



Государственное образовательное учреждение
высшего профессионального образования
Санкт-Петербургский
государственный технологический институт
(технический университет)



ДОБРО ПОЖАЛОВАТЬ



Кафедра системного анализа

Базовые программы обработки двумерного массива

1. Ввод двумерного массива

```
Dim a(,) As Single
Private Sub Button1_Click()
Dim n As Integer, m as Integer
Dim i As Integer, j as Integer
n = CSng(InputBox("Введите n"))
m = CSng(InputBox("Введите m"))
ReDim a(0 To n-1, 0 To m-1)
For i = 0 To n-1
    For j = 0 To m-1
        a(i,j)=Csng(InputBox("a(" + Cstr(i) + "," + Cstr(j) + ")"))
    Next j,i
```

2. Нахождение суммы элементов каждого столбца матрицы

```
For j=0 To m-1
```

```
S(j)=0
```

```
  For i=0 To n-1
```

```
S(j)=S(j)+x(i,j)
```

```
  Next i
```

```
Textbox1.Text=Textbox1.Text+"s("+CStr(j)+")="+CStr(S(j))+vbCrLf
```

```
Next j
```

3. Определение максимальных (минимальных) элементов строк матрицы

```
For i=0 To n-1
```

```
max(i)=x(i,1)
```

```
  For j=0 To m-1
```

```
    If x(i,j)>max(i) Then max(i) =x(i,j)
```

```
  Next j
```

```
  TextBox1.Text=TextBox1.Text+"max(" +CStr(i)+")=" +CStr(max(i))+vbCrLf
```

```
Next i
```

4. Вывод двумерного массива

```
For i = 0 To n-1
```

```
    For j = 0 To m-1
```

```
        TextBox1.Text = TextBox1.Text + CStr(x(i, j)) + " "
```

```
    Next j
```

```
    TextBox1.Text = TextBox1.Text + vbCrLf
```

```
Next i
```

Процедуры

Программирование можно упростить, разбивая задачу на небольшие логические компоненты - процедуры. Процедуры используются для реализации последовательности повторяющихся действий, например, часто повторяющихся вычислений.

преимущества:

- Процедуры позволяют разбивать программы на конечное число логических единиц, каждую из которых легче отладить, чем всю программу без процедур.
- Процедуры, разработанные для одной программы, могут выступать в качестве строительных блоков для других программ, обычно с небольшими изменениями или совсем без них.

Функция - подпрограмма которую вызывают чтобы выполнить какие либо расчеты. Когда она завершает работу то *возвращает управление* вызывающей программе и *передает* ее результат расчета.

Процедура - подпрограмма, которую вызывают для выполнения каких либо действий, но от которой не требуется возвращать основной программе какие либо значения. Другими словами процедура это любая подпрограмма которая не является функцией.

Объявление процедуры и функции

Синтаксис объявления процедуры:

```
[Private | Public] [Static] Sub Имя [(параметры)]  
    [Операторы]  
    .....  
End Sub
```

Синтаксис объявления функции:

```
[Private | Public] [Static] Function Имя [(параметры)] [As Тип]  
    [Операторы]  
    [Имя=Выражение]  
    .....  
End Function
```


Параметр	Описание
Private	Процедуры объявленные как Private , можно вызывать только в текущем модуле.
Public	Процедуры объявленные как Public , можно вызывать в <i>любом</i> модуле приложения.
Static	Все переменные объявление в процедуре будут <i>статическими</i> т.е. их значения сохраняются между вызовами.
Имя	Имя процедуры, удовлетворяющее стандартным правилам описания переменных. Этот идентификатор не может быть использован нигде, кроме обращения к процедуре.
Параметры	Список <i>формальных</i> параметров, значения которых передаются в процедуру. Разделителем в списке параметров является запятая (,).
Операторы	Любая группа операторов, выполняемых в процедуре или функции.
Тип	Тип возвращаемого значения функции.
Выражение	Возвращаемое значение функции.

Пример описание процедуры вычисления суммы
одномерного массива:

```
Public Sub sum(n As Integer, x() As Single, s As Single)
Dim i As Integer
s = 0
For i = 0 To n-1
s = s + x(i)
Next i
End Sub
```

Пример описание процедуры функции вычисления суммы одномерного массива

```
Public Function sum(n As Integer, x() As Single) As Single
Dim i As Integer, s As Single
s = 0
For i = 0 To n-1
s = s + x(i)
Next i
sum = s
End Function
```

Обращение к процедуре

Используется оператор вызова процедуры, который имеет следующий вид:

Call Имя [(Список *фактических* параметров)]

Где:

Имя – это имя вызываемой процедуры, определенной в программе при помощи оператора Sub.

Список фактических параметров – это список фактических параметров, разделенных запятыми.

В качестве фактических параметров можно использовать константы, переменные, выражения и массивы.

Если фактическим параметром является массив, то в списке параметров пишется имя массива, а затем ().

Фактические параметры должны соответствовать формальным по количеству, порядку следования и типу.

Порядок обращения к процедуре

1. Вызов процедуры при помощи оператора Call.
2. Формальные параметры в процедуре заменяются на фактические.
3. Вычисление при помощи данной процедуры с фактическими параметрами.
4. Возврат в основную программу к оператору, следующему после обращения.

Пример обращения к процедуре:

```
Private Sub button1_Click()  
Dim a() As Single, s As Single, i As Integer, n As Integer  
n = CSng(InputBox("n"))  
Redim a(0 to n-1)  
For i = 0 To n-1  
a(i) = CSng(InputBox("a(i)"))  
Next i  
Call sum(n, a, s)  
TextBox1.Text=TextBox1.Text + "s="+CStr(s)  
End Sub
```

Вызов процедуры Function

Процедура Function вызывается с помощью указателя функции, состоящего из имени, за которым следует список фактических параметров, заключенный в скобки.

Порядок обращения к функции

1. Вызов функции с помощью указателя функции;
2. Формальные параметры функции заменяются соответствующими фактическими;
3. Вычисляется значение функции и результат присваивается имени функции;
4. Управление передается к следующему после обращения действию.

Пример обращения к функции:

```
Private Sub button1_Click()  
Dim a() As Single, s As Single, i As Integer, n As Integer  
n = CSng(InputBox("n"))  
Redim a(0 to n-1)  
For i = 0 To n-1  
a(i) = CSng(InputBox("a(i)"))  
Next i  
s = sum(n,a)  
TextBox1.Text = TextBox1.Text + "s=" + CStr(s)  
End Sub
```


Формальные и фактические параметры

Формальные параметры подпрограммы указывают, с какими параметрами следует обращаться к этой подпрограмме (количество параметров, их последовательность, типы). Формальные параметры - это искусственные переменные, они показывают, что должно быть сделано с теми *реальными* переменными, которые будут указаны как фактические параметры.

Формальные параметры используются в операторе описания процедуры или функции.

Фактические параметры – это реальные переменные, с которыми происходит обращение к процедуре или функции.

Соответствие формальных и фактических параметров

Формальный параметр	Фактический параметр
Простая переменная	Константа Простая переменная Переменная с индексом (элемент массива) Выражение
Массив	Массив

VB требует строгого соответствия типа между формальными и фактическими параметрами, т.е. например, Integer - Integer , Single – Single.