

Массивы

Массивом называется упорядоченный набор данных, имеющих один и тот же тип и располагающихся в памяти непосредственно друг за другом.

Массивы могут быть:

- Статические и динамические;
- Числовые и символьные;
- Одномерные и многомерные.

Как правило, одномерный массив называют вектором:

$A(7) = \{3, 8, 4, 1, 2, 5, 4\}$

Двумерный массив является частным случаем многомерного массива и представляет собой матрицу (таблицу)

$A(4,4) = \begin{Bmatrix} 3, 4, 5, 1 \\ 2, 9, 0, 3 \\ 4, 7, 9, 5 \\ 3, 2, 6, 7 \end{Bmatrix}$

Описание массивов осуществляется с помощью оператора **Dim**.

Статический массив:

```
Dim A(1 to 7) As Integer, B(1 to 7, 1 to 7) as Single
```

Динамический массив:

```
Dim N as Integer, A() as Single
```

```
Private Sub Command1_Click()
```

```
.....
```

```
N=CInt(InputBox("N="))
```

```
ReDim A(1 to N) as Single
```

```
.....
```

```
End Sub
```

Основные способы ввода и вывода элементов массива

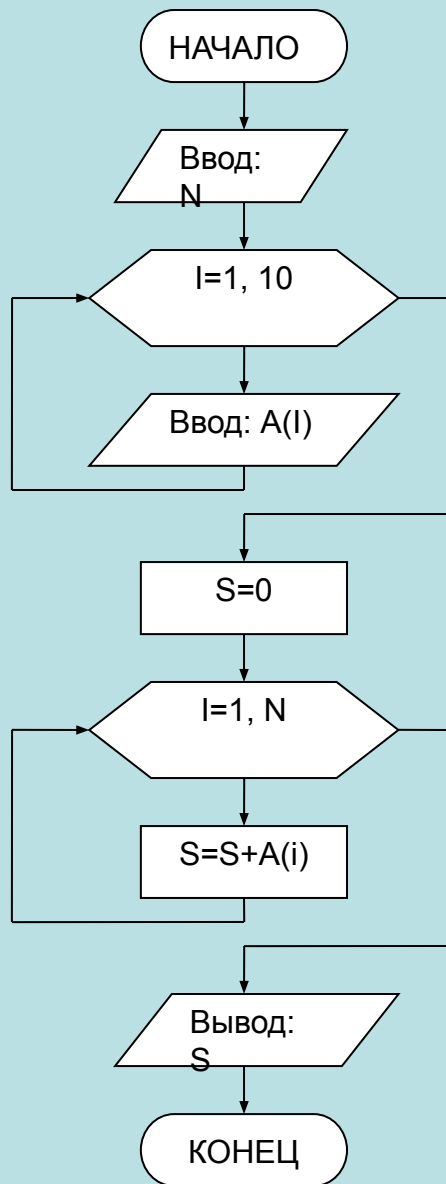
Ввод элементов одномерного массива осуществляется в цикле.

Ввод элементов двумерного массива осуществляется в двух

Вложенных циклах.

ПРИМЕР. Сумма десяти вещественных чисел.

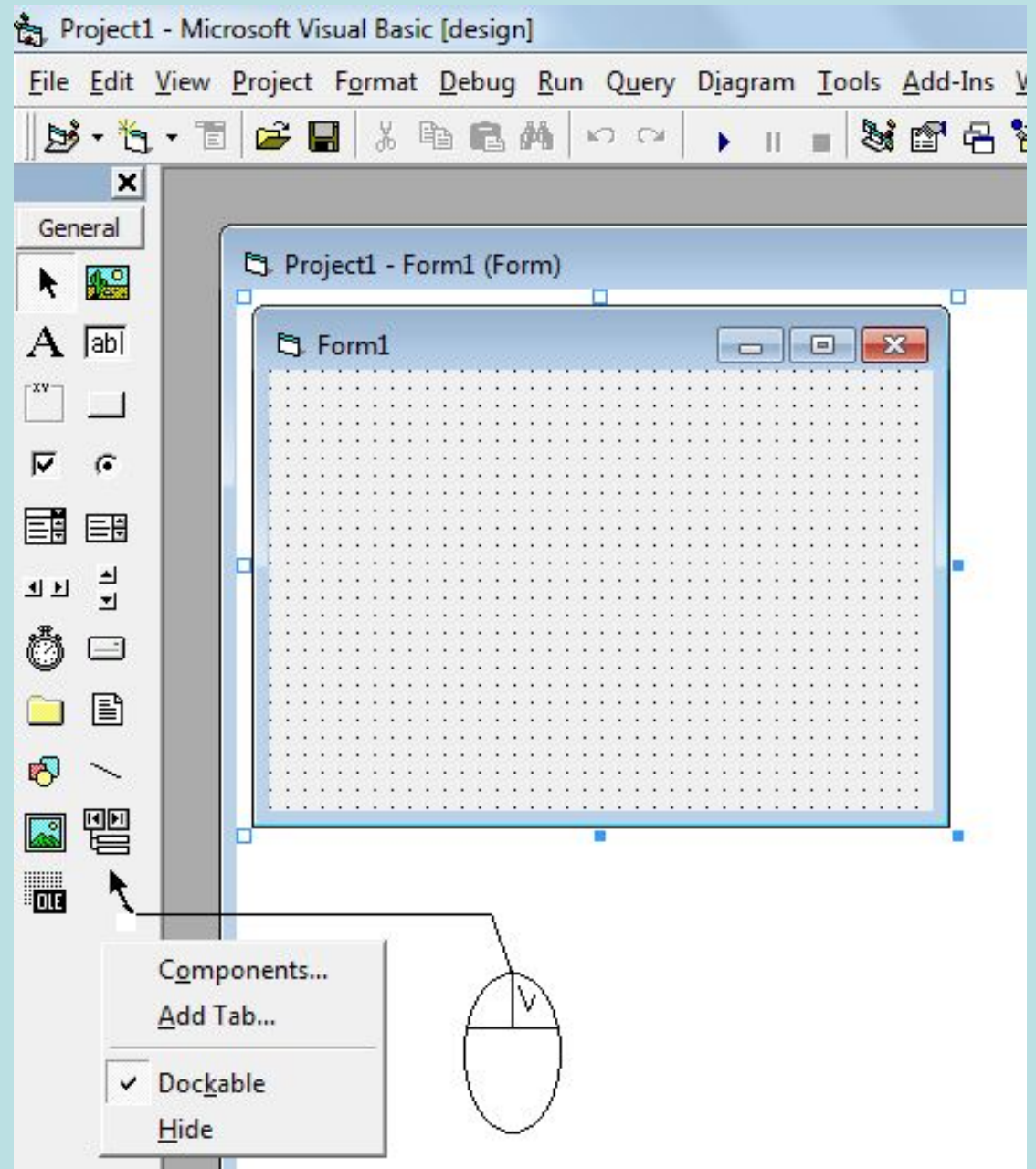
$$S = \sum_{i=1}^{10} a_i$$

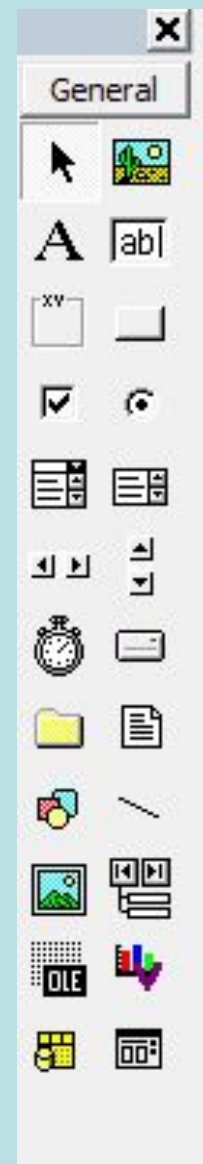
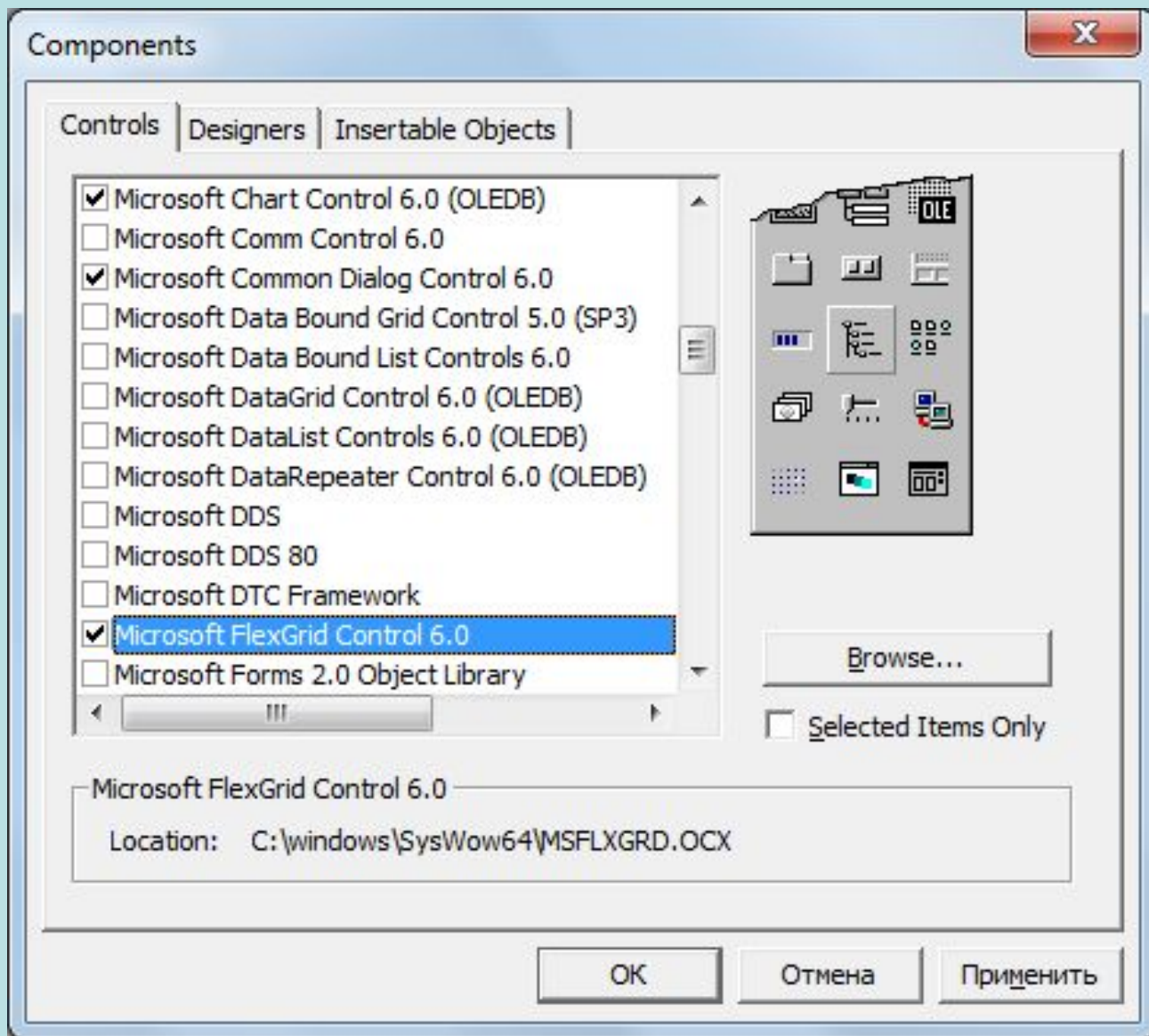


```

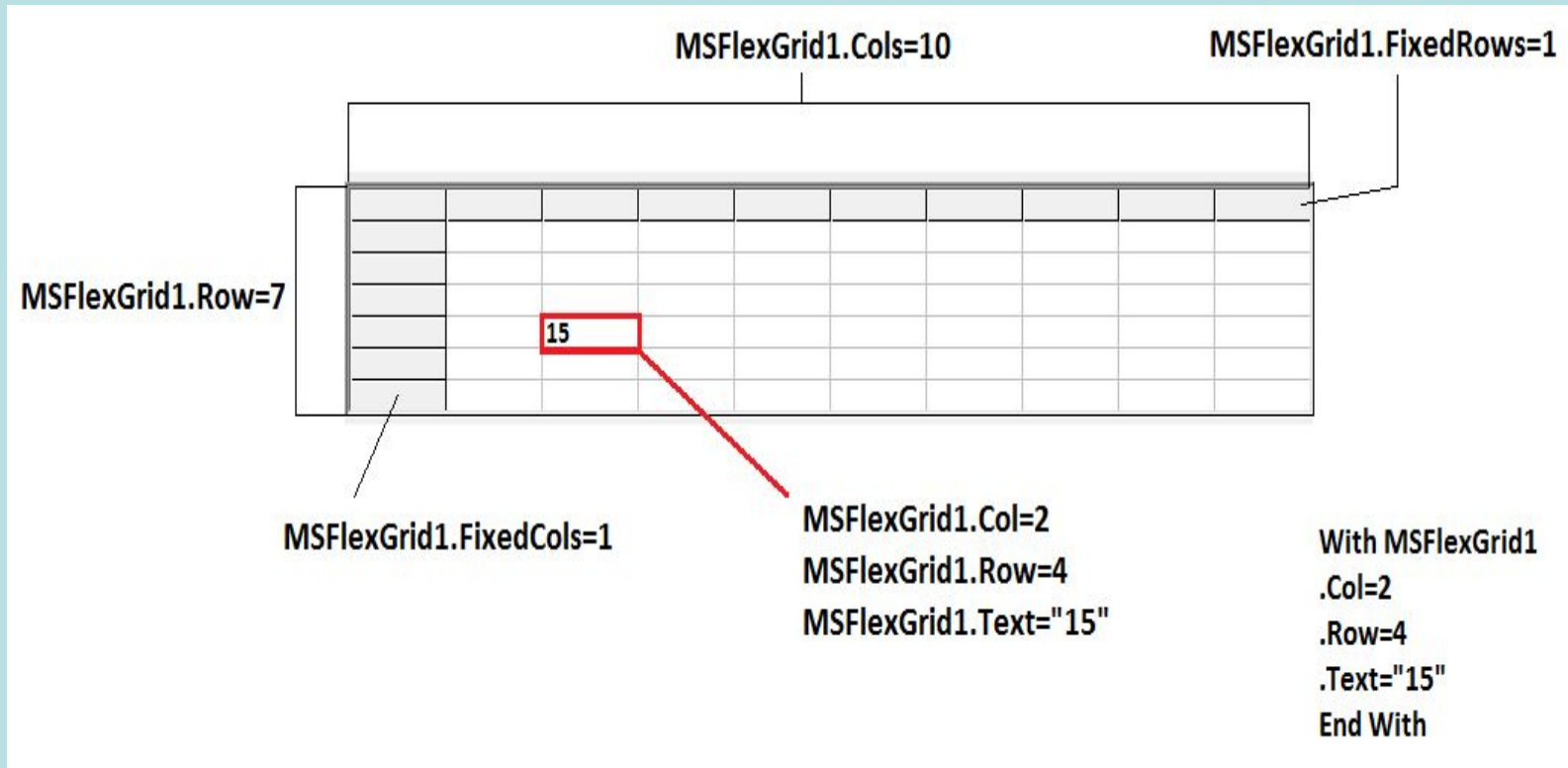
Dim a(1 To 10) As Single
Private Sub Command1_Click()
Dim i As Integer
For i = 1 To 10
a(i) = CSng(InputBox("A(" + CStr(i) + ")="))
Next i
End Sub
Private Sub Command2_Click()
Dim s As Single, i As Integer
s = 0
For i = 1 To 10
s = s + a(i)
Next i
Text1.Text = CStr(s)
End Sub
Private Sub Command3_Click()
End
End Sub
  
```

ПОДКЛЮЧЕНИЕ ДОПОЛНИТЕЛЬНЫХ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ

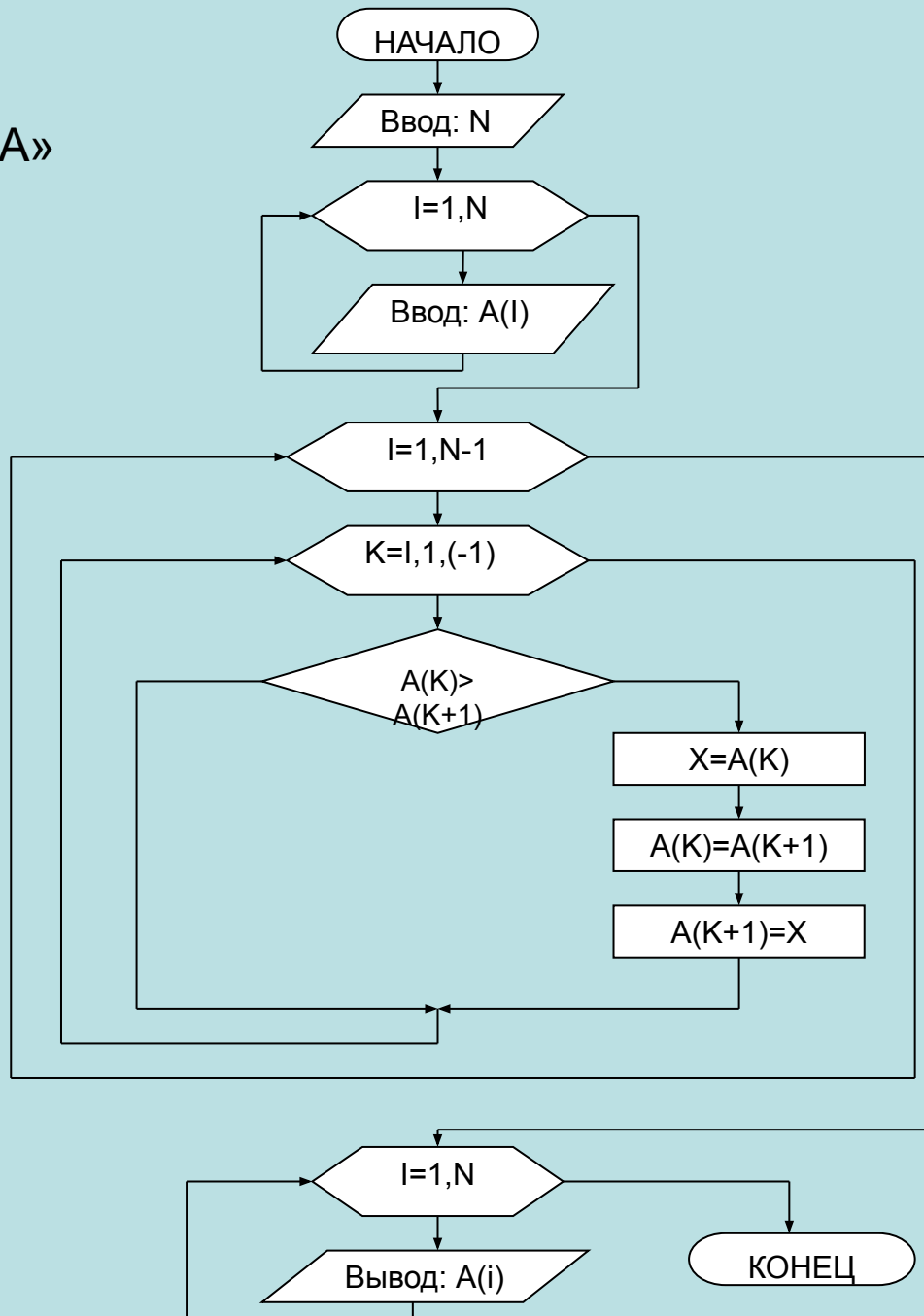




Microsoft FlexGrid Control 6.0



СОРТИРОВКА
МЕТОД «ПУЗЫРЬКА»



Form1

Не отсортированный массив	Отсортированный массив
9	2
4	3
3	3
5	4
4	4
3	5
2	9

Количество элементов массива 7

ВВОД

СОРТИРОВАТЬ

ВЫХОД

```
Dim a() As Single, n As Integer
```

```
Private Sub Command1_Click()
```

```
Dim i As Integer
```

```
List1.Clear
```

```
Open App.Path + "\massivin.txt" For Input As #1
```

```
Input #1, n
```

```
Label4.Caption = CStr(n)
```

```
ReDim a(1 To n) As Single
```

```
For i = 1 To n
```

```
Input #1, a(i)
```

```
List1.AddItem (CStr(a(i)))
```

```
Next i
```

```
Close #1
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Dim x As Single, i As Integer, k As Integer
```

```
For i = 1 To n - 1
```

```
For k = i To 1 Step -1
```

```
If a(k) > a(k + 1) Then
```

```
x = a(k)
```

```
a(k) = a(k + 1)
```

```
a(k + 1) = x
```

```
End If
```

```
Next k
```

```
Next i
```

```
For i = 1 To n
```

```
List2.AddItem (CStr(a(i)))
```

```
Next i
```

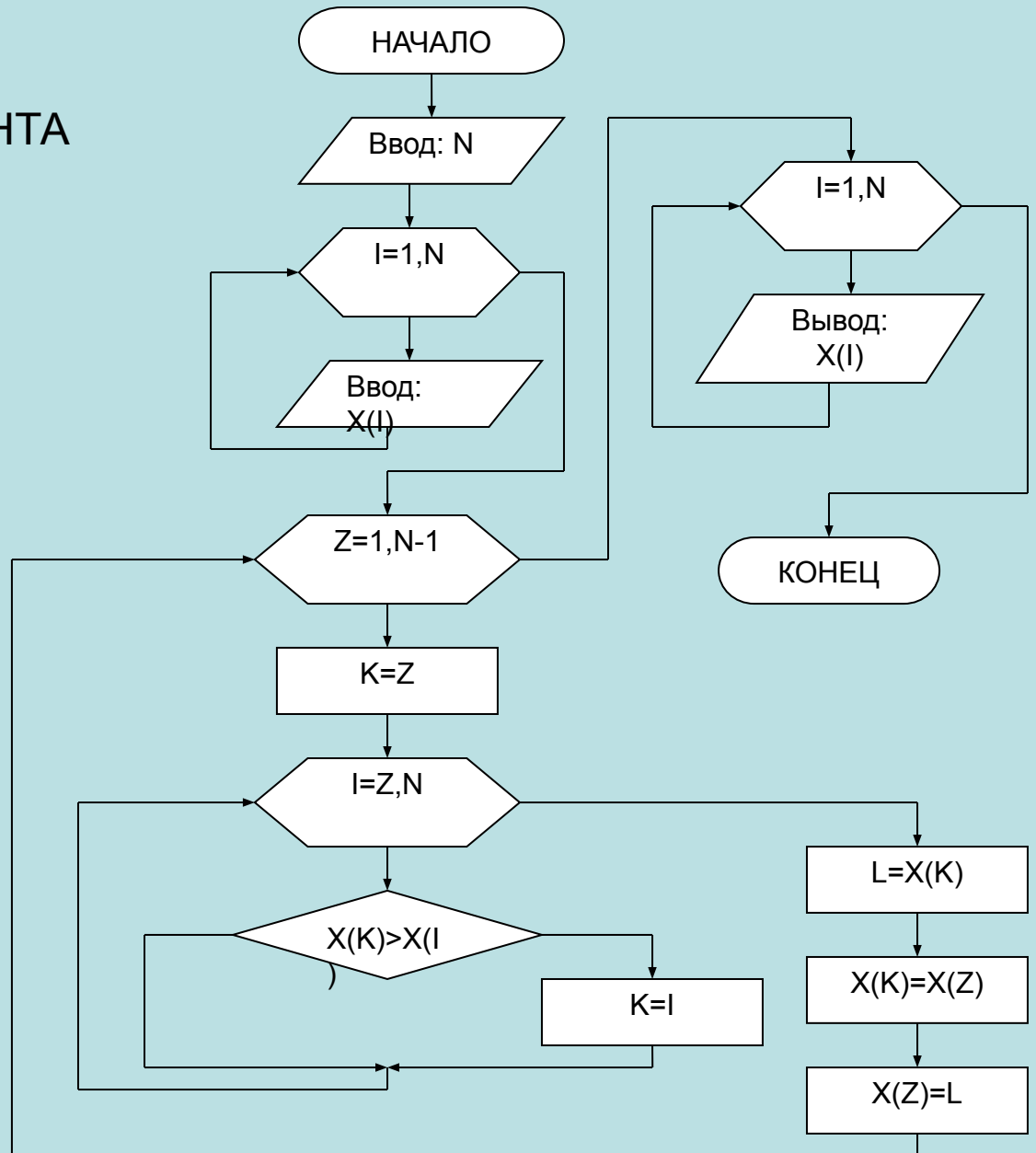
```
End Sub
```

```
Private Sub Command3_Click()
```

```
End
```

```
End Sub
```

СОРТИРОВКА ВЫБОРОМ МИНИМАЛЬНОГО ЭЛЕМЕНТА



Dim x() As Single, n As Integer

Private Sub Command1_Click()

Dim i As Integer

n = **CInt**(InputBox("N="))

MSFlexGrid1.Rows = n + 1

MSFlexGrid1.Row = 0

MSFlexGrid1.Col = 0

MSFlexGrid1.Text = "i"

MSFlexGrid1.Col = 1

MSFlexGrid1.Text = "Не отсорт"

MSFlexGrid1.Col = 2

MSFlexGrid1.Text = "Отсорт"

ReDim x(1 To n) As Single

For i = 1 To n

x(i) = **CSng**(InputBox("a(" + **CStr**(i) + ")="))

MSFlexGrid1.Row = i

MSFlexGrid1.Col = 0

MSFlexGrid1.Text = **CStr**(i)

MSFlexGrid1.Col = 1

MSFlexGrid1.Text = **CStr**(x(i))

Next i

End Sub

Private Sub Command3_Click()

End

End Sub

The screenshot shows a Windows application window titled "Form1". Inside the window, there is a table with the title "МАССИВЫ". The table has three columns: "i", "Не отсорт", and "Отсорт". The rows contain numerical data. Below the table, there are three buttons: "ВВОД", "СОРТИРОВКА", and "ВЫХОД".

i	Не отсорт	Отсорт
1	6	1
2	8	2
3	9	2
4	56	3
5	45	3,45
6	3,45	5
7	5	6
8	3	7
9	2	8
10	1	9
11	7	45
12	2	56

Private Sub Command2_Click()

Dim z As Integer, i As Integer, k As Integer, L As Single

For z = 1 To n - 1

k = z

For i = z To n

If x(k) > x(i) **Then** k = i

Next i

L = x(k)

x(k) = x(z)

x(z) = L

Next z

MSFlexGrid1.Col = 2

For i = 1 To n

MSFlexGrid1.Row = i

MSFlexGrid1.Text = **CStr**(x(i))

Next i

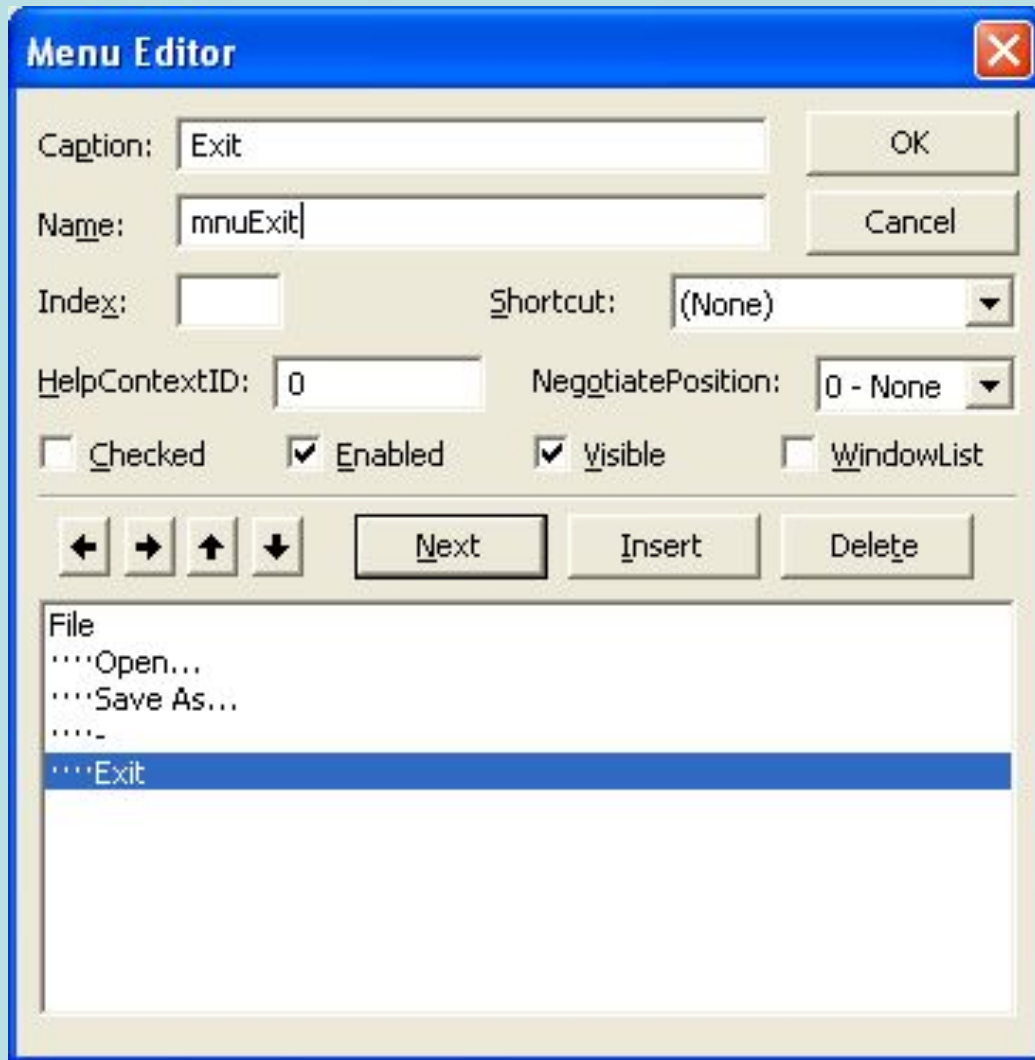
End Sub

ДВУМЕРНЫЕ МАССИВЫ

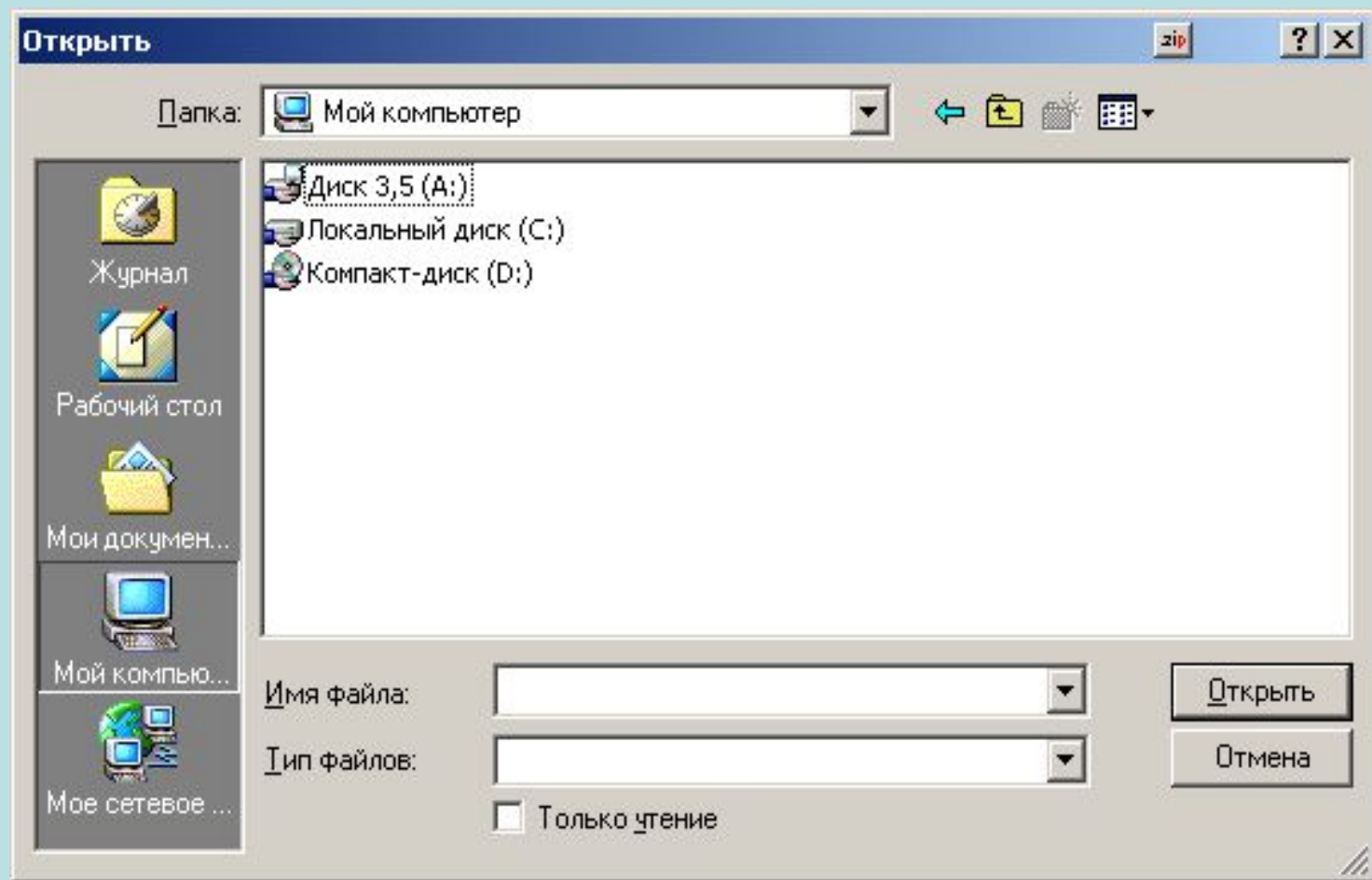
Ввод двумерного массива целесообразно осуществлять из файла, так как при отладке программы экономится время на ввод значений элементов массива.

ПРИМЕР. Ввод двумерного массива из файла и вывод его в таблицу и файл.

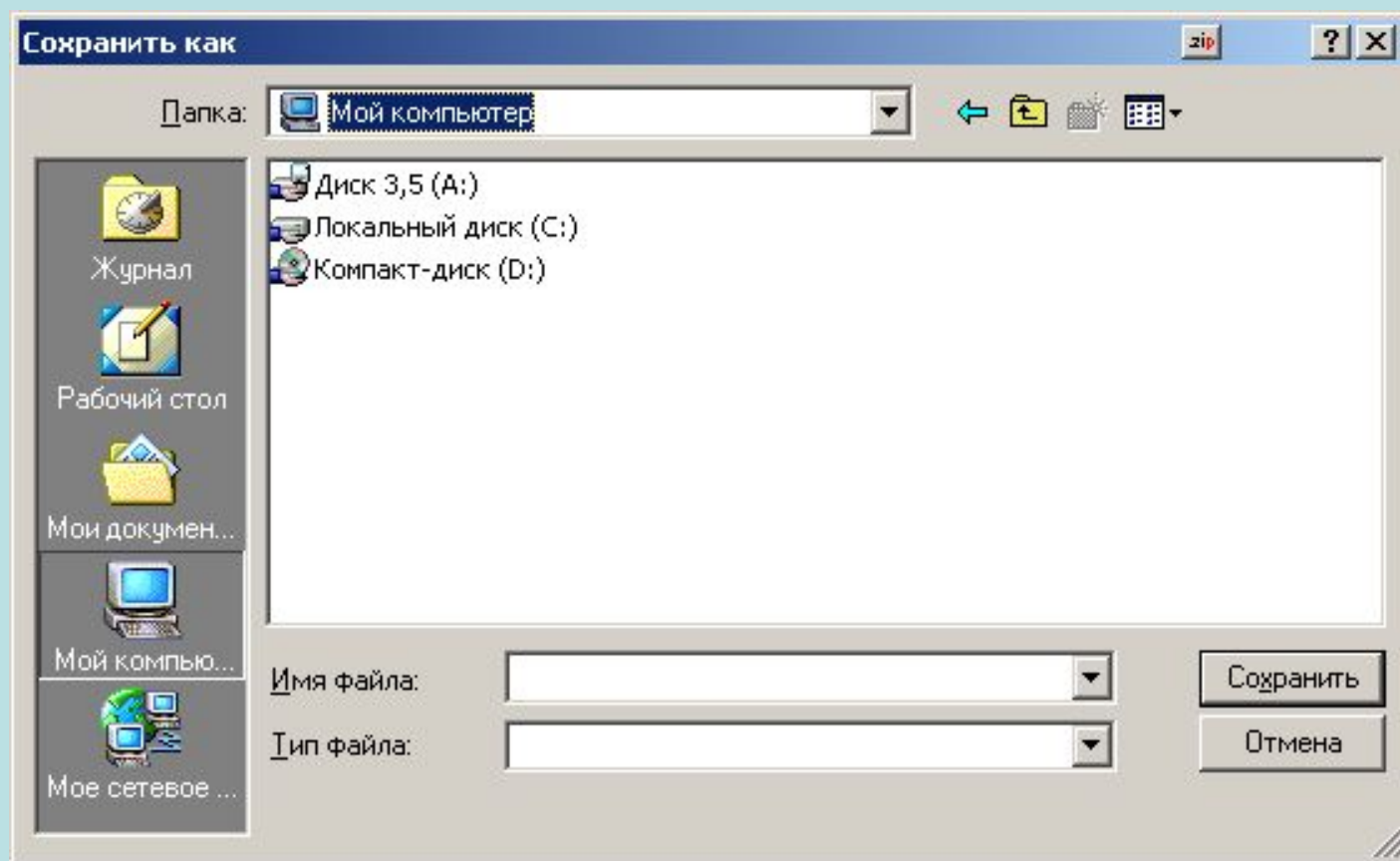
СОЗДАНИЕ ГЛАВНОГО МЕНЮ



ДИАЛОГОВЫЕ ОКНА



ДИАЛОГОВЫЕ ОКНА



ПРОГРАММНЫЙ КОД

```
Dim a() As Single, n As Integer, m As Integer
```

```
Private Sub mnuExit_Click()
```

```
End
```

```
End Sub
```

```
Private Sub mnuOpen_Click()
```

```
Dim s As String, i As Integer, j As Integer
```

```
CommonDialog1.Action = 1
```

```
s = CommonDialog1.FileName
```

```
Open s For Input As #1
```

```
Input #1, n, m
```

```
ReDim a(1 To n, 1 To m) As Single
```

```
For i = 1 To n
```

```
For j = 1 To m
```

```
Input #1, a(i, j)
```

```
Next j
```

```
Next i
```

```
Close #1
```

```
End Sub
```

```
Private Sub mnuSave_Click()  
Dim s As String, i As Integer, j As Integer  
CommonDialog1.Action = 2  
s = CommonDialog1.FileName  
Open s For Output As #1  
With MSFlexGrid1  
.Rows = n + 1  
.Cols = m + 1  
For i = 1 To n  
.Col = 0  
.Row = i  
.Text = CStr(i)  
For j = 1 To m  
.Row = 0  
.Col = j  
.Text = CStr(j)  
.Row = i  
.Text = CStr(a(i, j))  
Print #1, a(i, j); " ";  
Next j  
Print #1, ""  
Next i  
Close #1  
End With  
End Sub
```

ПРОЦЕДУРЫ

Visual Basic поддерживает следующие виды процедур:

- Процедуры общего вида, создаваемые пользователем Sub...End Sub.
- Процедуры – функции, создаваемые пользователем Function...End Function
- Процедуры обработки событий.
- Процедуры свойств Get() и Set().

Процедура Sub...End Sub

Формат:

```
Sub <Имя>( <перем1> as <тип1>, <перем2> as <тип2>, ..., <перемN> _  
                as <типN>)
```

```
<тело процедуры>
```

```
End Sub
```

Для вызова процедуры используется ключевое слово **Call**

```
Private Sub Command1_Click()
```

```
.....
```

```
Call <Имя>( <перем1>, <перем2>, ..., <перемN>)
```

```
.....
```

```
End Sub
```

Процедура-функция Function...End Function

Формат:

```
Function <Имя>( <перем1> as <тип1>, <перем2> as <тип2>, ..., <перемN> _  
                as <типN>) as <тип>
```

<тело функции>

```
End Function
```

Вызов процедуры-функции похож на вызов стандартной математической функции.

```
Private Sub Command1_Click()
```

```
.....
```

```
Z=<Имя>( <перем1>, <перем2>, ..., <перемN> )
```

```
.....
```

```
End Sub
```

ПЕРЕДАЧА ПАРАМЕТРОВ

Передача параметров в процедуры

Sub...End Sub и **Function...End Function**

может осуществляться как по ссылке, так и по значению.

Передача параметров по значению:

Sub Name1(**By Val** x() **as** Single, **By Val** N **as** Single)

....

End Sub

ПЕРЕДАЧА ПАРАМЕТРОВ

Передача параметров по ссылке:

```
Sub Name1(By Ref x() as Single, By Ref N as Single)
```

```
....
```

```
End Sub
```

или

```
Sub Name1(x() as Single, N as Single)
```

```
....
```

```
End Sub
```

По умолчанию параметры передаются по ссылке.

ПРИМЕРЫ

Поиск коэффициентов уравнения
регрессии $y=a*x+b$

**Sub MNK1(x() as single, y() as single, N as integer, a as single, _
b as single)**

Dim i as integer, s1 as single, s2 as single, s3 as single, s4 as single
s1=0:s2=0:s3=0:s4=0

For i=1 to N

s1=s1+x(i)

s2=s2+x(i)*x(i)

s3=s3+x(i)*y(i)

s4=s4+y(i)

Next i

$a=(N*s3-s1*s4)/(N*s2-s1*s1)$

$b=(s2*s4-s1*s3)/(N*s2-s1*s1)$

End Sub

ПРИМЕРЫ

Поиск коэффициента уравнения регрессии $y=a*x$

Function A (x() as single, N as integer) as single

Dim i as integer, s1 as single, s2 as single

s1=0:s2=0

For i=1 to N

s1=s1+x(i)*y(i)

s2=s2+x(i)*x(i)

Next i

A=s1/s2

End Function