно – «-1».

Выполните следующие действия:

Выделенный объект	Вкладка окна Object Inspector	Имя свойства/ Имя события	Значение/Действие
Listbox1	Events	OnKeyPress	<pre> if key=#13 then Case Listbox1.ItemIndex of 0: ZAPOLN_1 (N,M,MAS) ; 1: ZAPOLN_2 (N,M,MAS) ; 2: ZAPOLN_3 (N,M,MAS) ; 3: ZAPOLN_4 (N,M,MAS) ; 4: ZAPOLN_5 (N,M,MAS) ; 5: ZAPOLN_6 (N,M,MAS) ; 6: ZAPOLN_7 (N,M,MAS) ; end; For I:=1 to N do For J:=1 to M do MATR.Cells[J,I]:=IntToStr(MAS[I,J]); </pre>