



**Томский государственный университет**



**Факультет информатики**

**А.В. Скворцов, Т.Н. Поддубная**

# **Ввод и обработка массивов**

**(методические указания к  
лабораторной работе № 3)**

**Томск – 1999**

Указания РАССМОТРЕНЫ и УТВЕРЖДЕНЫ методической комиссией факультета информатики.

Протокол № \_\_\_\_\_ от « \_\_\_\_\_ » \_\_\_\_\_ 1999 г.

Председатель методической комиссии факультета информатики,  
профессор \_\_\_\_\_ Поддубный В.В.

*Утверждено.* Зав. кафедрой прикладной информатики, доцент  
\_\_\_\_\_ Сущенко С.П. « \_\_\_\_\_ » \_\_\_\_\_ 1999 г.

Методические указания посвящены программной среде Borland Delphi компании Inprise Corporation – одному из мощнейших современных средств разработки приложений для Windows.

Указания разработаны для студентов межфакультетской специализации факультета информатики Томского государственного университета. Содержат подробное описание лабораторной работы: цель, задачи, описание работы и используемых компонентов Delphi. В конце работы приведены тексты программ.

*Рецензент* – канд. техн. наук, доцент **Ю.Л. Костюк.**

© Скворцов А.В., Поддубная Т.Н., 1999

© Оформление и верстка: Скворцов А.В., 1999

В данной лабораторной работе вы познакомитесь с некоторыми компонентами Delphi для ввода и обработки числовых одномерных и двумерных массивов данных. При этом необходимо будет создать главную форму, позволяющую вывести на экран другие вспомогательные формы для ввода значений массивов.

После ввода значений необходимо будет их обработать (вычислить сумму и произведение всех элементов) и вывести результаты в главной форме.

## **Цель работы**

Освоение приёмов работы с массивами.

## **Задачи работы**

1. Ознакомление с компонентами для представления списков и таблиц, которые можно использовать в Delphi для работы с одномерными и двумерными массивами.
2. Создание программы для ввода и обработки одномерных и двумерных массивов.

## **Описание работы**

Перед началом работы необходимо создать новый проект и сохранить его в отдельном каталоге Lab3 в соответствии с рекомендациями, приведенными в предыдущих лабораторных работах.

Наше приложение будет состоять из трех модулей. Первый модуль UnitMain.pas, соответствующий форме MainForm, предназначен для координации работ других модулей и обработки массивов. Второй модуль ListInput.pas, соответствующий форме ListForm, предназначен для ввода одномерного массива. Третий модуль GridInput.pas, соответствующий форме GridForm, предназначен для ввода двумерного массива. Сам же проект необходимо сохранить под именем Lab3.dpr.

Для визуализации работы с одномерным массивом можно использовать компонент класса TListBox со страницы Standard палитры компонентов. Этот визуальный компонент представляет собой стандартное окно списка Windows, позволяющее работать с отдельными элементами. К элементам списка разрешен индексный способ доступа, как это делается при работе с массивами. В списках могут храниться строковые значения в паре с некоторым целочисленным значением. В данной работе предполагается хранить в списках только строковые значения. Кроме того, элементы списка могут динамически добавляться и удаляться, что делает этот компонент удобным для отображения меняющихся данных.

Для выполнения данной работы нам понадобятся приведенные в табл. 4 свойства объектов типа TListBox, а в табл. 5-6 свойства и методы объектов типа TStringList, используемых для представления наборов строк в Delphi, а в данном случае для списка строк объекта TListBox.

**Таблица 4.** Основные свойства объектов типа TListBox

Свойство	Тип	Комментарий
Items	TStrings	Содержит массив строк, появляющихся в списке. Строки пронумерованы, начиная с 0
ItemIndex	Integer	Номер текущего выбранного элемента в списке. Если не выбрана ни одна строка, то значение свойства равно -1

**Таблица 5.** Основные свойства объектов типа TStringList

Свойство	Тип	Комментарий
Strings[i: integer]	String	Массив строк с номерами 0...Count-1
Count	Integer	Общее количество элементов

**Таблица 6.** Основные методы объектов типа TStringList

Метод	Комментарий
Add(s: String)	Добавляет новую строку в массив строк
Clear	Очищает массив (делает Count=0)

Для визуализации работы с двумерным массивом можно использовать компонент TStringGrid со страницы Additional палитры компонентов, предназначенный для создания таблиц, в ячейках которых располагаются произвольные текстовые строки.

В табл. 7 приведены некоторые свойства компонента TStringGrid, необходимые для выполнения данной лабораторной работы.

### **Замечание**

*Индексация элементов таблицы производится с помощью двух чисел – номеров колонки (столбца) и ряда (строки). Нумерация ведется, начиная с 0.*

**Таблица 7.** Основные свойства объектов типа TStringGrid

Свойство	Тип	Комментарий
Cells[ACol, ARow: Integer]	String	Определяет содержимое ячейки с табличными координатами (ACol, ARow)
Cols[ACol: Integer]	TStrings	Все строки колонки с номером ACol
Rows[ARow: Integer]	TStrings	Все строки ряда с номером ARow
ColCount	Integer	Общее количество столбцов
RowCount	Integer	Общее количество рядов
FixedCols	Integer	Количество фиксированных (заголовочных) столбцов
FixedRows	Integer	Количество фиксированных (заголовочных) рядов
Col	Integer	Номер столбца текущей ячейки
Row	Integer	Номер ряда текущей ячейки
Options	TGridOptions	Данное свойство множественного типа определяет ряд дополнительных параметров таблицы. В частности, нам наиболее важны следующие:
	goEditing	Разрешается редактирование ячеек таблицы
	goTabs	Разрешается перемещение по ячейкам с помощью клавиши Tab и комбинации Shift+Tab

При создании приложения, работающего с данными неизвестной заранее размерности, желательно не ограничивать пользователя какими-то жесткими рамками в количестве элементов данных. Именно поэтому мы и будем использовать списки и таблицы, позволяющие динамически менять свои размеры. В то же время для

### **Замечание**

При создании формы «Ввод двумерного массива» не забудьте установить значение True для подсвойства goEditing свойства Options компонента TStringGrid для разрешения редактирования в ячейках таблицы.

написания процедур обработки массивов удобно использовать такие стандартные средства языка Pascal, как динамические массивы. Скорость обращения к ним значительно превосходит скорость обращения к элементам списков и ячейкам таблиц, да и памяти требуют они меньше, что также немаловажно. Поэтому в данной работе необходимо позволить пользователю ввести числовые данные визуально с помощью списков и таблиц, а затем скопировать введенные значения в динамические массивы.

Объявление динамического массива выполняется аналогично обычному, но диапазон изменения индексов не указывается. При этом нумерация элементов массива производится, начиная с 0. Для изменения количества элементов в динамическом массиве используется функция `SetLength` с двумя параметрами: именем переменной динамического массива и количеством элементов в массиве, например:

```
var SomeArray: array of integer;
.....
SetLength(SomeArray,10);
for i:=0 to 9 do
    SomeArray[i]:=0;
```

Для создания динамического двумерного массива необходимо объявить динамический массив динамических массивов, а для изменения его размера вызвать функцию `SetLength` с еще одним дополнительным параметром, указывающим размерность массива по второй координате, например:

```
var SomeArray2: array of array of
integer;
.....
SetLength(SomeArray2,10,5);
for i:=0 to 9 do
    For j:=0 to 4 do
        SomeArray2[i,j]:=0;
```

### Замечание

*Не забывайте при внесении изменений почаще сохранять проект во избежание потери результатов своей работы! Однако при запуске приложения на выполнение этого делать не нужно, т.к. Delphi сама сохранит все модули и проект.*

### Замечание

*Если вы хотите вывести на экран форму, находящуюся в другом модуле, используйте методы `Show` и `ShowModal`. При этом не забудьте внести в список `uses` в секции `implementation` имена этих модулей с формами.*

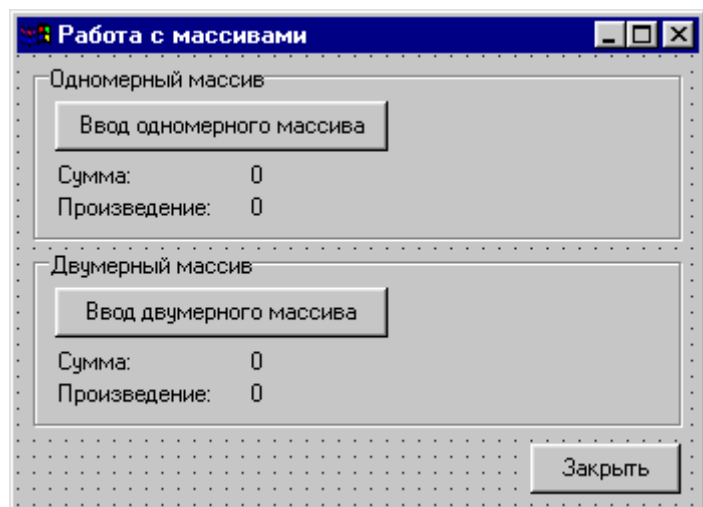


Рис. 10. Основная форма

На рис. 10-12 приведены варианты форм, которые могут быть использованы для выполнения лабораторной работы.

На главной форме используется компонент типа `TGroupBox` для объединения других компонентов, логически связанных друг с другом.

На форме для ввода одномерного массива основным является компонент `TListBox`. В метках, размещенных на этой форме, свойство `FocusedControl` позволяет их использовать для быстрого перемещения на указанный компонент с помощью метки. Перемещение можно осуществлять нажатием мышкой на метке либо нажатием на клавиатуре комбинации `Alt+<Подчеркнутая буква метки>`. Для того чтобы буква метки оказалась подчеркнутой, необходимо перед этой буквой в свойстве `Caption` поставить символ «&».

Например, для метки «Массив значений» необходимо установить свойство `Caption` равным «&Массив значений», а свойство `FocusedControl` – `ListboxArray`.

Нажатие на кнопку «Добавить» должно приводить к добавлению в конец списка текущего значения из строки ввода. Нажатие на кнопку «Заменить» должно приводить к замене выделенного значения в списке новым значением из строки ввода.

На форме для ввода двумерного массива основным является компонент `TStringGrid` для ввода значений массива. Компонент `TPanel` используется для отображе-

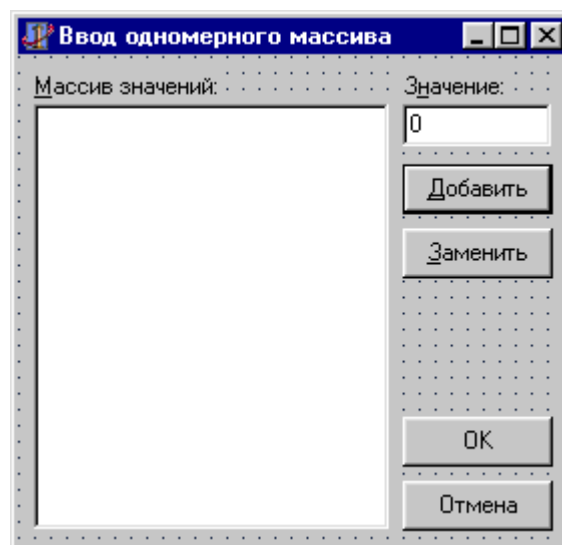


Рис. 11. Форма для ввода одномерного массива

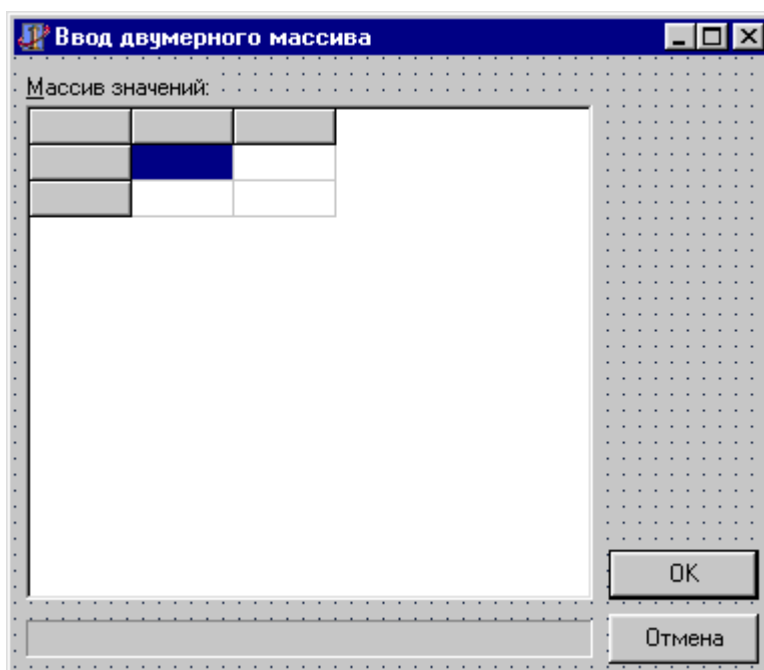


Рис. 12. Форма для ввода двумерного массива

ния сообщений об ошибке. У компонента `TPanel` необходимо установить свойства `BevelOuter` для изменения типа рамки, `Alignment` равным `alLeft`, для отображения текста, прижатого к левой стороне панели.

В первую очередь при создании модуля `GridInput.pas` необходимо написать обработчик события формы `OnCreate`. В нем следует установить начальные значения ячеек таблицы.

Также для компонента `GridArray` необходимо создать обработчик события `OnSetEditText` для проверки вводимых значений, который должен сообщать о возникающих ошибках, выводя информацию об этом в панель статуса `PanelStatus`. Также необходимо создать обработчик `OnKeyDown` нажатия клавиш. В нем следует сделать проверку на нажатие клавиши `Delete`, имеющей в `Windows` код `VK_DELETE`, чтобы при необходимости можно было стереть содержимое выделенной ячейки.

В данном модуле необходимо также создать метод формы `UpdateGrid`, используемый для автоматического сокращения количества ячеек в таблице. `Delphi` для такого рода методов не умеет создавать заготовки, поэтому это надо сделать полностью самостоятельно, описав этот метод в секции `private` и разместив тело метода в любом месте секции `implementation`.

В листингах 5-8 приведены тексты файла проекта и всех модулей.



**Листинг 5.** Текст файла проекта Lab3.dpr

```
program Lab3;

uses
  Forms,
  MainUnit in 'MainUnit.pas' {MainForm},
  ListInput in 'ListInput.pas' {ListForm},
  GridInput in 'GridInput.pas' {GridForm};

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(TMainForm, MainForm);
  Application.CreateForm(TListForm, ListForm);
  Application.CreateForm(TGridForm, GridForm);
  Application.Run;
end.
```

**Листинг 6.** Текст главного модуля MainUnit.pas

```
unit MainUnit;

interface

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

type
  TMainForm = class(TForm)
    GroupBox1: TGroupBox;
    ButtonList: TButton;
    Label1: TLabel;
    Label2: TLabel;
    LabelProduct1: TLabel;
    LabelSum1: TLabel;
    GroupBox2: TGroupBox;
    Label3: TLabel;
    Label4: TLabel;
    LabelProduct2: TLabel;
    LabelSum2: TLabel;
    ButtonGrid: TButton;
    ButtonClose: TButton;
    procedure ButtonListClick(Sender: TObject);
    procedure ButtonCloseClick(Sender: TObject);
    procedure ButtonGridClick(Sender: TObject);
  end;

var
  MainForm: TMainForm;

implementation

uses GridInput, ListInput;

{$R *.DFM}

procedure TMainForm.ButtonListClick(Sender: TObject);
var
  a: array of double; // Объявление динамического массива вещественных чисел
```

```
i: integer; s, p: double;
begin
  with ListForm do
    if ShowModal=mrOk then
      begin
        SetLength(a, ListBoxArray.Items.Count); // Установка длины массива
        for i:=0 to ListBoxArray.Items.Count-1 do
          a[i]:=StrToFloat(ListBoxArray.Items[i]); // Копирование значений в массив
        try
          s:=0;
          for i:=0 to Length(a)-1 do
            s:=s+a[i]; // Суммирование всех чисел
          LabelSum1.Caption:=FloatToStr(s);
        except
          LabelSum1.Caption:='Ошибка вычисления суммы';
        end;
        try
          p:=1;
          for i:=0 to Length(a)-1 do
            p:=p*a[i]; // Вычисление произведения всех чисел
          LabelProduct1.Caption:=FloatToStr(p);
        except
          LabelProduct1.Caption:='Ошибка вычисления произведения';
        end;
      end;
    end;
end;

procedure TMainForm.ButtonCloseClick(Sender: TObject);
begin // Выход из программы
  Close;
end;

procedure TMainForm.ButtonGridClick(Sender: TObject);
var
  a: array of array of double; // Объявление двумерного динамического массива
  x,y: integer; s, p: double;
begin
  with GridForm do
    if ShowModal=mrOk then
      begin
        SetLength(a, GridArray.ColCount-2); // Установка длины массива по столбцам
        for x:=0 to Length(a)-1 do
          SetLength(a[x], GridArray.RowCount-2); // Установка длины массива по рядам
        for x:=0 to Length(a)-1 do
          for y:=0 to Length(a[x])-1 do
            a[x,y]:=StrToFloat(GridArray.Cells[x+1,y+1]); // Копирование значений
          try
            s:=0;
            for x:=0 to Length(a)-1 do
              for y:=0 to Length(a[x])-1 do
                s:=s+a[x,y]; // Суммирование всех чисел
              LabelSum2.Caption:=FloatToStr(s);
            except
              LabelSum2.Caption:='Ошибка вычисления суммы';
            end;
            try
              p:=1;
              for x:=0 to Length(a)-1 do
                for y:=0 to Length(a[x])-1 do
                  p:=p*a[x,y];
                LabelProduct2.Caption:=FloatToStr(p);
```

```

    except
      LabelProduct2.Caption:='Ошибка вычисления произведения';
    end;
  end;
end;

end.

```

### Листинг 7. Текст вспомогательного модуля ListInput.pas.

```

unit ListInput;

interface

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

type
  TListForm = class(TForm)
    Label1: TLabel;
    ListBoxArray: TListBox;
    EditValue: TEdit;
    Label2: TLabel;
    ButtonAdd: TButton;
    ButtonReplace: TButton;
    ButtonOk: TButton;
    ButtonCancel: TButton;
    procedure ButtonAddClick(Sender: TObject);
    procedure ListBoxArrayClick(Sender: TObject);
  end;

var
  ListForm: TListForm;

implementation

{$R *.DFM}

procedure TListForm.ButtonAddClick(Sender:
  TObject);
var V: Double; S: string;
begin // Нажаты "Добавить" или "Заменить"
  EditValue.SetFocus; // Сделать строку ввода активной
  EditValue.SelectAll; // Выделить все символы в строке
  try
    V:=StrToFloat(EditValue.Text);
  except
    ShowMessage('Неверное вещественное значение');
    exit;
  end;
  S:=FloatToStr(V);
  if (Sender=ButtonReplace) and (ListBoxArray.ItemIndex>=0) then // Проверка нажатой кнопки
    ListBoxArray.Items[ListBoxArray.ItemIndex]:=S // Нажата кнопка "Заменить"
  else
    begin // Нажата кнопка "Добавить"
      ListBoxArray.Items.Add(S); // Добавление нового элемента в конец списка
      ListBoxArray.ItemIndex:=ListBoxArray.Items.Count-1; // Перемещение фокуса в конец списка
    end;
  end;
end;

// Следующая процедура предназначена для копирования выбранного элемента в списке для

```

#### Замечание

Обработчик ButtonAddClick события нажатия кнопки ButtonAdd должен быть назначен для кнопки ButtonReplace с помощью инспектора объектов на странице Events.

```
// последующего редактирования в строку ввода. Это позволяет изменять ранее введенные элементы
procedure TListForm.ListBoxArrayClick(Sender: TObject);
begin
  if ListBoxArray.ItemIndex>=0 then
    EditValue.Text:=ListBoxArray.Items[ListBoxArray.ItemIndex];
end;

end.
```

### Листинг 8. Текст вспомогательного модуля GridInput.pas.

```
unit GridInput;

interface

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

type
  TGridForm = class (TForm)
    GridArray: TStringGrid;
    ButtonOk: TButton;
    ButtonCancel: TButton;
    PanelStatus: TPanel;
    Label1: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure GridArraySetEditText(Sender: TObject; ACol, ARow: Integer; const Value: String);
    procedure ButtonOkClick(Sender: TObject);
    procedure GridArrayKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
  private
    procedure UpdateGrid; // Рассчитать количество строк в таблице
  end;

var
  GridForm: TGridForm;

implementation

{$R *.DFM}

procedure TGridForm.FormCreate(Sender: TObject);
begin
  with GridArray do
    begin
      ColWidths[0]:=35; // Установка ширины первой колонки
      Cells[0,1]:='1'; // Занесение значений в ячейки по умолчанию
      Cells[0,2]:='2';
      Cells[1,0]:='1';
      Cells[2,0]:='2';
      Cells[1,1]:='0';
    end;
end;

procedure TGridForm.GridArraySetEditText(Sender: TObject; ACol,
  ARow: Integer; const Value: String);
begin // Параметры ACol и ARow задают номер измененной ячейки, а Value - новое значение
  with GridArray do
    if Value='' then
      UpdateGrid
    else
      begin
```

```
    if ARow=RowCount-1 then // Добавить новый ряд
    begin
        RowCount:=ARow+2;
        Cells[0,ARow+1]:=IntToStr(ARow+1);
    end;
    if ACol=ColCount-1 then // Добавить новую колонку
    begin
        ColCount:=ACol+2;
        Cells[ACol+1,0]:=IntToStr(ACol+1);
    end;
    try
        StrToFloat(Value);
    except
        PanelStatus.Caption:='Неверное вещественное значение';
        exit;
    end;
end;
PanelStatus.Caption:='';
end;

procedure TGridForm.UpdateGrid;
var x,y,maxx,maxy: integer;
begin
    with GridArray do
    begin
        maxx:=1; maxy:=1; // Вычисляем максимальной номер заполненной строки
        for x:=1 to ColCount-1 do
            for y:=1 to RowCount-1 do
                if Cells[x,y]<>' ' then
                begin
                    if x>maxx then maxx:=x;
                    if y>maxy then maxy:=y;
                end;
            ColCount:=maxx+2;
            RowCount:=maxy+2;
        end;
    end;
end;

procedure TGridForm.GridArrayKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
begin // Нажата клавиша на клавиатуре, когда фокус ввода находится в таблице
    if Key=VK_DELETE then
        with GridArray do
            if not EditorMode and ((Col<>1) or (Row<>1)) then
                begin
                    Cells[Col,Row]:=' ';
                    UpdateGrid;
                end;
        end;
end;

procedure TGridForm.ButtonOkClick(Sender: TObject);
var x,y: integer;
begin // Перед закрытием формы необходимо проверить правильность введенных значений
    with GridArray do
        for y:=1 to RowCount-2 do
            for x:=1 to ColCount-2 do
                if Cells[x,y]=' ' then
                    begin
                        Col:=x; Row:=y; // Сделать незаполненную ячейку текущей
                        SetFocus; // Сделать таблицу активным элементом в форме
                        EditorMode:=True; // Включить режим редактирования значений ячейки
                        PanelStatus.Caption:='Значение не введено';
                    end;
            end;
        end;
end;
```

```
        exit;
    end else
    try
        StrToFloat(Cells[x,y]);
    except
        Col:=x; Row:=y; // Сделать неверную ячейку текущей
        SetFocus;      // Сделать таблицу активным элементом в форме
        EditorMode:=True; // Включить режим редактирования значений ячейки
        PanelStatus.Caption:='Неверное вещественное значение';
        exit;
    end;
    ModalResult:=mrOk; // Форму надо закрыть, т.к. ошибок не найдено
end;

end.
```

### **Вопросы и задания для самостоятельной работы**

1. Добавьте кнопку «Удалить» в форму для ввода одномерного массива и создайте соответствующий обработчик.
2. Разрешите ввод вещественных чисел как с десятичной точкой, так и с десятичной запятой, написав для этого соответствующую процедуру для преобразования строки в число.
3. Сделайте возможность сортировки списка строк в одномерном массиве.
4. Назначьте на все кнопки всех форм горячие клавиши.
5. Добавьте кнопку «Показать» в форму для ввода двумерного массива для отображения в окне сообщения значения текущей ячейки в таблице.

Учебное издание

Алексей Владимирович Скворцов  
Поддубная Тамара Николаевна

Ввод и обработка массивов

(методические указания к  
лабораторной работе № 2)