

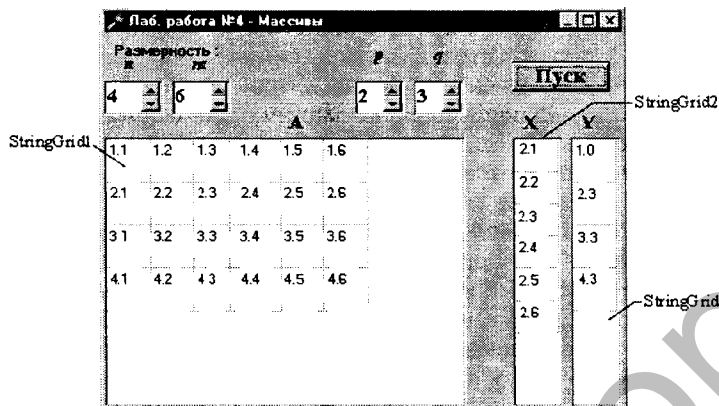
Программирование алгоритмов с использованием массивов

Цель этой статьи — освоить применение компонента *StringGrid* и создать приложение, в котором используются массивы.

1. Пример создания приложения

Необходимо создать Windows-приложение для вычисления вектора $x = \{x_1, x_2, \dots, x_m\}$, равного p -й строке матрицы $A = \{a_{ij}\} (x_j = a_{pj}, j = 1, 2, \dots, m)$; и вектора $y = \{y_1, y_2, \dots, y_n\}$, равного q -му столбцу матрицы $A = \{a_{ij}\} (y_i = a_{iq}, i = 1, 2, \dots, n)$ ($n \leq 6, m \leq 8$). В панели интерфейса следует предусмотреть возможность управления размерностью массивов.

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рисунке.



1.1. Размещение компонентов на Форме

При работе с массивами ввод и вывод информации на экран удобно организовывать с помощью компонента *StringGrid*.

Компонент *StringGrid* используется для отображения информации в виде таблицы. Таблица содержит две зоны — фиксированную и рабочую. Фиксированная зона служит для вывода наименований строк и столбцов рабочей зоны и управления их размерами с помощью мыши. Она выделена другим цветом, и в нее запрещен ввод информации с клавиатуры. Количество строк и столбцов фиксированной зоны устанавливается в свойствах *FixedRows* и *FixedCols* соответственно.

Рабочая зона содержит *RowCount* строк и *ColCount* столбцов информации, которую можно изменять как программно, так и с помощью мыши или клавиатуры.

Доступ к информации в программе осуществляется с помощью свойства *Cells[ACol, ARow: integer]: string*, где *ACol* — номер столбца, а *ARow* — номер строки таблицы, причем нумерация начинается с нуля.

Пиктограмма компонента *StringGrid* находится на странице *Additional* Палитры Компонентов. Так как в нашем случае для всех компонентов *StringGrid* фиксированная зона не используется, в Инспекторе Объектов значения свойства *FixedCols* и *FixedRows* следует установить равными 0. Установим предельные значения количества строк n и столбцов m для компонента *StringGrid1* — *ColCount*=8, а *RowCount*=6 (восемь стол-

бцов и шесть строк). Для компонента *StringGrid2* — *ColCount*=1, *RowCount*=8, а для компонента *StringGrid3* — *ColCount*=1, *RowCount*=6.

По умолчанию в компонент *StringGrid* запрещен ввод информации с клавиатуры, поэтому для компонента *StringGrid1* необходимо в Инспекторе Объектов дважды щелкнуть мышью на символе "+" свойства *+Options*, и в открывшемся списке опций установить значение *goEditing* в *True*.

Для удобства работы с компонентами *SpinEdit* установим для компонента *SpinEdit1* следующие значения свойств — *MinValue*=1, *MaxValue*=6, а для компонента *SpinEdit2* — *MinValue*=1, *MaxValue*=8.

1.2. Создание процедур обработки событий *SpinEdit1Change* и *SpinEdit2Change*

События *SpinEdit1Change* и *SpinEdit2Change* возникают при любом изменении значения в поле редактора *SpinEdit1* и *SpinEdit2* соответственно. Создадим процедуры обработки этих событий, в которых присвоим значения n и m , полученные из полей редакторов *SpinEdit*, свойствам *ColCount* и *RowCount* компонентов *StringGrid*. Это позволит управлять размерами таблиц *StringGrid* с помощью компонентов *SpinEdit* без дополнительных кнопок, так как изменение значений в поле редактора *SpinEdit* сразу приведет к изменению размера таблиц *StringGrid*.

Дважды щелкните мышью на компоненте *SpinEdit1* — курсор установится в тексте процедуры-обработчика события *SpinEdit1Change*: **procedure TForm1.SpinEdit1Change(Sender: TObject)**. Внимательно наберите операторы этой процедуры, используя текст модуля *UnMas*. Аналогичным образом создайте процедуру-обработчик события *SpinEdit2Change*: **procedure TForm1.SpinEdit2Change(Sender: TObject)**.

1.3. Текст модуля *UnMas*

```
Unit UnMas;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Spin, Grids;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    SpinEdit1: TSpinEdit;
    SpinEdit2: TSpinEdit;
    Label8: TLabel;
    StringGrid1: TStringGrid;
    StringGrid2: TStringGrid;
    StringGrid3: TStringGrid;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    SpinEdit3: TSpinEdit;
    SpinEdit4: TSpinEdit;
    Label6: TLabel;
    Label7: TLabel;
    Button1: TButton;
  procedure FormCreate(Sender: TObject);
  procedure SpinEdit1Change(Sender: TObject);
```



```

procedure SpinEdit2Change(Sender: TObject);
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.DFM}
var
  A:array[1..6,1..8] of extended; // объявление двумерного массива A
  X:array[1..8] of extended; // объявление одномерного массива X
  Y:array[1..6] of extended; // объявление одномерного массива Y
  n,m,p,q:integer; // объявление глобальных переменных
procedure TForm1.FormCreate(Sender: TObject);
begin
  SpinEdit1.Text:='4'; // начальное значение n
  SpinEdit2.Text:='6'; // начальное значение m
  SpinEdit3.Text:='2'; // начальное значение p
  SpinEdit4.Text:='3'; // начальное значение q
  StringGrid1.RowCount:=4; // количество строк массива A
  StringGrid1.ColCount:=6; // количество столбцов массива A
  StringGrid2.RowCount:=6; // количество строк массива X
  StringGrid3.RowCount:=4; // количество строк массива Y
end;
procedure TForm1.SpinEdit1Change(Sender: TObject);
begin
  n:=StrToInt(SpinEdit1.Text); // n присваивается содержимое
  // поля редактора
  StringGrid1.RowCount:=n; // устанавливается количество
  // строк массива A
  StringGrid3.RowCount:=n; // устанавливается количество
  // строк массива Y
end;
procedure TForm1.SpinEdit2Change(Sender: TObject);
begin
  m:=StrToInt(SpinEdit2.Text); // m присваивается содержимое
  // поля редактора
  StringGrid1.ColCount:=m; // устанавливается количество
  // столбцов массива A
  StringGrid2.RowCount:=m; // устанавливается количество
  // строк массива X
end;
procedure TForm1.Button1Click(Sender: TObject);
var
  i,j:integer; // объявление локальных переменных
begin
  n:=StrToInt(SpinEdit1.Text);
  StringGrid1.RowCount:=n;
  StringGrid3.RowCount:=n;
  m:=StrToInt(SpinEdit2.Text);
  StringGrid1.ColCount:=m;
  StringGrid2.RowCount:=m;
  p:=StrToInt(SpinEdit3.Text);
  q:=StrToInt(SpinEdit4.Text);
  // ввод значений из таблицы в массив A
  for i:=1 to n do
    for j:=1 to m do
      A[i,j]:=StrToFloat(StringGrid1.Cells[j-1,i-1]);
    for j:=1 to m do // формирование массива X и вывод его
      // значений в таблицу
    begin
      X[j]:=A[p,j];
      StringGrid2.Cells[0,j-1]:=FloatToStrF(X[j],ffFixed,3,1);
    end;
  for i:=1 to n do // формирование массива Y
    // и вывод его значений в таблицу
  begin
    Y[i]:=A[i,q];
    StringGrid3.Cells[0,i-1]:=FloatToStrF(Y[i],ffFixed,3,1);
  end;
end;
end.

```

1.4. Работа с приложением

Теперь можно запустить созданное приложение, занести числовые значения в элементы матрицы A и убедиться в том, что приложение функционирует.

2. Индивидуальные задания

Выберите себе индивидуальное задание из приведенных ниже. В соответствии с выбранным заданием создайте приложение и протестируйте его работу.

1. Задана целочисленная матрица A размером $N \times M$. Получить массив B , присвоив его k -му элементу значение 0 , если все элементы k -го столбца матрицы нулевые, и значение 1 в противном случае ($k=1,2,\dots,M$).

2. Задана целочисленная матрица A размером $N \times M$. Получить массив B , присвоив его k -му элементу значение 1 , если элементы k -й строки матрицы упорядочены по убыванию, и значение 0 в противном случае ($k=1,2,\dots,N$).

3. Задана целочисленная матрица A размером $N \times M$. Получить массив B , присвоив его k -му элементу значение 1 , если k -я строка матрицы симметрична, и значение 0 в противном случае ($k=1,2,\dots,N$).

4. Задана целочисленная матрица размером $N \times M$. Определить k — количество "особых" элементов матрицы, считая элемент "особым", если он больше суммы остальных элементов своего столбца.

5. Задана целочисленная матрица размером $N \times M$. Определить k — количество "особых" элементов матрицы, считая элемент "особым", если в его строке слева от него находятся элементы, меньшие его, а справа — большие.

6. Задана символьная матрица размером $N \times M$. Определить k — количество различных элементов матрицы (т.е. повторяющиеся элементы считать один раз).

7. Дана вещественная матрица размером $N \times M$. Упорядочить ее строки по убыванию их первых элементов.

8. Дана вещественная матрица размером $N \times M$. Упорядочить ее строки по убыванию суммы их элементов.

9. Дана вещественная матрица размером $N \times M$. Упорядочить ее строки по убыванию их наибольших элементов.

10. Определить, является ли заданная квадратная матрица n -го порядка симметричной относительно побочной диагонали.

11. Для заданной целой матрицы размером $N \times M$ вывести на экран все ее седловые точки. Элемент матрицы называется седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце или, наоборот, является наибольшим в своей строке и наименьшим в своем столбце.

12. В матрице n -го порядка переставить строки так, чтобы на главной диагонали матрицы были расположены элементы, наибольшие по абсолютной величине.

Литература

- В.В.Феофанов. Delphi 3. Учебный курс. — М.: Нолидж, 1981.
- Э.Возневич. Delphi. Освой самостоятельно. — М.: Восточная книжная компания, 1996.
- Дж.Матчо, Д.Р.Фолкнер. Delphi. — М.: БИНОМ, 1995.
- М.Канту. Delphi 2 для Windows 95/NT. — М.: ООО "Малип", 1997.