

Министерство образования и науки РФ  
ФГБПОУ ВПО Тульский государственный университет  
КАФЕДРА АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

**ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ МАССИВОВ.  
ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ.**

Лабораторная работа № 3  
по курсу «Программирование на ЯВУ»

Вариант № 4

Выполнил: студент группы 220601

\_\_\_\_\_ Белым А.А.  
(подпись)

Проверил: к. ф.-м. н., доцент

\_\_\_\_\_ Сулимова В.В.  
(подпись)

Тула 2011

## **Цель работы**

Цель работы заключается в том, чтобы изучить свойства компонента TStringGrid и написать программу с использованием массивов.

## **Задание на работу**

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

[illegible]



[illegible]

### **Схема алгоритма**

На рисунке ?? представлена схема общего алгоритма расчета количества “особых чисел” в массиве, т.е. таких, которые больше суммы остальных элементов столбца. На рисунке ?? представлена схема алгоритма ввода данных для расчета количества “особых чисел”. На рисунке ?? представлена схема алгоритма расчета количества “особых чисел”. На рисунке ?? представлена схема алгоритма вывода количества “особых чисел” на экран.

## Инструкция пользователю

Программа позволяет посчитать количество “особых” чисел в заданном массиве. Число считается таковым, если оно больше суммы остальных элементов своего столбца.

Для работы заполните таблицу, в которой строки соответствуют строкам массива. После заполнения таблицы нажмите кнопку “Посчитать”. Если в данных были допущены ошибки, программа сообщит об этом во всплывающем окне.

Если данные введены правильно, то программа выведет количество “особых” чисел в правом нижнем углу окна.

## Инструкция программиста

В программе объявлен тип `DDArray=Array of array of Double` - динамический двумерный массив. При разработке программы построение интерполяционного многочлена Ньютона с разделенными разностями были написаны следующие процедуры и функции:

1. Процедура `btnRunClick` - основная процедура программы расчета кол-ва особых значений.

```
procedure TfrmMain.btnRunClick(Sender: TObject);
```

Параметры процедуры представлены в таблице 1 :

Таблица 1 - Параметры основной процедуры программы

имя	тип	предназначение
Sender	TObject	объект-возбудитель события

Локальные переменные представлены в таблице 2 :

Таблица 2 - Локальные переменные основной процедуры программы

имя	тип	предназначение
a	DDArray	массив для поиска особых элементов
k	integer	кол-во особых чисел

2. Функция `Input` - вводит данные для поиска количества особых чисел матрицы.

Возвращает True, если операция прошла успешно, иначе возвращает False.

```
function TfrmMain.Input(var a:DDArray):boolean;
```

Параметры функции представлены в таблице 3 :

Таблица 3 - Параметры функции ввода данных

имя	тип	предназначение
a	DDArray	массив для поиска особых элементов

Локальные переменные функции представлены в таблице 4 :

Таблица 4 - Локальные переменные функции ввода данных

имя	тип	предназначение
i,j	integer	переменные-счетчики для доступа к элементам массива,
n,m	integer	кол-во строк и столбцов массива
errors	string	список информации о возникших ошибках I/O.

3. Функция GetSpec для расчета количества “особых чисел” в массиве, т.е. таких, которые больше суммы остальных элементов столбца.

Возвращает количество “особых” чисел.

```
function GetSpec(const a:DDArray):Integer;
```

Параметры функции представлены в таблице 5 :

Таблица 5 - Параметры функции расчета количества “особых чисел”

имя	тип	предназначение
a	DDArray	массив для поиска особых элементов

Локальные переменные функции представлены в таблице 6 :



Таблица 6 - Локальные переменные функции расчета количества “особых чисел”

<b>имя</b>	<b>тип</b>	<b>предназначение</b>
i,j	integer	переменные-счетчики для доступа к элементам массива,
n,m	integer	кол-во строк и столбцов массива
k	integer	кол-во особых чисел в массиве
s	Double	сумма элементов текущей строки

4. Процедура Output для вывода количества особых чисел на форму.

```
procedure TfrmMain.Output(const k:Integer);
```

Параметры процедуры представлены в таблице 7 :

Таблица 7 - Параметры процедуры вывода количества особых чисел

<b>имя</b>	<b>тип</b>	<b>предназначение</b>
k	integer	кол-во особых чисел

## Текст программы

Ниже представлен текст программы на языке Delphi 7, реализующей расчет количества “особых чисел” в массиве, т.е. таких, которые больше суммы остальных элементов столбца.

```
unit UnitMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, StdCtrls;

type
  DDArray = Array of array of Double;
  TfrmMain = class (TForm)
    sgdArray: TStringGrid; //таблица — элементы массива,
    btnRun: TButton; //кнопка запуска
    btnAddCol: TButton; //кнопка добавления столбца
    btnDelCol: TButton; //кнопка добавления столбца
    btnDelRow: TButton; //кнопка добавления столбца
    btnAddRow: TButton; //кнопка добавления столбца
    grpResult: TGroupBox; // группа вывода результата
    lblResult: TLabel; // поле вывода результата
    lblInfo: TLabel; // информация о программе
    procedure btnRunClick(Sender: TObject);
    procedure btnAddRowClick(Sender: TObject);
    procedure btnDelRowClick(Sender: TObject);
    procedure btnDelColClick(Sender: TObject);
    procedure btnAddColClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure grpResultClick(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
  private
    { Private declarations }
    function Input(var a:DDArray):boolean;
    procedure Output(const k:Integer);
  public
    { Public declarations }
  end;

var
  frmMain: TfrmMain;

implementation
  (*
  Функция GetSpec для расчета количества ''особых чисел'' в массиве, т.е. таких,
  которые больше суммы остальных элементов строки.
  Возвращает количество ''особых'' чисел.
  Параметры:
  a — массив для поиска особых элементов
  Локальные переменные:
  i, j — переменные-счетчики для доступа к элементам массива,
  n, m — кол-во строк и столбцов массива
  k — кол-во особых чисел в массиве
  s — сумма элементов текущей строки
  *)
  function GetSpec(const a:DDArray):Integer;
  var n, m, k, i, j:Integer; s:Double;
begin
  k:=0;
  n:=length(a); m:=length(a[0]);
```

```

for j:=0 to m-1 do begin
  s:=0;
  for i:=0 to n-1 do
    s:=s+a[i,j];
  for i:=0 to n-1 do begin
    MessageDlg(floattostr(i),mtError,[mbOK],0) ;
    if (a[i,j]>s-a[i,j]) then
      inc(k);
    end;
  end;
  GetSpec:=k;
end;

```

(\*

*Функция Input – вводит данные для поиска количества особых чисел матрицы. Возвращает True, если операция прошла успешно, иначе возвращает False.*

*Параметры:*

*a – массив для поиска особых элементов*

*Локальные переменные:*

*i, j – переменные-счетчики для доступа к элементам массива,*

*n, m – кол-во строк и столбцов массива*

*errors – список информации о возникших ошибках I/O.*

\*)

```

function TfrmMain.Input(var a:DDArray):boolean;
var i,j,n,m:Integer; errors:String;
begin
  n:=sgdArray.RowCount;m:=sgdArray.ColCount;
  SetLength(a,sgdArray.RowCount-1,sgdArray.ColCount-1);
  i:=0;
  if a<>NIL then
    while (i<n)and(a[i]<>NIL) do
      inc(i);
    if i<>n then
      errors:=errors+'Ошибка выделения памяти!'+'#13#10';
  if errors='' then
    for i:= 1 to n-1 do
      for j:=1 to m-1 do
        try
          a[i-1][j-1]:=StrToFloat(sgdArray.Cells[j,i])
        except
          on EConvertError do
            errors:=errors+'Ошибка при чтении ячейки ['+'
              IntToStr(i)+'+', '+'IntToStr(j)+'']!'+'#13#10';
        end;
    if errors<>'' then begin
      Input:=False;
      MessageDlg('Ошибки:'+errors,mtError,[mbOK],0)
    end else Input:=True;
end;

```

(\*

*Процедура Output для вывода количества особых чисел на форму.*

*Параметры:*

*k – кол-во особых чисел*

\*)

```

procedure TfrmMain.Output(const k:Integer);
begin
  lblResult.Caption:= ' "Особых" элементов:'+'IntToStr(k);
end;
{$R *.dfm}

```

(\*

*Процедура btnRunClick – основная процедура программы расчета кол-ва особых значений.*

*Локальные переменные:*

*a – массив для поиска особых элементов*

*k – кол-во особых чисел*

```

*)
procedure TfrmMain.btnRunClick(Sender: TObject);
var a:DDArray;k:Integer;
begin
    if Input(a) then begin
        k:=GetSpec(a);
        Output(k);
    end;
end;

procedure TfrmMain.btnAddRowClick(Sender: TObject);
begin
    with sgdArray do begin
        RowCount:=RowCount+1;
        Cells[0,RowCount-1]:=intToStr(RowCount-1);
    end;
end;

procedure TfrmMain.btnDelRowClick(Sender: TObject);
begin
    with sgdArray do begin
        RowCount:=RowCount-1;
    end;
end;

procedure TfrmMain.btnDelColClick(Sender: TObject);
begin
    with sgdArray do begin
        ColCount:=ColCount-1;
    end;
end;

procedure TfrmMain.btnAddColClick(Sender: TObject);
begin
    with sgdArray do begin
        ColCount:=ColCount+1;
        sgdArray.Cells[ColCount-1,0]:=intToStr(ColCount-1);
    end;
end;

procedure TfrmMain.FormCreate(Sender: TObject);
var i:integer;
begin
    for i:=1 to 2 do begin
        sgdArray.Cells[0,i]:=intToStr(i);
        sgdArray.Cells[i,0]:=intToStr(i);
    end;
end;

end.

```

## Тестовый пример

Ниже на рисунке 1 представлен пример работы программы с правильными исходными данными.

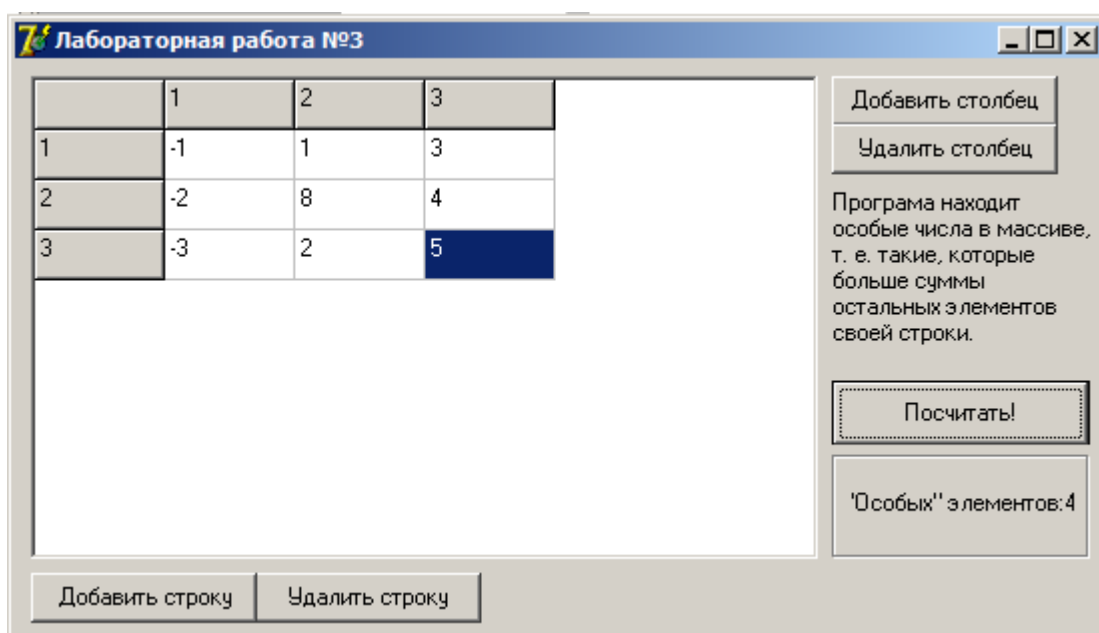


Рисунок 1 - Пример работы программы с правильными исходными данными

На рисунке 2 представлен пример работы программы с неправильными исходными данными.

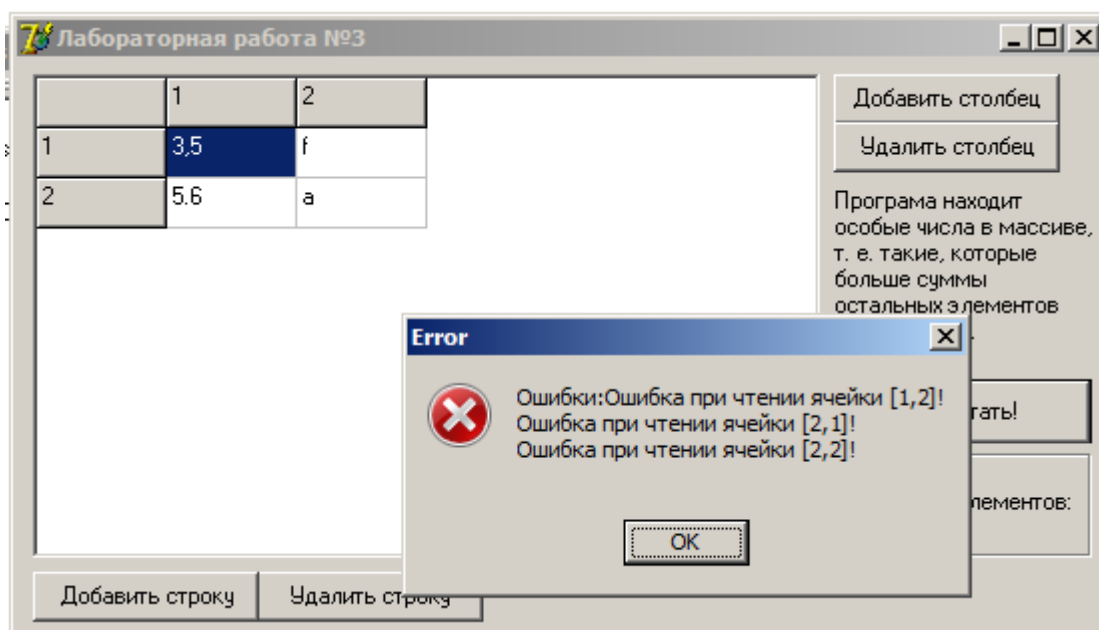


Рисунок 2 - Пример работы программы с правильными исходными данными

## **Вывод**

В этой лабораторной работе я изучил реализацию массивов в Delphi, и методы обработки исключительных ситуаций. Массивы позволяют обрабатывать большие объёмы данных, кроме того, динамические массивы позволяют просто и удобно масштабироваться при изменении объема данных. Концепция исключений и обработка исключений предоставляют высокоуровневый интерфейс для предотвращения ошибок времени выполнения.